Contributions to Software Engineering and to the Development and Deployment of International Software Engineering Standards for Very Small Entities

Claude Laporte

▶ To cite this version:

Claude Laporte. Contributions to Software Engineering and to the Development and Deployment of International Software Engineering Standards for Very Small Entities. Computer Science [cs]. Université de Bretagne occidentale - Brest, 2009. English. NNT : . tel-00483255

HAL Id: tel-00483255

https://theses.hal.science/tel-00483255

Submitted on 13 May 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE / UNIVERSITÉ DE BRETAGNE OCCIDENTALE

sous le sceau de l'Université européenne de Bretagne

pour obtenir le titre de

DOCTEUR DE L'UNIVERSITÉ DE BRETAGNE OCCIDENTALE

Mention : Informatique et applications

École Doctorale SICMA

présentée par

Claude Y. Laporte

Préparée à l'Equipe d'accueil 3883
Laboratoire d'Informatique des Systèmes Complexes

Contributions to Software Engineering and to the Development and Deployment of International Software Engineering Standards for Very Small Entities

Contributions au génie logiciel et au développement et déploiement de normes internationales en génie logiciel pour de très petites organisations.

Thèse soutenue le 28 octobre 2009
devant le jury composé de :

Robert Dupuis
Professeur, UQAM Québec, Rapporteur

Naji Habra
Professeur, FUNDP Namur, Rapporteur

Jean Philippe Babau
Professeur, UBO Brest, Examinateur

Patrick Bosc
Professeur, ENSSAT Lannion, Examinateur

Philippe Saliou
Maître de conférences, UBO Brest, Examinateur

Jacques Tisseau
Professeur, ENIB Brest, Directeur de thèse

Vincent Ribaud
Maître de conférences, UBO Brest, Invité

Contents

List of Figures

List of Tables

Dedication

This thesis is dedicated to Dr. Maurice Cormier and Dr Bertrand Banville, my physics professors at the Collège Militaire Royal de Saint-Jean and the Université de Montréal. They communicated their passion for experimental physics and research to me over twenty years ago when I was their student.

Acknowledgments

I would first like to express my gratitude to Dr. Jacques Tisseau, Director of the École Nationale d'Ingénieurs de Brest and my thesis supervisor at the Université de Bretagne Occidentale. His counsel during the writing of this thesis was insightful. I also wish to thank thesis examiners, Professor Robert Dupuis of the Université du Québec à Montréal (Canada) and Professor Naji Habra of the Université de Namur (Belgium).

Deserving of my special thanks is Professor Pierre Bourque, a colleague and a friend of ten years' standing. It has been, and continues to be, a pleasure working with him.

I would also like to express my gratitude to the management of the École de technologie supérieure for granting me study leave to write this thesis.

Thank you, Professor Vincent Ribaud Philippe Saliou of the Université de Bretagne Occidentale, for introducing me to Professor Tisseau.

Résumé

Comme le mentionne l'Académie des Sciences nationales « le logiciel n'est pas qu'un produit essentiel du marché, mais, en fait, incarne la fonction de production de l'économie elle-même » (Jorgenson et al. 2006). Les trois exemples suivants illustrent cette situation : d'ici 2010, on prévoit que le téléphone mobile contiendra 20 millions de lignes de code et un fabricant d'automobiles a estimé que ses voitures auront jusqu'à 100 millions de lignes de code (Charrette, 2005). Dans l'industrie aérospatiale, la société Boeing compte dépenser environ 4 milliards de dollars pour le développement des logiciels de son nouvel avion, le modèle 787 (Longtemps, 2008). Le développement de ces logiciels de haute qualité pourrait exiger jusqu'à 20 000 personnes-année d'effort. Même les organismes tels que Boeing ne peuvent développer cette quantité de logiciels à l'interne dans un délai raisonnable. Il est très probable que beaucoup de petites et très petites organisations fourniront des composants à la société Boeing ou à ses fournisseurs principaux qui les intégreront à d'autres composants ou sous-systèmes.

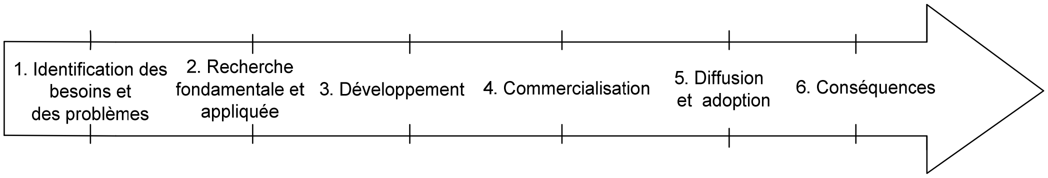
L'auteur a passé une partie significative de sa carrière dans de grandes organisations comme le Ministère de la Défense nationale du Canada, en tant qu'ingénieur des processus pour un fabricant d'un système équipé de missiles pour la défense antiaérienne. Il a œuvré en outre comme conseiller en amélioration des processus pour un fabricant ferroviaire important. Dans ces organisations, l'utilisation de normes pour le développement ou l'entretien des logiciels faisait partie de leur culture. Quand l'auteur a joint, en tant que professeur de génie logiciel, l'École de technologie supérieure (ÉTS) en 2000, il a été confronté avec un nombre d'étudiants de premier cycle en génie logiciel qui travaillaient dans de petites entreprises. Les étudiants n'avaient pratiquement aucune connaissance concrète des normes du génie logiciel. L'auteur a dû apporter des modifications à sa stratégie d'enseignement des normes pour s'adapter au contexte des étudiants, puisque les normes en génie logiciel ont été développées par des professionnels travaillant dans de grandes organisations, pour répondre aux besoins de ces dernières. Les petites organisations n'ont pas l'expertise, ni le budget ni le temps pour adapter de telles normes à leur contexte.

Après avoir assisté à quelques réunions du comité responsable du développement des normes en génie logiciel et en génie des systèmes de l'Organisation internationale de normalisation (ISO), l'auteur a décidé d'aborder ces problèmes. Dans les paragraphes suivants, l'auteur décrit l'approche adoptée pour aider les très petites organisations (TPOs) à améliorer leurs pratiques en génie logiciel en vue d'augmenter leurs performances et leur compétitivité. Les TPOs sont des entreprises, des organismes, des départements ou des projets ayant jusqu'à 25 personnes.

Les TPOs en technologie de l'information (TI) sont très importantes pour l'économie mondiale. Un grand pourcentage des organisations produisant des logiciels dans le monde ont moins de 25 employés. Par exemple en Europe, 85 % des sociétés du secteur des technologies de l'information (TI) ont entre 1 et 10 employés[1]. Au Canada, la région de Montréal comporte près de 80 % de sociétés de moins de 25 employés (Gauthier, 2004). Leurs composants logiciels sont souvent intégrés dans les produits de plus grandes entreprises. Les retards de livraison, le non-respect du budget et un produit parfois de faible qualité menacent la compétitivité des clients et des TPOs. Une approche visant à atténuer ces risques est d'utiliser des fournisseurs ayant instauré des pratiques éprouvées en génie logiciel comme celles documentées dans les normes de l'ISO.

Beaucoup de normes internationales, telles que la norme définissant les processus de cycle de vie de logiciel, ISO/CEI 12207 (ISO 2008d) et de modèles tels que le modèle d'évolution des capacités (SEI 2006) développé par le Software Engineering Institute, ont été développés pour documenter des pratiques éprouvées. Cependant, ces normes et modèles n'ont pas été conçus en ayant les TPOs à l'esprit. Il est difficile les appliquer dans de telles organisations. Un groupe de travail de l'ISO/IEC JTC1/SC7[2] a été établi, le groupe de travail 24 (WG24) pour aborder ces difficultés et développer des normes de génie logiciel et des rapports techniques conçus spécifiquement en fonction des besoins des TPOs. L'auteur de cette thèse a été nommé, par l'ISO en 2005, éditeur de ce nouveau groupe de travail.

Cette thèse utilise le modèle à six (6) phases du processus de développement d'innovations de Rogers (Rogers, 2003) illustré à la figure ci-dessous, pour décrire la démarche et les contributions de l'auteur.


1. Identification des besoins et des problèmes
2. Recherche fondamentale et appliquée
3. Développement
4. Commercialisation
5. Diffusion et adoption
6. Conséquences

Phases du processus d'innovation (adapté de Rogers 2003)

La phase 1 du processus d'innovation, intitulée identification des besoins et problèmes, a débuté en 2004 en Australie, lors de la réunion du comité de normalisation SC7 de l'ISO responsable des normes en génie logiciel. À cette réunion, le Canada a souligné les problèmes des petites organisations requérant l'adaptation des normes à leur taille et à leur niveau de maturité. Des participants à cette réunion ont mentionné que les normes sont, pour les TPOs, trop volumineuses et compliquées. Les normes actuelles d'ingénierie logicielle ciblent ou semble cibler

[1] http://www.esi.es/en/main/iitmark.html

[2] ISO/IEC JTC 1/SC7 signifie International Organization for Standardization/ International Electrotechnical Commission Joint Technical Committee 1/Sub Committee 7. Ce comité est responsable du développement et de l'amélioration des normes en génie logiciel et en génie des systèmes.

les grandes organisations. Une réunion fut organisée avec des représentants de cinq instances nationales (Australie, Canada, République tchèque, Afrique du Sud et Thaïlande) où un consensus fut atteint quant aux objectifs généraux (ISO 2004c) d'un futur groupe de travail :

- Rendre les normes d'ingénierie logicielle de l'ISO davantage accessibles aux TPOs.
- Fournir aux TPOs de la documentation demandant un effort minimum d'adaptation.
- Fournir aux TPOs une documentation harmonisée, intégrant les normes existantes:
    - Normes de processus
    - Normes décrivant les produits du travail et les livrables
    - Normes portant sur l'évaluation et la qualité

Il fut également décidé de créer un groupe d'intérêt spécial (SIG) en vue d'explorer ces objectifs, de mieux articuler les priorités et d'élaborer un plan de travail.

La phase 2 du processus d'innovation, intitulée recherche fondamentale et appliquée, a été amorcée par l'invitation d'un groupe d'experts à l'Institut de standardisation industrielle de Thaïlande (TASI), pour faire progresser les travaux définis lors de la réunion en Australie. Un des sujets de discussion fut de définir clairement la taille des TPOs visées par les travaux de ce groupe. Il a été convenu qu'une TPO est définie comme suit : une entreprise, une organisation, un département ou un projet ayant jusqu'à 25 personnes.

L'auteur a proposé, lors de cette réunion, de développer et de conduire un sondage international des TPOs. Ce sondage, traduit dans 9 langues, a été mené sur l'utilisation des normes par les TPOs. Des données ont été rassemblées pour valider la liste de problèmes et de besoins documentés par les membres du groupe de travail. Plus de 435 réponses ont été reçues de 32 pays. Les résultats du sondage ont été utilisés par le groupe de travail pour définir les exigences qui seront employées aux fins de développer les normes pour les TPOs. En 2005, l'ISO a approuvé la formation d'un groupe de travail (*Working Group*), le WG24.

Puisque le WG24 désirait préparer une première norme le plus rapidement possible, il a effectué une recherche des normes ou des modèles existants pouvant être adaptés aux besoins des TPOs. Une norme nationale mexicaine, la norme MoProsoft (NMX 2005), développée pour les petites et moyennes entreprises mexicaines, a été choisie pour atteindre cet objectif.

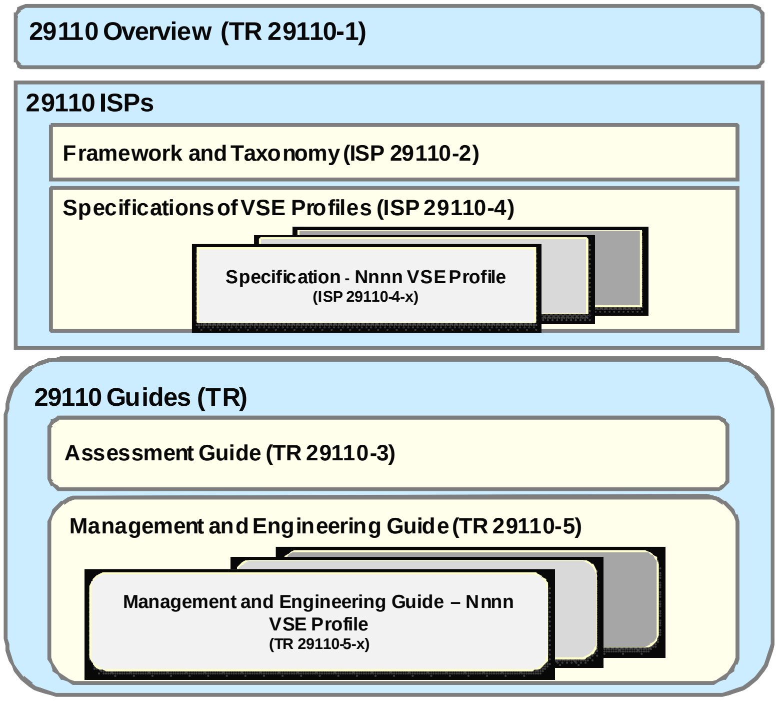
À la phase 3 du processus d'innovation intitulée développement, le WG24 a développé un ensemble de normes et de rapports techniques. Le développement a été réalisé en deux étapes : à l'étape 1, le groupe de travail a choisi, à partir de la norme nationale mexicaine, un sous-ensemble de processus et de produits de travail (*Work Products*) applicables aux TPOs. À l'étape 2, le groupe a adapté ce sous-ensemble aux besoins des TPOs et a développé des normes et des rapports techniques. Chaque document développé par le WG24 vise une clientèle spécifique.

De plus, pour guider les TPOs à la mise en œuvre des pratiques de génie logiciel adaptées à leurs besoins et à leur croissance, le WG24 a développé un parcours (*roadmap*) composé de quatre (4) étapes ou profils :

- L'étape 1 s'adresse aux TPOs en démarrage (*Start-up*) et aux TPOs qui développent des projets de 6 personnes-mois ou moins.
- L'étape 2 s'adresse aux TPOs qui n'exécutent qu'un seul projet logiciel à la fois.
- L'étape 3 s'adresse aux TPOs qui exécutent plusieurs projets logiciels à la fois.
- L'étape 4 s'adresse aux TPOs désirant améliorer notablement la gestion de leurs affaires (Business Management, Portfolio Management).

La figure ci-dessous illustre les documents développés à ce jour pour les TPOs exécutant un seul projet à la fois :

- Le document intitulé *Overview* est un rapport technique ISO présentant les concepts nécessaires à la compréhension de l'ensemble des documents liés à la norme, c'est-à-dire les documents intitulés *Framework and Taxonomy* et *Specification of Profile*, ainsi que les rapports techniques intitulés *Assessment Guide* et *Management and Engineering Guide*. Le document *Overview* est destiné à une clientèle désirant comprendre la norme: les TPOs, les donneurs d'ordre, les évaluateurs, les auteurs d'un profil, les développeurs d'outils, les consultants, etc.
- Le document intitulé *Framework and Taxonomy* est une norme ISO. Ce document explique le concept et la structure des profils ainsi que la terminologie spécifique à cette norme.
- Le document intitulé *Assessment Guide* est un rapport technique ISO. Il présente des guides pour développer une méthode d'évaluation ou pour effectuer l'évaluation d'un profil.
- Le document intitulé *Specification* of Profile est une norme ISO. Il énumère les éléments de normes telles que la ISO/CEI 12207 (ISO 2008d) et la ISO/CEI 15289 (ISO 2006a), qui sont utilisées pour la description d'un profil spécifique.
- Le document intitulé *Management and Engineering Guide* est un rapport technique ISO. Il fournit des guides d'application en matière de gestion et d'ingénierie pour un profil spécifique.


29110 Overview (TR 29110-1)
29110 ISPs
Framework and Taxonomy (ISP 29110-2)
Specifications of VSE Profiles (ISP 29110-4)
Specification - Nnnn VSE Profile
(ISP 29110-4-x)
29110 Guides (TR)
Assessment Guide (TR 29110-3)
Management and Engineering Guide (TR 29110-5)
Management and Engineering Guide – Nnnn VSE Profile
(TR 29110-5-x)

Vue d'ensemble des documents ISO pour les TPOs

À la phase 4 du processus d'innovation intitulée commercialisation, le WG24 et l'ISO ont coordonné des cycles de révision des documents et ont rassemblé des centaines de commentaires des délégations nationales. Le WG24 a analysé chaque commentaire et a produit une version révisée des documents. L'ensemble de ces documents devrait être approuvé par les organismes nationaux de normalisation et publié par l'ISO en 2010.

Même si le groupe de travail a créé un guide de gestion et d'ingénierie, la plupart des TPOs ne possèdent pas l'expertise pour transformer ce guide en un processus utilisable et utile. L'auteur a donc proposé aux délégués du groupe 24, lors de sa réunion à Moscou en 2007, le développement de matériel utilisable 'tel quel' par les TPOs. À la phase 5 du processus d'innovation intitulée diffusion et adoption, l'auteur a encadré le développement d'un ensemble de documents intitulé trousse de déploiement (*Deployment Package*), à partir du guide de gestion et d'ingénierie (ISO/IEC TR29110-5.1 VSEP Basic Profile- Management and Engineering Guide), pour faciliter l'adoption et l'implémentation des pratiques logicielles dans les TPOs. Une trousse de déploiement (TD) est un ensemble d'artefacts visant à faciliter et à accélérer l'implantation de la norme ISO dans les TPOs en leur donnant des processus prêts à être utilisés. Par exemple : processus documentés comportant les activités, rôles, intrants, extrants, listes de vérification, gabarits, exemples et outils de support. Pour le profil s'adressant aux TPOs qui n'exécutent qu'un seul projet logiciel à la fois, les membres du groupe de travail ont élaboré bénévolement les trousses de déploiement énumérées au tableau suivant.

Nom de la trousse de déploiement	Pays responsable
Analyse des exigences	Belgique, Canada
Architecture et conception détaillée	Canada
Construction (i.e. codage et tests unitaires)	Mexique
Intégration et tests	Colombie
Vérification et validation	Pérou
Gestion des versions	Thaïlande
Gestion de projets	Irlande
Livraison du produit	Thaïlande
Auto-évaluation	Finlande
Conduite de projets pilotes	Canada, Uruguay

Liste des trousses de déploiement

Ces trousses de déploiement forment un ensemble cohérent, permettant la mise en place de la norme, pièce par pièce, pour répondre aux besoins des TPOs et à leurs capacités spécifiques à implémenter et utiliser de nouvelles pratiques. Par exemple, une TPO qui éprouve des difficultés à gérer les versions de ses documents et logiciels appliquera les pratiques de la trousse 'version control'. En ce moment, les trousses ne sont disponibles qu'en anglais puisqu'elles sont développées et révisées par les membres du groupe 24. Par la suite, elles seront traduites par le délégué d'un pays pour satisfaire les besoins des TPOs de son pays.

Lors de la réunion du WG24 à Mexico en 2008, l'auteur a proposé la création d'un réseau international de support aux TPOs. Comme plusieurs membres du WG24 travaillent dans des centres de transfert technologique, il a été proposé d'instaurer un réseau de collaboration entre ces centres. Chaque centre a la responsabilité de déployer les trousses et d'offrir d'autres produits et services en tenant compte des particularités nationales. Les principaux objectifs de ce réseau sont d'accélérer le déploiement de la norme de l'ISO et des guides pour les TPOs, l'accélération du développement et l'application des guides et des trousses de déploiement. Un document décrivant les modalités de collaboration a été rédigé et signé entre chacun des membres du réseau et l'ÉTS. Les participants au réseau sont :

- Belgique - Centre d'Excellence en Technologies de l'Information et de la Communication (CETIC)
- Colombie - Parquesoft Foundation
- Finlande - Université de technologie de Tampere, Pori
- France - Université de Bretagne Occidentale
- Hong Kong - Université Polytechnique
- Irlande - Lero, The Irish Software Engineering Research Centre
- Luxembourg - Centre de Recherche Public Henri Tudor
- Thaïlande – Federation of Thai Industries

Afin de s'assurer que les normes, documents techniques et les trousses de déploiement satisferont les besoins des TPOs, des projets pilotes seront réalisés en 2009.

La Thaïlande annoncé, lors de la réunion du groupe 24 en Inde en mai 2009, un réseau de collaboration dont elle sera le chef de file (*Regional Hub*). Ce réseau est composé des 10 pays de l'organization ASEAN[3]: Thaïlande (2000[4]), Cambodge (100), Laos (200), Myanmar (200), Vietnam (500), Malaisie (1200), Singapour (1000), Indonésie (1500), Philippines (2500) et Brunei (500). L'Universidad Nacional Autónoma de México (UNAM) a également manifesté son intérêt à devenir membre du réseau. D'autres centres de transfert ainsi que d'autres universités seront contactés et invités à se joindre au réseau en 2009 (i.e. Corée, Japon, Chine, Afrique du Sud).

Lors de la réunion du groupe de travail de l'ISO en Inde en mai 2009, l'auteur a proposé aux 21 délégués, représentant 11 pays, la création d'un groupe d'intérêt sur l'éducation (*Education Interest Group*). L'objectif d'un tel groupe est de développer un ensemble de cours pour les étudiants de premier et second cycle en informatique ou en génie logiciel/informatique, de telle sorte qu'ils apprennent et appliquent les normes ISO pour les TPOs durant leurs études plutôt que d'attendre leur arrivée en industrie. Les cours développés seront, pour les universités, similaires aux trousses de déploiement développées pour les TPOs. Ces trousses d'enseignement comporteront les éléments suivants : un plan de cours, du matériel de présentation, des exercices, des études de cas et des suggestions de lecture. Un premier jeu de six cours a été proposé aux délégués :

- Cours 1 - Introduction aux normes ISO/CEI en génie logiciel.
- Cours 2 - Introduction aux normes, rapports techniques ISO/CEI 29110 et aux trousses de déploiement pour les TPOs.
- Cours 3 - Développement d'un processus d'ingénierie de logiciels (Process Engineering) utilisant le Rapport technique ISO/CEI 29110 Partie 5 – Guide d'ingénierie et de gestion.
- Cours 4 - Développement de logiciels utilisant le Rapport technique ISO/CEI 29110 Partie 5 – Guide d'ingénierie et de gestion.
- Cours 5 - Évaluation de la conformité des processus de développement de logiciels à la norme ISO/CEI 29110.
- Cours 6 – Conduite de projets pilote pour implanter la norme ISO/ CEI 29110 dans une TPO.

L'auteur a développé un gabarit de trousse d'enseignement qui sera utilisé pour le développement et la documentation de chaque cours. Les délégués de quatre pays ont offert de développer les cours suivants :

- Irlande- Introduction aux normes ISO/CEI en génie logiciel.
- Canada - Introduction aux normes, rapports techniques ISO/CEI 29110 et aux trousses de déploiement pour les TPOs.

[3] Association of Southeast Asian Nations.
[4] Ce nombre indique le nombre de TPOs qui pourraient bénéficier des travaux du WG24.

- République Tchèque - Développement de logiciels utilisant le Rapport technique ISO/CEI 29110 Partie 5 – Guide d'ingénierie et de gestion.
- Thaïlande - Conduite d'un projet pilote pour implanter la norme ISO/ CEI 29110 dans une TPO.

Lors de la prochaine réunion du groupe de l'ISO au Pérou en novembre 2009, les cours seront présentés aux délégués. Ils seront ensuite disponibles gratuitement sur Internet pour les professeurs des universités.

Dans le but de publiciser, à l'extérieur du groupe de travail 24 de l'ISO, le travail effectué par celui-ci, l'auteur a développé un site Internet en français et un en anglais. Ce site est hébergé à l'ÉTS et s'avère une extension du site professionnel de l'auteur. Ce site comporte les informations suivantes ;

- Une page d'accueil expliquant le mandat du groupe ISO.
- Une page énumérant les membres du groupe de travail afin de bien illustrer la participation de plusieurs pays au groupe ISO.
- Une page décrivant le réseau de support aux TPOs.
- Une page dépeignant les trousses de déploiement et qui permet de télécharger ces trousses (les trousses ne sont disponibles qu'en anglais présentement).
- Une page dépeignant les trousses d'enseignement (les trousses ne sont disponibles qu'en anglais présentement).
- Une page énumérant les publications et communications effectuées par les membres du groupe ISO. La plupart des publications peuvent être téléchargées.

Au dernier chapitre de cette thèse, l'auteur présente les travaux futurs et une conclusion. En ce qui a trait aux travaux futurs, les sujets suivants sont discutés : l'application des concepts des profils et des trousses de déploiement au domaine de l'ingénierie de systèmes pour le développement des produits par des TPOs; la mesure de l'adoption, par les TPOs dans le monde entier, des normes de l'ISO et des trousses de déploiement développées par le WG24; l'établissement d'un centre de transfert technologique, pour les TPOs québécoises à l'École de technologie supérieure (ÉTS); le développement de profils pour les développeurs de logiciels critiques et pour les développeurs de logiciels scientifiques; le développement de modules appelés '*plug-in*' pour faciliter et accélérer l'implémentation des normes ISO et des trousses de déploiement. Finalement, au chapitre de la phase 6 du processus d'innovation, l'auteur propose des pistes visant l'étude des conséquences positives et négatives résultant de la publication de la norme ISO, pour les TPOs.

Summary

According to the National Academy of Science, “software is not merely an essential market commodity, but, in fact, embodies the economy’s production function itself” (Jorgenson et al. 2006). The following three examples illustrate this statement: by 2010, it is estimated that the cellular phone will contain 20 million lines of code; an automobile manufacturer estimates that by then its cars will have up to 100 million lines of code (Charette 2005); and, in the aerospace industry, Boeing is expecting to spend about $4 billion to develop the software for its 787 airplane (Long 2008). The development of this high-quality software may require up to 20,000 staff-years. Even an organization like Boeing cannot develop this quantity of software in-house within a reasonable time frame. It is very likely, therefore, that many small and very small enterprises will be providing components to Boeing or to its main suppliers, who will then integrate them into other components or subsystems.

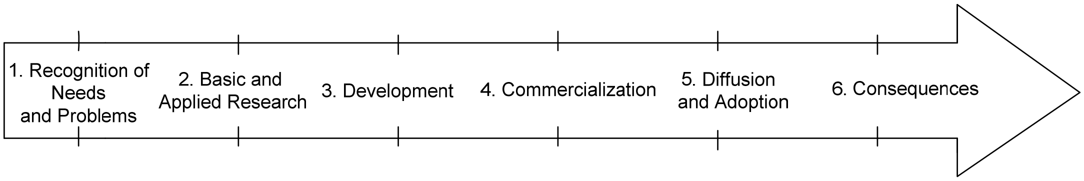
I, the author of this thesis, have spent a significant portion of my career in large organizations, such as the Department of National Defence of Canada, as a process engineer for a manufacturer of a missile air-defense system and as a consultant for a major railway manufacturer. In these organizations, the utilization of software engineering standards for the development and maintenance of their software was part of their culture. When I joined the staff of the École de technologie supérieure (ÉTS) in 2000 as a software engineering professor, I was confronted with a large number of undergraduate and professional graduate students who were working for small and very small organizations which had almost no practical knowledge and sometimes no knowledge at all, of software engineering standards. As a result, I had to make many modifications to my teaching strategy in order to adapt my software engineering courses to the students’ context, since the software engineering standards I had been using for teaching were developed by professionals working in large organizations for the development of large software projects in other large organizations. Small organizations do not have the expertise, the budget, or the time required to adapt such standards to their context.

After attending a few meetings of the committee responsible for the development of software engineering standards of the International Organization for Standardization (ISO), I decided to address the following issues: identify the problems small software organizations have with the application of software engineering standards to their development projects, and identify and develop ways to transition software engineering best practices, documented in standards, to small organizations. Many international standards and models, such as the standard defining software life cycle processes, ISO/IEC 12207 (ISO 2008), or the Software Engineering Institute’s

Capability Maturity Model Integration (SEI 2006), have been developed to capture proven engineering practices. However, these standards and models were not designed with Very Small Entities (VSEs) in mind. VSEs are defined as enterprises, departments, or projects having fewer than 25 people (ISO 2005b). Standards and models are difficult to apply in such settings. An ISO/IEC JTC1/SC7[5] Working Group has been established, Working Group 24 (WG24), to address these difficulties by developing software engineering standards and technical reports which are specifically tailored to the needs of VSEs. The ISO appointed me to be Project Editor for this new Working Group in 2005.

Information Technology VSEs are very important to the world economy. A large percentage of software entities worldwide have fewer than 25 employees. In Europe, for instance, 85% of Information Technology (IT) sector companies have 1 to 10 employees[6]. A survey of the Montreal, Canada, area has revealed that 78% of software development enterprises have fewer than 25 employees and 50% have fewer than 10 (Gauthier 2004). Their software components are often integrated into the products of larger entities. Failure to deliver a quality product on time and within budget threatens the competitiveness of VSEs and impacts customers. One way to mitigate these risks is for the suppliers of a product chain to put in place proven software engineering practices, such as those documented in standards.

The purpose of the thesis is to present my contributions to software engineering and to the development, deployment, and utilization of international software engineering (SE) standards, specifically those for VSEs, and both in Canada and abroad. To achieve this, I use the 6-phase model of the innovation process developed by Rogers (Rogers 2003), as illustrated in the figure below.


1. Recognition of Needs and Problems
2. Basic and Applied Research
3. Development
4. Commercialization
5. Diffusion and Adoption
6. Consequences

Phases of the Innovation Process (adapted from Rogers 2003)

Phase 1 involves the recognition of needs and problems. It was begun at a 2004 meeting in Australia of an ISO subcommittee (SC7) mandated to develop international software engineering standards. During the meeting, the members of the Canadian delegation were told that SC7 standards were "too bulky" and "too complicated" to use. A meeting of interested parties was

[5] ISO/IEC JTC 1/SC7 stands for the International Organization for Standardization/ International Electrotechnical Commission Joint Technical Committee 1/Sub Committee 7, which is in charge of the development and maintenance of software and systems engineering standards.

[6] http://www.esi.es/en/main/iitmark.html

held to discuss the issues surrounding needs and problems. The participants decided to create an informal one-year special interest group (SIG) to explore them, better articulate the target, devise the plan to get there, and prepare and table a proposal at the next SC7 Plenary meeting in Finland in 2005 (ISO 2004c).

In phase 2 of the 6-phase model, the research phase, the Thai Industrial Standards Institute (TASI) invited a group of experts to Bangkok in 2005 to explore the research issues. At that meeting, I proposed to the working group that a survey be conducted of VSEs worldwide. The international survey was developed and translated into 9 languages. It was conducted among VSEs and concerned their utilization of standards. The data collected were used to validate the list of needs and problems that had been documented by the working group during the previous phase. Over 435 responses were received from 32 countries. The working group used the results to define the requirements that would form the basis for the standards to be developed for VSEs. It was then, in 2005, that the ISO established WG24 to address the issues, and I was appointed editor of this new group. Since WG24's objective was to prepare an initial set of standards as quickly as possible, the group analyzed international reference standards and models that could help develop a subset of ISO standards for VSEs. WG24 began a search for existing standards and models that could be tailored or adapted to the needs of VSEs. Moprosoft, a Mexican National Standard (NMX 2005), developed to assist small and medium-sized Mexican enterprises had been selected to achieve this objective.

In phase 3, the development phase, WG24 developed a set of standards and technical reports. The approach consisted of two steps: in step 1, the processes and work products applicable to VSEs were selected from the Mexican National Standard; and in step 2, this subset was tailored to fit the needs of VSEs. The group had to use the concept of the ISO profile (ISP – International Standardized Profile) to develop the new standards. A profile is a kind of matrix which precisely differentiates all the elements that are taken from existing standards from those that are not. The decision was made to develop a four-phase roadmap:

- Phase 1 targets VSEs typically developing 6 person-month projects or start-up VSEs;
- Phase 2 targets VSEs developing only one project at a time;
- Phase 3 targets VSEs developing more than one project at a time;
- Phase 4 targets VSEs wishing to put in place business management practices and portfolio management practices.

The following set of documents, targeted by audience, has been developed by the group. These are (ISO2009a): Part 1: Overview; Part 2: Framework and Taxonomy; Part 3: Assessment Guide;

Part 4: Profile Specifications; and Part 5: Management and Engineering Guides. Parts 1 and 5 target VSEs, Part 3 targets Assessors and VSEs, and Parts 2 and 4 target standards producers, tool vendors, and methodology vendors. If a new profile is needed, Parts 4 and 5 can be developed without impacting existing documents, and they become Part 4-x and Part 5-x respectively through the ISO/IEC process (ISO 2009a).

In phase 4, on commercialization, the ISO coordinated a series of review cycles and collected hundreds of comments from national bodies. WG24 analyzed all the comments and produced a revised version of the set of documents to be sent to national bodies for the next round of review. The set of documents should be approved by those national bodies and published by the ISO in 2010.

To address the fifth phase, on diffusion and adoption, I proposed, at the Moscow meeting of the WG24 in 2007, the development of a set of documents, entitled Deployment Packages, to facilitate the adoption and implementation of a set of ISO standard practices for VSEs. A deployment package is a set of artifacts developed to facilitate the implementation of a set of practices, of the selected framework, in a VSE. The elements of a typical deployment package are: a detailed description of processes, activities, tasks, roles, and products; a template; a checklist; an example; a reference and a mapping to standards and models; and a list of tools. The following deployment packages have been developed: Requirements Analysis, Architecture and Detailed Design, Construction and Unit Testing, Integration and Test, Verification and Validation, Version Control, Project Management, Product Delivery, and Self-Assessment. These documents will be validated through a series of pilot projects in VSEs around the world.

At the Mexico meeting of WG24 in 2008, I proposed the establishment of an international network of collaborators whose aim is to promote, facilitate, and develop collaborative activities between institutions to improve VSE capabilities. The main activities of the network are to accelerate the deployment of Standards and Guides for VSEs and to accelerate the development and application of Guides and Deployment Packages. The following organizations have signed a collaboration agreement with the École de technologie supérieure (ÉTS): the Centre d'Excellence en Technologies de l'Information et de la Communication (Belgium), the Parquesoft Foundation (Columbia), the Tampere University of Technology (Finland), the Université de Bretagne Occidentale (France), the Irish Software Engineering Research Centre (Ireland), the Public Research Centre Henri Tudor (Luxembourg), and the Institute of Software Promotion for Industries (Thailand).

At the 2009 meeting of the working group in India, I proposed the establishment of an education interest group, the main objective of which is to develop a set of courses for software undergraduate and graduate students to enable students to learn about the ISO standards for

VSEs before they graduate. The courses developed for academia will consist of education deployment packages similar to the development packages for VSEs. The objective of the educational deployment packages is to facilitate and accelerate the teaching of the new ISO standards in educational institutions by providing them with readily usable teaching material, such as course plans, presentation material, exercises, case studies, and reading lists.

In the last chapter of this thesis, Conclusion and Future Work, I address the last phase, concerning the consequences of an innovation. With regard to future work, the following topics are discussed: the application of the concepts of profiles and deployment packages to the domain of systems engineering for the development of products by VSEs; the measurement of the adoption, by VSEs worldwide, of the ISO standards and the deployment packages; the establishment of a technology transfer center for VSEs at the ÉTS; the development of profiles for critical software development and for scientific software development; the development of software plug-ins to facilitate and accelerate the utilization of the deployment packages; and the development, by the education interest group, of courses and course material for teaching the new standards in universities worldwide. With regard to the consequences of innovation, the author presents the positive and negatives consequences of the publication of standards for VSEs.

Abbreviations and Acronyms

ASEC	Applied Software Engineering Center
CMM[7]	Capability Maturity Model[8]
CMMI	Capability Maturity Model Integration
CMR	Collège Militaire Royal (St-Jean)
CRIM	Centre de recherche informatique de Montréal
DND	Department of National Defence (Canada)
DoD	Department of Defence (USA)
DP	Deployment Package
ÉTS	École de technologie supérieure
IEEE	Institute of Electrical and Electronics Engineers
IS	International Standard
ISO[9]	International Organization for Standardization
ISO/IEC JTC1 SC7	International Organization for Standardization/ International Electrotechnical Commission Joint Technical Committee 1 Sub-Committee 7
NATO	North Atlantic Treaty Organization
PAL	Process Asset Library
PMI	Project Management Institute
SE	Software Engineering
SESC	Software Engineering Standards Committee
SEI	Software Engineering Institute
SME	Small and Medium-sized Enterprise
SPI	Software Process Improvement
SPICE	Software Process Improvement and Capability Determination
STEP	Software Technology and Practice
SWEBOK	Software Engineering Body of Knowledge
TCSE	Technical Council on Software Engineering
TR	Technical Report
VSE	Very Small Entity
VSEs	Very Small Entities
WG	Working Group

[7] CMMI, CMM are registered with the US Patents and Trademarks Office by Carnegie Mellon University.

[8] Capability Maturity Model Integration is a service mark of Carnegie Mellon University.

[9] ISO is not an acronym; it stands for the Greek word 'ISOS'.

Note on Access to Contents

I hereby declare that, with effect from the date on which the thesis is deposited in the Library of the Université de Bretagne Occidentale (UBO), I permit the Librarian of the University to allow the thesis to be copied in whole or in part without reference to me on the understanding that such authority applies to the provision of single copies made for study purposes or for inclusion within the stock of another library. This restriction does not apply to the Université de Bretagne Occidentale Library Thesis Service (which is permitted to copy the thesis on demand for loan or sale under the terms of a separate agreement), nor to the copying or publication of the title and abstract of the thesis.

IT IS A CONDITION OF USE OF THIS THESIS THAT ANYONE WHO CONSULTS IT MUST RECOGNIZE THAT THE COPYRIGHT RESTS WITH THE AUTHOR AND THAT NO QUOTATION FROM THE THESIS AND NO INFORMATION DERIVED FROM IT MAY BE PUBLISHED UNLESS THE SOURCE IS PROPERLY ACKNOWLEDGED.

**The copyright of this thesis
is vested in the author**

Chapter 1

Introduction

1 Chapter 1 – Introduction

This thesis is being submitted to the Université de Bretagne Occidentale to satisfy the requirements for a Doctorate. Its purpose is to demonstrate that the author's contributions have had an impact in software engineering (SE) and to the development, deployment, and utilization of international SE standards, specifically those for Very Small Entities (VSEs) in Canada and internationally. VSEs are defined as enterprises, departments, or projects having fewer than 25 people (ISO 2005b).

The impact of the author's published contributions will be shown in the thesis:

- by presenting a large number of his published works and contributions covering a 29-year period;
- by describing his contributions to the military, to industry and academia;
- by describing his contributions to the establishment of a software engineering center in Montréal;
- by describing his contributions as the editor of Working Group 24 of ISO/IEC JTC1 SC7[10], and as the Canadian delegate to that group, which is mandated to develop International Standards (ISs) and International Technical Reports (TRs) for VSEs;
- by describing the wide acceptance of the draft ISs and TRs developed by Working Group 24;
- by describing his contributions, beyond his role as the editor of Working Group 24, to accelerate and facilitate the diffusion of ISs and TRs to VSEs worldwide, and their utilization by those entities;
- by describing the infrastructure developed to accelerate and facilitate the deployment and utilization of future standards by VSEs worldwide;
- by describing his contributions to the establishment of a technology transfer center for VSEs at the École de technologie supérieure (Montréal);
- by describing his contributions, as the co-editor of Working Group 20 of ISO/IEC JTC1 SC7, and as the Canadian delegate to that group;

[10] International Organization for Standardization/International Electrotechnical Commission Joint Technical Committee 1, Sub-Committee 7.

- by describing the research performed by SE graduate students under his supervision or co-supervision.

In the next sections of this chapter, the author presents an overview of his contributions to SE in Canada and internationally.

1.1 Historical perspectives and accomplishments

This section describes the accomplishments of the author to enable the reader to better understand his involvement in the development of international standards for Very Small Entities (VSEs) and his motivation for that involvement. This description is presented in two parts: his accomplishments between 1976 and 2000, and the period between 2000 and the present. A more detailed description can be found in Appendix A of this thesis.

1.1.1 Accomplishments of the author between 1976 and 2000

Accomplishments at the Collège militaire Royal de Saint-Jean

Between 1978 and 1991, I was a professor at the Collège militaire Royal de Saint-Jean (CMR), a military school associated with the Department of National Defence. While there, I published papers on the application of microprocessors, mainly in military applications (Laporte 1986, Laporte et al. 1986a, Laporte et al. 1986b, Laporte et al. 1988, Laporte et al. 1989). As a project manager, I was tasked in 1988 by the dean of Science and Engineering to lead the development of a graduate program in SE for the Department of National Defence. The first students in this SE Master's degree program graduated in 1992.

I was appointed project manager in June 1988 by the dean of Science and Engineering for the establishment of a Defence Software Engineering Center. A steering committee was established, mainly composed of senior executives of large defence and aerospace contractors, such as Bombardier Aerospace, Oerlikon Aerospace, CAE, SPAR Aerospace, and Lockheed Martin (then called Paramax) in the Montréal area. A $250,000 feasibility study was sponsored by 13 defence and aerospace companies, along with the Québec and federal governments and the participation of the CMR. This led to the establishment of the Applied Software Engineering Center (ASEC) of the Computer Research Institute of Montréal (CRIM) (Laporte 1995, Laporte 1996).

Development of Engineering Processes at Oerlikon Aerospace

From 1992 to 1999, I worked at Oerlikon Contraves as a senior analyst responsible for coordinating the development and deployment of software and systems engineering processes and management processes. In Appendix A, I describe in detail the steps taken by Oerlikon Aerospace to assess and improve its engineering processes. This description will help the reader better understand the challenges associated with the development and deployment of engineering processes in an organization. My objective is to illustrate how the new international standard and associated guide, presented in the next chapter, can help VSEs without the need to undertake the expensive and lengthy tasks an organization such as Oerlikon Aerospace must perform. I led the software engineering, systems engineering, and project management process improvement activities at Oerlikon Aerospace between 1992 and 1999 (Laporte et al. 1993, Laporte et al. 1996, Laporte et al. 1997, Laporte et al. 1998a, Laporte 1998b, Laporte et al. 1998c, Laporte et al. 1998d, Laporte et al. 1999a, Laporte et al. 1999b, Laporte et al. 1999c).

Development of Engineering Processes for Bombardier Transportation

In 1999, I became a consultant in software process improvement at the software Centre of Competence (CoC) of Bombardier Transportation (BT). BT established the CoC to support the various divisions in reducing technical risks and quality deficiency costs, and in order to help continuously improve the reliability of BT products. The major role of this CoC is to bridge the gap between the increasing demand for new functionalities and cost reduction, while at the same time increasing the maturity level of Bombardier's SE capacity. As an example, the CoC has led the development of integrated SE processes (BES), a set of SE Roles and Responsibilities (Bourque et al. 2004, Laporte et al. 2004a, Laporte et al. 2007b, Laporte et al. 2007c, Laporte 2007e), a set of Peer Reviews ranging from informal desk check reviews, to walkthroughs and inspections (IEEE 1028).

Miscellaneous contributions

I co-founded and chaired the Montréal Software Process Improvement Network (SPIN)[11] for many years.

1.1.2 Accomplishments of the author between 2000 and the present

In 2000, I was invited to join the department of Electrical Engineering of the École de technologie supérieure (ÉTS). At that time, the ÉTS was already offering a graduate program, in collaboration with UQAM and the INRS (Institut National de Recherche Scientifique en Telecommunications, another branch of the Université du Québec) in SE and was preparing the launch of an undergraduate SE program.

In this section, the courses I developed and taught are briefly presented, showing the changes I made to them in response to the challenges and needs of VSEs. Finally, the projects of graduate SE students related to VSEs are briefly described.

[11] SPINs are sponsored by the Software Engineering Institute of Carnegie Mellon University.

Graduate Courses in Software Engineering

I have spent a significant portion of my career in large organizations, such as the Department of National Defense of Canada as a process engineer for a manufacturer of a missile air-defense system, and a major railway manufacturer as a consultant. In these organizations, the utilization of SE standards for the development and maintenance of their software was part of their culture. When I joined the ÉTS in 2000 as a software engineering professor, I found that a large number of the undergraduate and professional graduate students were working in small and very small organizations, where there was almost no practical knowledge and sometimes no knowledge at all, about SE standards. I had to modify my teaching strategy considerably in order to adapt my SE courses to the context of the students. This was because the SE standards I was using for teaching were developed by professionals working in large organizations for the development of large software projects in other large organizations. Small organizations do not have the expertise, the budget, or the time required to adapt such standards to their context.

I was asked to develop and teach a graduate SE course titled Case Study. The course was designed to teach students about software process improvement. Initially, they study a real industrial case study, in which the findings and recommendations resulting from a formal assessment of the set of SE processes of an organization using the Software Capability Maturity Model (CMM) (Paulk et al. 1993) are described.

The students are then asked to develop, in teams of 3 to 4, a software process improvement plan, a communication plan, an installation plan, and a description of processes based on those findings and recommendations. Each team has to present their documents to the class for discussion. I also taught a graduate course on software quality assurance (SQA). Like the process improvement course, this course is initially rather 'static', but soon the students are asked to perform an intervention in an organization.

Undergraduate Courses in Software Engineering

The ÉTS began offering its SE undergraduate program in 2001. I developed an SQA course for this new program, which covers the Software Quality knowledge area of the SWEBOK Guide (Abran et al. 2004) in depth, and also some elements of Software Configuration Management. The aim of the SQA course, which is mandatory in the SE curriculum, is to ensure that SE students are aware of the importance of SQA and that they understand and are able to manage its theoretical and practical aspects (Laporte et al. 2007d). This includes knowledge of related ISO and IEEE standards, as well as how to use

SQA tools in practice. The course allows students to apply SQA practices across the whole software life cycle.

The SWEBOK Guide was a project of the IEEE Computer Society. I participated as a co-editor at the final review stage of this consensually validated document, which is available free of charge from the IEEE. It has been published as ISO/IEC Technical Report 19759 (ISO 2005) and is also available free of charge, from the ISO.

Miscellaneous contributions

I participated in a project titled Improving Processes for Small Settings as an affiliate member of the SEI, and have been a member of the program committee of many SE conferences in Canada, France, the USA, and Latin America.

1.2 Overview of contributions to software engineering

In this section, the author briefly describes his contributions to SE in Canada and internationally using a model developed to characterize the maturity of a profession in terms of eight components and four developmental or evolutionary stages (Ford et al. 1996).

As illustrated in Figure 1-1, the eight components are:

- Initial professional education
- Accreditation of professional education programs
- Skills development
- Certification
- Licensing
- Professional development
- Code of Ethics
- Professional societies

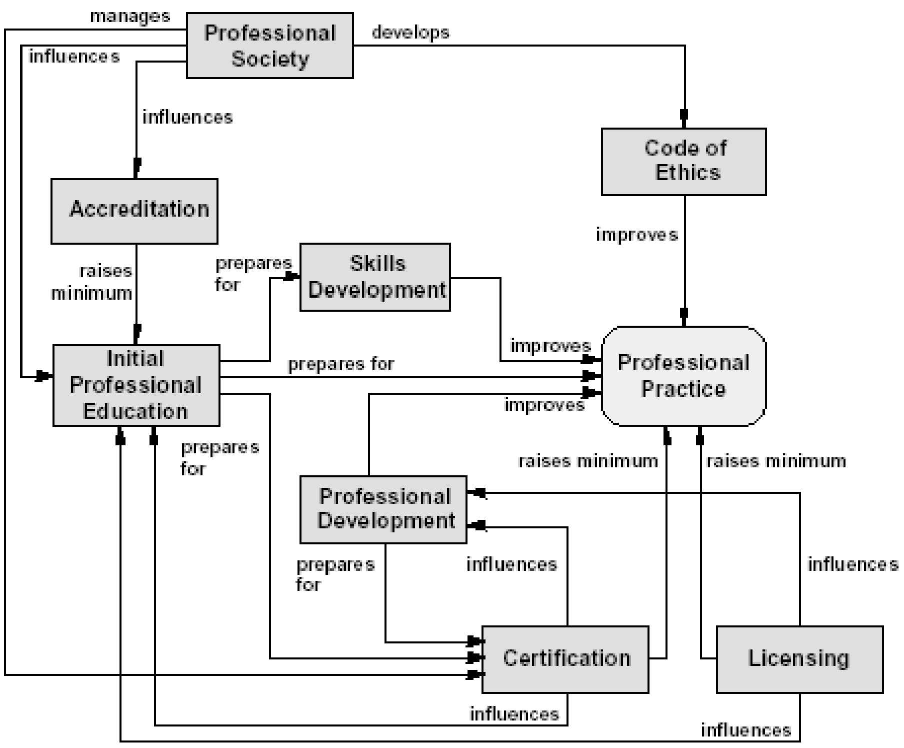


Figure 1-1 Examples of interaction among the components of a profession (Ford et al. 1996)

For this thesis, the model has been adapted to be used as a framework to summarize my contributions to SE in Canada and internationally. The eight components of the Ford model are listed on the left-hand side of Table 1-1, while on the right-hand side are listed my contributions to these components, along with a description of each.

Components of the Ford Model	My Contributions
Initial professional education The education that provides the specialized knowledge needed to practice the profession Programs completed prior to practice of the profession at the baccalaureate and post-baccalaureate levels	**Initial professional education** • Teaching at the baccalaureate level in Canada at the Collège militaire royal de Saint-Jean (CMR) and at the École de technologie supérieure (ÉTS) • Development of a new course in SQA for the SE baccalaureate program at the ÉTS • Development of a new SQA course for the post-baccalaureate program in at the ÉTS • Development of a new course in software process development for the SE post-baccalaureate program at the ÉTS
Accreditation of professional education programs Specialized accreditation bodies that accredit individual programs within a school	**Accreditation of professional education programs** • Contribution at the post-baccalaureate level at Concordia University (Montréal).
Skills development Professionals entering the profession are expected to develop skill in the application of that knowledge, in informal apprenticeships, for example Engineering graduates spend a period of time as "engineers-in-training" prior to taking the licensing examination	**Skills development** • Contribution to professionals entering a profession in industry at the Department of National Defence of Canada, in industry at Oerlikon Aerospace and Bombardier Transportation
Certification Voluntary process administered by the profession or by a professional society, such as the Institute of Electrical and Electronics Engineers (IEEE) or the American Society for Quality Control (ASQC)	**Certification** • Canadian delegate to the ISO SC7 Working Group 20, which developed an ISO standard for the certification of professionals (ISO 2008a)
Licensing Mandatory process administered by a governmental authority for the protection of the public (engineering is a licensed profession in Canada)	**Licensing** • Member of the Professional Engineering Association of the Province of Québec (Ordre des Ingénieurs du Québec – OIQ)
Professional development Activities intended to improve or maintain the currency of the knowledge and skills of a professional after he or she begins professional practice, from reading an article in a professional magazine, for example, to lengthy continuing education or training required for relicensing or recertification	**Professional development** • Teaching at the post-baccalaureate level in Canada at the Université du Québec à Montréal (UQAM) and at the ÉTS. • Co-director of 1 Ph D student at the ÉTS • Director/co-director of 23 SE Master's degree students, and co-director of 1 PhD student at the ÉTS • Publication of papers and communications worldwide • Training of software engineers in Belgium, Canada, France, Germany, Sweden, the USA, and Thailand.
Code of Ethics To ensure that SE practitioners behave in a responsible manner	**Code of Ethics** • Contribution at the baccalaureate and post-baccalaureate levels at the ÉTS by translating, into French, the IEEE Code of Ethics and teaching the Code at the baccalaureate and post-baccalaureate levels

Components of the Ford Model	My Contributions
	- Contribution in teaching at the University of Chiang Mai (Thailand) and coordinating translation into Thai by software engineering undergraduate students
Professional societies Voluntary associations of professionals in societies representing a community of peers, such as the Institute of Electrical and Electronics Engineers (IEEE) or the American Society for Quality Control (ASQC)	**Professional societies** - Contribution to the instigation and chairmanship of the Montréal Software Process Improvement Network (SPIN) - Canadian delegate to ISO/IEC JTC1 SC7 Working Groups 20 and 24 - Member of the IEEE Computer Society, the International Council on Systems Engineering (INCOSE), the Project Management Institute (PMI) - Contribution as a reviewer of papers submitted to professional journals (IEEE Software) - Contribution as a reviewer for the following professional conferences: o Software Engineering Process Group Conference of the SEI o Software Engineering Process Group – Latin America Conference (SEPG-LA) o International Council on Systems Engineering (INCOSE) Symposium

Table 1-1 Components of the Ford model and contributions of the author (adapted from *Ford et al. 1996*)

The Ford model describes four developmental or evolutionary stages of the maturity of a profession. I have adapted these stages to illustrate my contributions. Table 1-2 lists the four evolutionary stages of the model on the left-hand side, and my adaptation on the right-hand side.

Level	Ford Model	The author's adaptation
0	Non-existence: The component does not exist in any form even remotely related to the given profession.	Non-existence: No contribution to this component
1	Ad hoc: Some related form of the component exists, but it is not identified with the given profession.	Small: Some contribution to this component
2	Specific: The component exists and is clearly identified with the given profession.	Moderate: A fair amount of contribution to this component
3	Maturing: The component has existed for many years, during which time it has come under the active stewardship of an appropriate body within the profession and is being continually improved.	Significant: A large amount of contribution to the component

Table 1-2 Four developmental or evolutionary stages (adapted from *Ford et al. 1996*)

Table 1-3 summarizes significant contributions I have made to each component of the framework, both in Canada and abroad, using the four developmental or evolutionary stages. In a next section of this chapter, those contributions are briefly explained. In Appendix A, the contributions are explained in more detail.

Model Components	Canadian contribution	International Contribution
Initial professional education	Significant	Small
Accreditation	Small	None
Skills development	Small	Small
Certification	None	Significant
Licensing	None	None
Professional development	Significant	Significant
Code of Ethics	Small	None
Professional societies	Significant	Significant

Table 1-3 The author's contributions in Canada and internationally (adapted from *Ford et al. 1996*)

1.3 Published works and other contributions

Table 1-4 lists the referred and edited published works and Table 1-5 lists other contributions discussed in this thesis. The tables include International Standards and International Technical Reports, refereed and edited papers, book and book chapters, refereed conference proceedings papers, and public events communications. I am either the main author or a co-author of the papers, books and book chapters, and I am the ISO Project Editor of ISO SC7 Working Group 24, responsible for developing software life cycle standards for VSEs. I was the ISO Co-project Editor and the Canadian delegate to Working Group 20, responsible for the development of the International Technical Report describing the Guide to the Software Engineering Body of Knowledge (SWEBOK) (ISO 2004), and the Canadian delegate to ISO SC7 Working Group 20, responsible for the development of the standards for the certification of software professionals (ISO 2008a).

Topic	Edited Book, Book Chapter, Thesis	Refereed Journal Paper	Edited Journal Paper	Refereed Conference Proceedings Paper
Software and Systems Engineering for industry	(Laporte 1980) (Laporte 1986a) (Laporte 1995b) (April et al. 2005) (April et al. 2009)	(Laporte et al. 1986a) (Laporte et al. 1988) (Laporte et al. 1989) (Laporte et al. 1999d) Laporte et al. 1996b) Laporte et al. 1997b) (April et al. 2005) (April et al. 2006a) (April et al. 2006b) (Berrhouma et al. 2009) (Côté et al. 2004a) (Côté et al. 2005) (Corrales et al. 2007) (Laporte et al. 2003) (Laporte et al. 2005d) (Laporte et al. 2008f)	(Laporte 1985) (Laporte et al. 1998d) (Laporte et al. 1999a) (Laporte 2004a)	(Laporte et al. 1985) (Laporte et al. 1986b) (Laporte et al. 1986c) (Laporte 1994) (Laporte 1996) (Laporte et al. 1996) (Laporte et al. 1997a) (Laporte 1998) (Laporte et al. 1998a) (Laporte et al. 1998b) (Laporte et al. 1998c) (Laporte et al. 1999b) (Laporte et al. 1999c) (Laporte 2004b) (Laporte 2007e) (Laporte et al. 2007b) (Laporte et al. 2007c) (Bourque et al. 2004) (Corrales et al. 2007) (Côté et al. 2004b) (Suryn et al. 2002) (Suryn et al. 2004) (Laporte et al. 2008f)
Software Engineering for Education	(April and Laporte 2010[12])	(Laporte et al. 2007e)		(Laporte et al. 2007d)
Software Engineering for Very Small Entities (VSEs)	(Laporte et al. 2008c)	(Laporte 1995) (Laporte et al. 2005a) (Laporte et al. 2006a) (Laporte 2007a) (Laporte et al. 2007a) (Laporte et al. 2008a) (Laporte et al. 2008b) (Deniger et al. 2006) (Desfossés et al. 2008) (Habra et al. 2008) (Rheault et al. 2006)	(Laporte et al. 2006d) (Laporte et al. 2006e) (Laporte et al. 2007a)	(Laporte 2005a) (Laporte et al. 2005b) (Laporte et al. 2006b) (Laporte et al. 2006c) (Laporte et al. 2008d) (Laporte et al. 2008e) (Laporte 2007d) (Desharnais et al. 2007) (Stambollian et al. 2006)

Table 1-4 Refereed or Edited Published Works

[12] In preparation

Topic	Contribution to ISO Standards	Contribution to ISO Technical Reports	Technical Reports[13] and Contribution to Technical Reports	Keynote Speaker and other Communications
Software and Systems Engineering for industry			(SECOR 1990a) (SECOR 1990b) (SECOR 1992) (Laporte et al. 1993)	(Laporte 1981a) (Laporte 1981b) (Laporte 1981c) (Laporte 1981d) (Laporte 1982) (Laporte 1986a) (Laporte 1986b) (Laporte 1991) (Laporte 1992) (Laporte 1997) (Laporte 1999a) (Laporte 1999b) (Laporte 1995a) (Laporte 2001)
Software Engineering for Education	(ISO/IEC 2008a)	(ISO/IEC 2005a)		
Software Engineering for Very Small Entities (VSEs)	(ISO/IEC 2009b) (ISO/IEC 2009d)	(ISO/IEC 2009a) (ISO/IEC 2009c) (ISO/IEC 2009e)	(Laporte et al 2005c) (Laporte et al. 2009g) (Laporte 2009b) (Laporte 2009c) (Laporte 2009d) (BAR-026-2005a) (BAR-026-2005b) (BAR-026-2005c) (BAR-026-2005d) (BAR-026-2005e) (BAR-026-2005f) (BAR-026-2005g) (BAR-026-2005h) (BAR-026-2005i) (Vasquez 2009) (Arenas 2009) (Alexandre et al. 2009) (Palza 2009) (Guillemot 2009) (Sanyakorn 2009) (O'Connor 2009) (Varkoi 2009)	(Laporte 2006a) (Laporte 2006b) (Laporte 2007b) (Laporte 2007c) (Laporte 2008a) (Laporte 2008b) (Laporte 2009a)

Table 1-5 Other Contributions

1.4 Overview of the author's contributions to the future ISO standard for VSEs

The purpose of the thesis is to present the author's contributions to SE and to the development, deployment, and utilization of international SE standards, specifically those for VSEs, and both in Canada and abroad. To achieve this, I use the 6-phase model of the innovation process developed by Rogers (Rogers 2003), as illustrated in Figure 1-2, which

[13] Deployment Packages and survey questionnaires are included in this category.

he defines as follows: all the decisions, activities, and their impacts that occur from recognition of a need or problem, through research, development, and commercialization of an innovation through diffusion and adoption of the innovation by users, to its consequences. Although the figure shows a sequence of activities, in reality some activities are performed in parallel.

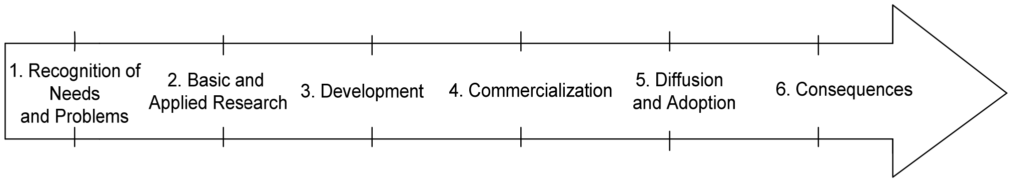

1. Recognition of Needs and Problems
2. Basic and Applied Research
3. Development
4. Commercialization
5. Diffusion and Adoption
6. Consequences

Figure 1-2 Phases of the Innovation Process (adapted from *Rogers 2003*)

Phase 1 involves the recognition of needs and problems. It was begun at a 2004 meeting in Australia of an ISO sub-committee (SC7) mandated to develop international SE standards. During the meeting, the members of the Canadian delegation were told that SC7 standards were "too bulky" and "too complicated" to use. A meeting of interested parties was held to discuss the issues surrounding needs and problems. The participants decided to create an informal one-year special interest group (SIG) to explore them, to better articulate the target, to devise a plan to get there, and to prepare and table a proposal at the next SC7 Plenary meeting in Finland in 2005 (ISO 2004c). The proposal had been sent to SC7 participating countries for comment and approval. The proposal was approved in 2005 and a new working group, Working Group 24 (WG24) was established by the ISO.

In phase 2 of the 6-phase model, involving basic and applied research, the Thai Industrial Standards Institute (TASI) invited a group of experts to Bangkok in 2005 to explore the research issues. At that meeting, I proposed to the working group that a survey be conducted of VSEs worldwide. The international survey was developed and translated into 9 languages. It was conducted among VSEs and concerned their utilization of standards. The data collected were used to validate the list of needs and problems that had been documented by the working group during the previous phase. Over 435 responses were received from 32 countries. The working group used the results to define the requirements that would form the basis for the standards to be developed for VSEs. It was then, in 2005, that the ISO formally established WG24 to address the issues, and I was appointed editor of this new group. Since WG24's objective was to prepare an initial set of standards as quickly as possible, the group analyzed international reference standards and models that could help develop a subset of ISO standards for VSEs. WG24 began a search for existing standards and models that could be tailored or adapted to the needs of VSEs. Moprosoft, a

Mexican National Standard (NMX 2005), developed to assist small and medium-sized Mexican enterprises, was selected to achieve this objective.

Phase 3, the development phase, is defined as 'a process of putting a new idea in a form that is expected to meet the needs of an audience of potential adopters' (Rogers 2003). In phase 3, WG24 developed a set of standards and technical reports. It was begun at the October 2005 meeting of WG24 in Italy, followed by bi-annual meetings held in Thailand, Luxembourg, Canada, Russia, Germany, Mexico, and India. The approach consisted of two steps: in step 1, the processes and work products applicable to VSEs were selected from the Mexican National Standard; and in step 2, this subset was tailored to fit the needs of VSEs. The group had to use the concept of the ISO profile (ISP – International Standardized Profile) to develop the new standards. A profile is a kind of matrix which precisely differentiates all the elements that are taken from existing standards from those that are not. The decision was made to develop a 4-phase roadmap:

- Phase 1 targets VSEs typically developing 6 person-month projects or start-up VSEs;
- Phase 2 targets VSEs developing only one project at a time;
- Phase 3 targets VSEs developing more than one project at a time;
- Phase 4 targets VSEs wishing to put in place business management practices and portfolio management practices.

The following set of documents, targeted by audience, has been developed by the group. These are (ISO2009a): Part 1: Overview; Part 2: Framework and Taxonomy; Part 3: Assessment Guide; Part 4: Profile Specifications; and Part 5: Management and Engineering Guides.

During this phase, some skunk work was performed. Skunk work is defined as work carried out in 'an environment that is intended to help a small group of individuals design a new idea by escaping routine organizational procedure' (Rogers 2003). This is the period when I proposed to the members of WG24 the development, in parallel with the development of standards and technical reports, a means to facilitate and accelerate the adoption of those standards and technical reports by VSEs worldwide. The set of documents should, in principle, be approved by national standards bodies and published by the ISO in 2010. Its publication by the ISO will mark the end of the development phase.

Phase 4 of the 6-phase model, involving commercialization, will start with the above-mentioned ISO publication, according to the innovation model. However, some commercialization activities began with approval by the ISO and by the national standards bodies in 2005 for a new working group for VSEs, WG24. Other commercialization activities were also performed when the ISO coordinated a series of review cycles and collected hundreds of comments from those national standards bodies. WG24 analyzed all the comments and produced a revised version of the set of documents to be sent to the national standards bodies for the next round of review. Commercialization activities have also been performed in the years since 2005, when the first papers on the work of WG24 were published in French (Laporte et al. 2005a) and in English (Laporte 2005b).

To address phase 5 of the 6-phase model, on diffusion and adoption, I proposed, at the Moscow meeting of the WG24 in 2007, the development of a set of documents, called Deployment Packages, to facilitate the adoption and implementation of a set of ISO standard practices for VSEs. A deployment package is a set of artifacts developed to facilitate the implementation of a set of practices, of the selected framework, in a VSE. The elements of a typical deployment package are: a detailed description of processes, activities, tasks, roles, and products; a template; a checklist; an example; a reference and a mapping to standards and models; and a list of tools. The following deployment packages have been developed: Requirements Analysis, Architecture and Detailed Design, Construction and Unit Testing, Integration and Test, Verification and Validation, Version Control, Project Management, Product Delivery, and Self-Assessment. These documents will be validated through a series of pilot projects in VSEs around the world.

At the Mexico meeting of WG24 in 2008, I proposed the establishment of an international network of collaborators whose aim is to promote, facilitate, and develop collaborative activities between institutions to improve VSE capabilities. The main activities of the network are to accelerate the deployment of the Standards and Guides for VSEs and to accelerate the development and application of the Guides and Deployment Packages. The following organizations have signed a collaboration agreement with the ÉTS: the Centre d'Excellence en Technologies de l'Information et de la Communication (Belgium), the Parquesoft Foundation (Columbia), the Tampere University of Technology (Finland), the Université de Bretagne Occidentale (France), the Irish Software Engineering Research Centre (Ireland), the Public Research Centre Henri Tudor (Luxembourg), and the Institute of Software Promotion for Industries (Thailand).

At the 2009 meeting of WG24 in India, I proposed the establishment of an education interest group, the main objective of which is to develop a set of courses for undergraduate and graduate SE students to enable them to learn about the ISO standards for VSEs before they graduated. The courses developed for academia would consist of education deployment packages similar to the development packages for VSEs. The objective of the educational deployment packages is to facilitate and accelerate the teaching of the new ISO standards in educational institutions by providing them with readily usable teaching material, such as course plans, presentation material, exercises, case studies, and reading lists.

In the final phase of the 6-phase model, on the consequences of an innovation, I present the potential positive and negative consequences of the publication of standards for VSEs by the ISO.

1.5 Conclusion

In this chapter, the author has presented an overview of its contributions to SE and to the development of software standards for VSEs. Appendix A describes my accomplishments in more detail to enable the reader to better understand my involvement and motivation in the development of international standards for VSEs.

In chapter 2, the author relates industry trends and challenges. The characteristics of VSEs are presented to better understand why VSEs need standards adapted to their needs. This chapter covers phase 1, on the recognition of needs and problems.

Phase 2, on basic and applied research, is covered in chapter 3, titled Overview of Process Improvement Initiatives for VSEs, and in chapter 5, which describes the survey conducted to gain a better understanding of the problems and needs of VSEs regarding their use of standards.

Phase 3, on development, is covered in chapter 4, titled The Development Process of International Standards, and in chapter 6, titled The Development of International Standards for VSEs.

Phase 4, on commercialization, and phase 5, on diffusion and adoption, are covered in chapter 7, titled Development of the Means to Accelerate the Adoption and Utilization of International Standards by VSEs.

The final phase, phase 6, on consequences, is covered in chapter 8, titled Future Work and Conclusion.

This page is intentionally left blank

Chapter 2

Trends in Software Engineering and the Need for standards adapted for VSEs

2 Chapter 2 – Trends in Software Engineering and the Need for standards adapted for VSEs

2.1 Introduction

Phase 1 involves the recognition of the needs and problems of Very Small Entities (VSEs) with regard to software standards, and the task of researching and documenting them, as illustrated in Figure 2-1. This phase was begun at a 2004 meeting in Australia of an ISO subcommittee (SC7) mandated to develop international software engineering (SE) standards. During the meeting, the members of the Canadian delegation were told that SC7 standards were "too bulky" and "too complicated" for VSEs to use. A meeting of interested parties was held to discuss the issues surrounding the needs and problems of VSEs, at which the participants decided to create an informal one-year special interest group (SIG) to explore them, better articulate the target, devise the plan to get there, and prepare and table a proposal at the next SC7 Plenary meeting in Finland in 2005 (ISO 2004c).

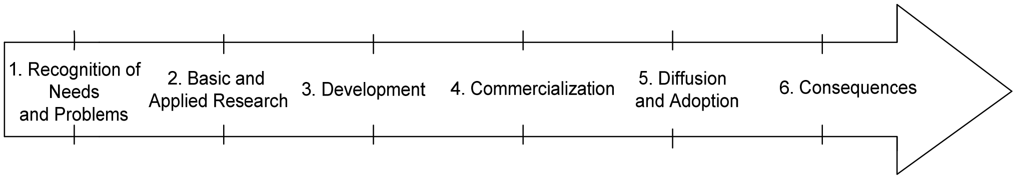

1. Recognition of Needs and Problems
2. Basic and Applied Research
3. Development
4. Commercialization
5. Diffusion and Adoption
6. Consequences

Figure 2-1 Phases of the Innovation Process (adapted from Rogers 2003)

VSEs have been defined as enterprises, organizations, departments, or projects with fewer than 25 employees. Information Technology VSEs are very important to the world economy, as their software components are often integrated into the products of larger entities and Failure to deliver a quality product on time and within budget is a threat to the competitiveness of both VSEs and their customers. One way to mitigate this threat is by having the suppliers of a product chain put in place proven SE practices, such as those documented in standards.

In the next section, the author presents the current trends in the software industry and the challenges facing it[14], followed by the concept of 'best practices' and an account of the evolution of SE standards. The Capability Maturity Model (CMM), developed by the Software Engineering Institute (SEI), is discussed, since CMMs are considered to be de facto standards. The deficiencies of the standards are then listed, to demonstrate that the

[14] The term *software industry* includes both the industries that produce software only and industries that produce systems with software components.

current standards do not meet the needs of VSEs. The characteristics and needs of VSEs are presented in the final sections.

2.2 The trends facing industry

According to the National Academy of Science, "software is not merely an essential market commodity but, in fact, embodies the economy's production function itself" *(Jorgenson et al. 2006)*. In this section, we present the challenges, trends, and consequences with regard to the development of software in the near future (adapted from *(Boehm 2006)* and *(Humphrey*[15]):

- **Increasingly large software systems –** Figure 2-2 shows the trends, in lines of code (LOC), of the software owned by the US Department of Defense (DoD) between 1950 and 2000. The figure for the year 2000 is about 200 trillion LOCs. *(Boehm 2006)*.

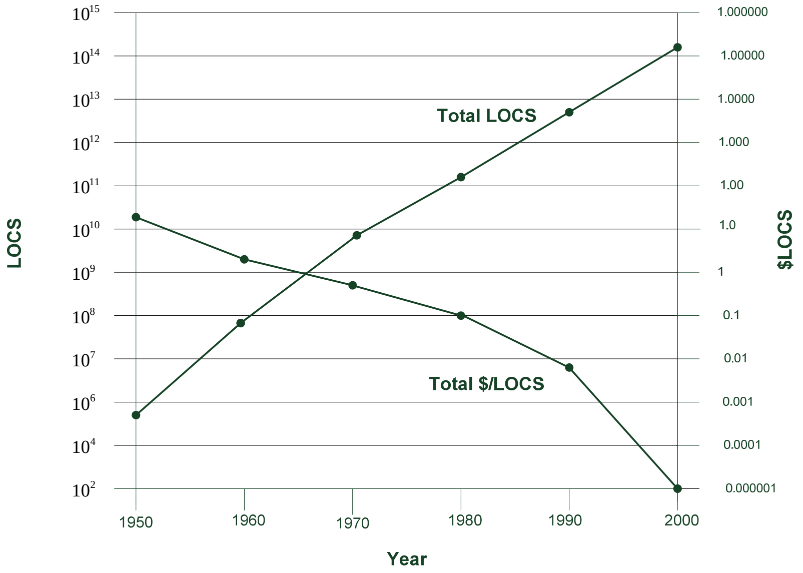


Figure 2-2 US DoD Lines of Code in Service and Cost/LOC (Boehm 2006)

Figure 2-3 shows that the size of the software required for various kinds of systems and applications has been growing exponentially for the past 40 years. The trend line in the center of the chart shows a compound growth rate of 10 times every 5 years. The size of the software used for any given function has been growing at a similar rate, of roughly 10 times every 5 years. If software growth continues at this historic rate, it will will mean an increase of 10,000 times in the next 20 years.

[15] www.sei.cmu.edu/publications/news-at-sei/columns/watts_new/watts-new-compiled.pdf

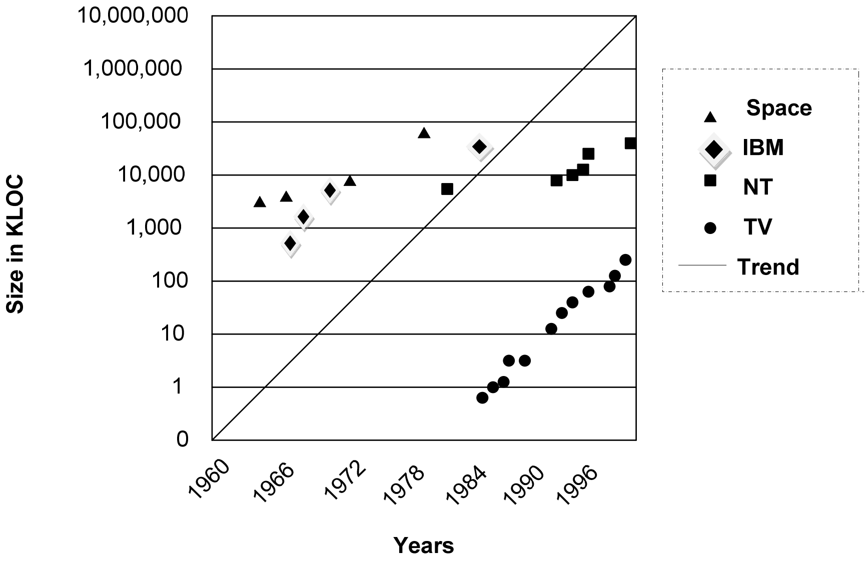


Figure 2-3 Exponential software growth (Humphrey[16])

- Increasingly complex software systems – Figure 2-4 illustrates a transportation system as an example of a complex system,. Every major system is composed of many subsystems, and each subsystem is made up of computer-based components with thousands, if not millions, of lines of code. A main concern with growing complexity is the scalability of the development process, which means that organizations will have to develop processes that scale up, in order to handle the massive increases in system scale expected in the future.

[16] www.sei.cmu.edu/publications/news-at-sei/columns/watts_new/watts-new-compiled.pdf

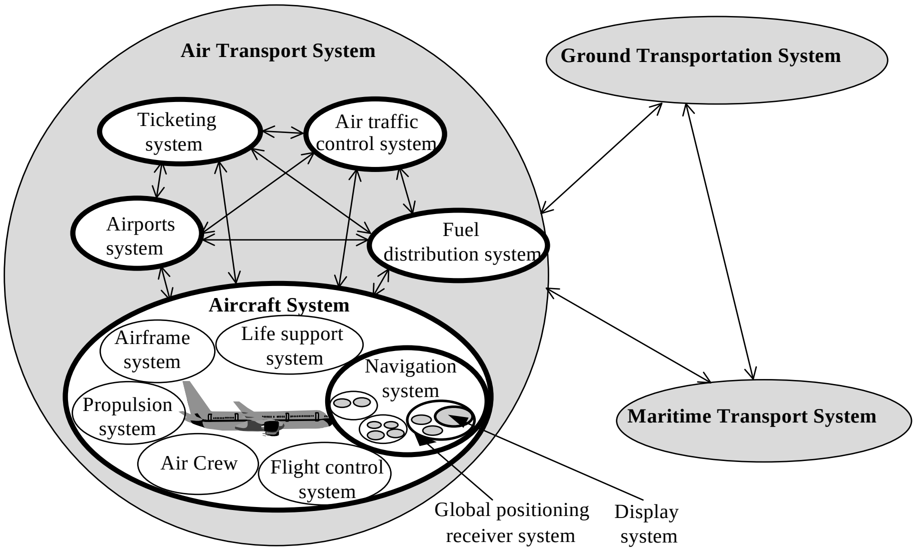


Figure 2-4 Typical system view of the transportation sector translated from AFNOR 2003)

- **The emergence of Software-Intensive Systems of Systems (SISOS)** – Very large, dynamically evolving, and previously unknown software-intensive systems of systems (SISOS) are emerging, with their requirements, behaviors, and complex socio-technical issues to address, for example, the International Space Station. Table 2-1 provides some additional characteristics of SISOS.

Characteristics	**Range of Values**
Size	10-100 million lines of code
Number of external interfaces	30-300
Number of cooperative suppliers	20-200
Depth of supplier hierarchy	6-12 levels
Number of coordination groups	20-200

Table 2-1 SISOS complexity (adapted from Boehm 2006)

- **The increasing integration of software engineering and systems engineering** – Traditionally, and even recently for some forms of agile methods, systems and software development processes have been recipes for standalone *stovepipe* systems. Modern systems, in contrast, are intricate combinations of hardware and software, and the industry of the future needs employees who can understand both sides of the system and appreciate how they must seamlessly communicate and interface *(Korneck 2008)*.

- **Increasing demand for secure software –** We will probably always have criminals and terrorists, so we must write our programs to operate in a threatening and unfriendly world. Software acquirers are becoming more and more aware of the security, integrity, privacy, and safety issues affecting software, and are demanding assurance that software engineers have defined and adopted mature practices in these areas *(Magee et al. 2004).*

- **Increasingly rapid change –** The pace of unforeseeable change from competitive threats, technology, stakeholder preferences, organizational realignments, and the environment continues to accelerate. Traditional requirements management and change control processes are no match for large volumes of change traffic across the multitudes of suppliers and operational stakeholders involved in a SISOS. Global connectivity also accelerates the ripple effects of technology, marketplace, and technology changes; for example, Hewlett Packard's initiative to reduce product line software development times from 48 to 12 months *(Grady 1997).* The pace of change places a high priority on systems, and SE process agility techniques that enable agile methods to work well on small to medium-sized projects can become sources of difficulty when applied to larger, more mission-critical projects.

- **Increasing demand for globalization** – Market forces, as well as politicians, are asking for some degree of 'local' development, which means that multinational corporations will have to develop software systems all around the world to keep or capture new markets. They also want to benefit from globalization in two ways:
    - Differential salaries provide opportunities for cost savings through global outsourcing.
    - The ability to develop across multiple time zones creates the prospect of very rapid development via three-shift operations.

- **High failure rates of IT projects** – The Standish Group's 2003 report stated that only 34% of all projects were classified as successful *(Standish 2003).* Figure 2-5 illustrates the project resolution history.

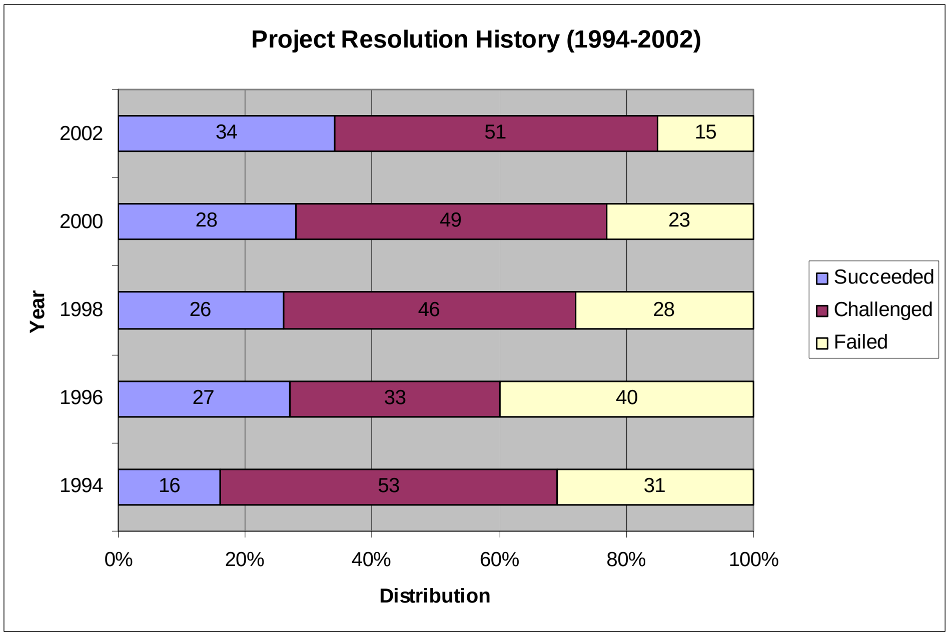


Figure 2-5 Project resolution history (Standish 2003)

- **Increasing demand for defect-free software** – Since safety, security, and privacy are becoming largely software issues, software development processes must provide defect-free software systems. But organizations at CMM level 1 deliver systems with an average of 7,500 defects per MLOC (million lines of code) and those at level 5 average 1,050 defects per MLOC *(Davies et al. 2003)*. A 200 KLOC (thousand lines of code) program with 5 undetected defects per KLOC would have 1,000 defects. When software systems are defective, they cannot be safe or secure.

- **High cost of rework –** In a project, the total cost is composed of four components: the cost of performance, the cost of appraisal, the cost of prevention, and the cost of anomalies, often called the cost of rework. A study performed at Raytheon showed (Figure 2-6) that the cost of rework in 1987 was about 41% of total project cost at CMM level 1. Rework was 18% at level 2, 11% at level 3, and 6% at level 4 *(Dion 1993, Haley 1996)*.

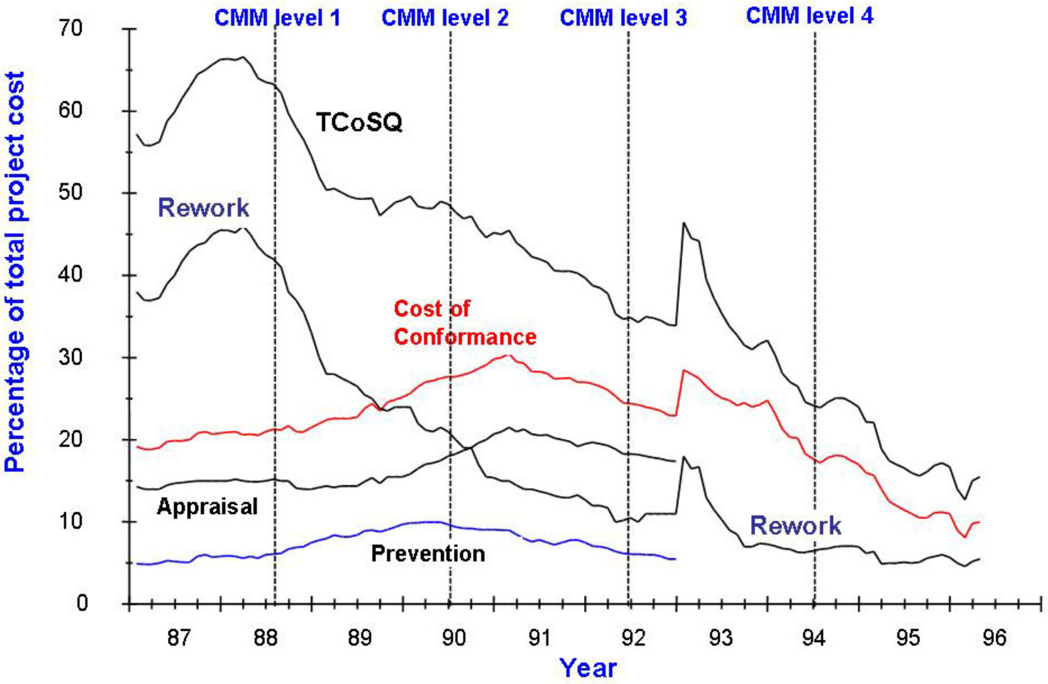


Figure 2-6 Improvement data (Dion, 1993, Haley, 1996)

A study by Krasner *(Krasner 1998)* showed that rework varies between 15% and 25% of the cost of developing a maturity level of 3 (see Table 2-2).

CMM Maturity Level	Percentage of rework
1	≥ 50%
2	25 % à 50%
3	15 % à 25%
4	5 % à 15%
5	≤ 5%

Table 2-2 Relationship between CMM maturity levels and the cost of rework (Krasner 1998)

According to Charette, 'Studies have shown that software specialists spend about 40 to 50 percent of their time on avoidable rework rather than on what they call value-added work, which is basically work that's done right the first time. Once a piece of software makes it into the field, the cost of fixing an error can be 100 times as high as it would have been during the development stage' *(Charette 2005)*.

I have collected, from professional engineers and graduate students in the professional software engineering program at the ETS, data about the cost of quality in their environment. As illustrated in Table 2-3, the cost of rework is about 30%. Most of these data were collected in two large multinational companies: one involved in the transport sector and the other in the aerospace sector.

	Site A American Engineers (19)	**Site A American Managers (5)**	**Site B European Engineers (13)**	**Site C European Engineers (14)**	**Site D European Engineers (9)**	**Course A Master's Level 2008 (8)**	**Course B Master's Level 2008 (14)**
Cost of Performance	41%	44%	34%	31%	34%	29%	43%
Cost of Rework	30%	26%	23%	41%	34%	28%	29%
Cost of Appraisal	18%	14%	32%	21%	26%	24%	18%
Cost of Prevention	11%	16%	11%	8%	7%	14%	10%
Quality (Defects/ KLOC)	71	8	23	35	17	403	19

Table 2-3 Cost of quality

- **Complex life cycle development processes** – The development of modern systems, composed of many hardware and software subsystems, involves a complex life cycle, as illustrated in Figure 2-7. When only one hardware subsystem and the associated software are integrated, the tasks are relatively easier than the integration of dozens of hardware and software subsystems.

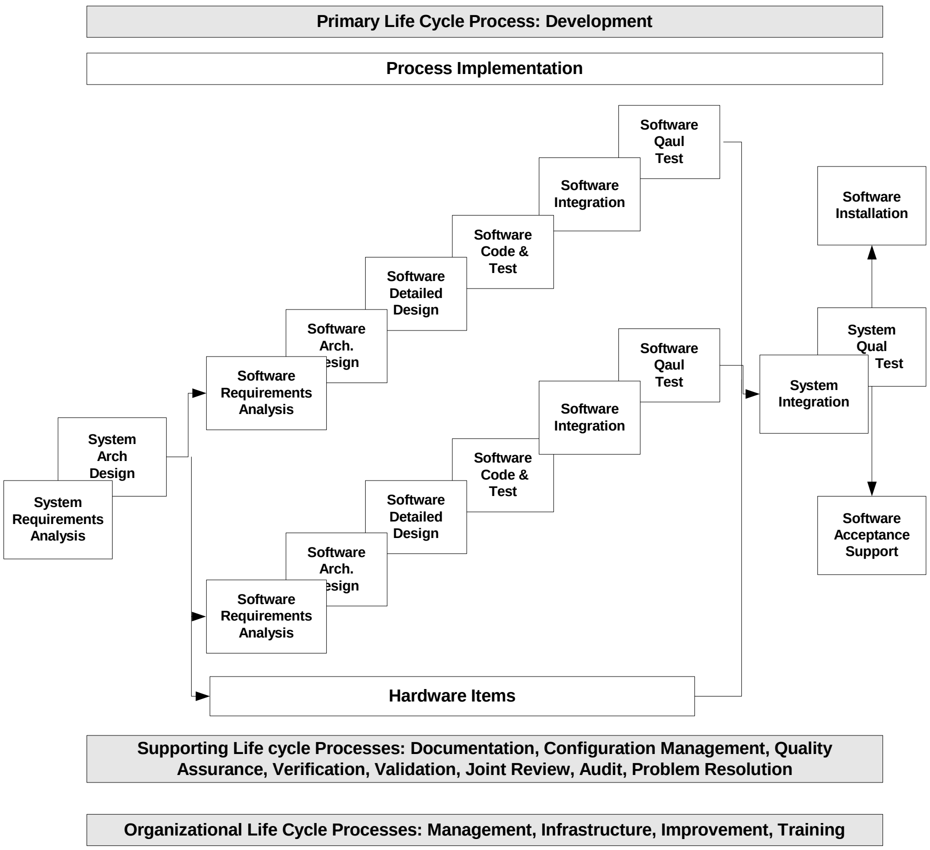

Primary Life Cycle Process: Development
Process Implementation
Software Qaul Test
Software Integration
Software Code & Test
Software Detailed Design
Software Arch. esign
Software Requirements Analysis
System Arch Design
System Requirements Analysis
Software Qaul Test
Software Integration
Software Code & Test
Software Detailed Design
Software Arch. esign
Software Requirements Analysis
Hardware Items
Software Installation
System Qual Test
System Integration
Software Acceptance Support
Supporting Life cycle Processes: Documentation, Configuration Management, Quality Assurance, Verification, Validation, Joint Review, Audit, Problem Resolution
Organizational Life Cycle Processes: Management, Infrastructure, Improvement, Training

Figure 2-7 Life cycle of product development (DoD 1994)

- **Growing interconnectedness of systems** – More and more systems are interconnected, and they are now exposed to hackers, criminals, and terrorists. These people want to break into our systems. If they succeed, they can do us great harm.

- **A decline in interest in the computer profession in developed countries** – Secondary school students perceive opportunities in computing as rapidly vanishing or being offshored to Asia. For example, there has been a decrease of 39% in the number of American students choosing computer science as a major since 2000, while the predicted proportion of the world's scientists and engineers living in Asia by 2010 is 90% (Microsoft 2006).

- **A rapid retirement of expertise in developed countries** – In the US in 2005, the average age of an aerospace engineer was 54. In addition, more than 26 percent of US aerospace workers were eligible for retirement in 2008 *(Long 2008)*.

Consequences

In the previous paragraphs, a few of the trends facing industry have been described. Table 2-4 lists some of the consequences resulting from those trends.

• As with some domains like railroads and telecommunications, a standards-based infrastructure will be essential for effective global collaboration.
• Traditional requirements management and change control processes are no match for large volumes of change traffic across the multitudes of suppliers and operational stakeholders involved in a SISOS.
• Global connectivity accelerates the ripple effects of technology, marketplace, and technology changes.
• The pace of change places a high priority on systems and SE process agility.
• Techniques that enable agile methods to work well on small to medium-sized projects can become sources of difficulty when applied to larger, more mission-critical projects.
• As we move to an increasingly global economy, vendors are putting more faith in international standards as a means of selling their products.
• At current quality levels, the impact of software defects will grow.
• As programs become larger, the number of possible program conditions increases exponentially. This has two related consequences. The number of tests required to achieve any given level of test coverage also increases exponentially, with both program size and the time it takes to find defects and fix a program.
• The demand for software engineers is at an all-time high, and it continues to increase. As application programs become larger and more sophisticated, the knowledge and experience required for those programs will increase as well.
• To handle the massive increases in system scale, organizations must employ processes that scale up.
• Safety, security, and privacy are becoming largely software issues.
• With the poor state of software design and the growing likelihood of serious incidents caused by poor quality and non secure or unsafe software, we can expect increased numbers of life-critical or business-critical catastrophes to occur.
• The current cottage-industry approach to developing application programs must give way to a more professional and well-managed discipline.

Table 2-4 Consequences of the industry trends
(adapted from Humphrey[17] and Boehm 2006)

Since one of the challenges facing the international community will be to develop increasingly high-quality code, and a great deal of it, and highly complex systems, VSEs will play a critical role in the development and maintenance of these systems. As explained in more detail below, VSEs will need help in identifying recognized SE practices and guidance in applying them.

2.3 The search for best practices

Why address this topic when most engineers and scientists know what a best practice is? Labeling a practice 'best' raises controversy in organizations and among software practitioners. In a young discipline such as this, a best practice is a not only situation-specific, but a moving target. So, most people who use this term have in mind their own

[17] www.sei.cmu.edu/publications/news-at-sei/columns/watts_new/watts-new-compiled.pdf

definition when discussing SE practices. I realized when I first became involved in ISO standard development, that there was no well-known and accepted definition of the term. Below are a few definitions from papers or from the Internet.

A best practice can be defined as a technique, method, process, activity, incentive, or reward that is more effective at delivering a particular outcome than any other technique, method, process, etc[18]. The idea is that, with proper processes, checks, and testing, a desired outcome can be delivered with fewer problems and unforeseen complications. It can also be defined as the most efficient (least amount of effort) and effective (best results) way of accomplishing a task, based on repeatable procedures that have proven themselves over time for large numbers of people.

Despite the need to improve on processes as times and conditions change, some consider the term to be business jargon for the process of developing and following a standard way of doing things that multiple organizations can use for management, policy, and especially software systems.

The term is often misused, however. It is frequently used to support politically correct ideals which, in reality take no account of individual needs or circumstances. If this is the case, that practice is far from 'best' when the resulting effects do not relate to real ideals. It is also used to prevent challenges to rules and systems which are not, in fact, best practices.

As the term has become more popular, some organizations have begun using it to refer to what are actually rules, causing a linguistic drift in which a new term, such as 'good idea', is needed to refer to what would previously have been called a 'best practice'.

Chevron Corporation has a different approach, defining a best practice as any practice, knowledge, know-how, or experience that has proven to be valuable or effective within one organization that may have applicability to other organizations. They recognize the following four levels of best practice (adapted from APCQ 1997):

- **Good Idea** — This is an unproven practice, i.e. not yet substantiated by data, but makes sense intuitively, and could have a positive impact on business performance. It requires further review/analysis. If substantiated by data, it could be a candidate for implementation at one or more Chevron locations.

- **Good Practice** — This is a technique, methodology, procedure, or process that has been implemented and has improved business results for an organization (satisfying some element of customers' and key stakeholders' needs). It is substantiated by data

[18] http://en.wikipedia.org/wiki/Best_practice

collected at the location; however a limited amount of comparative data exists from other organizations. It is a candidate for application at one or more Chevron locations or departments, and possibly at other Chevron locations.

- **Local Best Practice** — This is a good practice that has been determined to be the best approach for an entire organization or large parts of it, based on an analysis of process performance data which includes some review of similar practices outside Chevron (competitive intelligence data). It is applicable at most or all Chevron locations or departments, and may be applicable at other Chevron locations

- **Industry Best Practice** — This is a practice that has been determined to be the best approach for an entire organization, or large parts of it, based on internal and external benchmarking, including the analysis of performance data. External benchmarking is not confined to the organization's industry. It may be applicable at other Chevron locations.

The American Society for Quality (ASQ) defines a best practice as 'a superior method or innovative practice that contributes to the improved performance of an organization, usually recognized as "best" by other peer organizations'. Harrison, then editor of the IEEE Software Magazine, challenged this definition with the following questions (Harrison 2004):

- Who are these "peer organizations"?
- How many peer organizations have to recognize the practice as best?
- Who's responsible for collecting data about which peer organizations recognize the practice and which don't?
- How does a peer organization recognize a best practice?
- What about organizations in which some units use a particular practice and others don't?
- What about conflicting best practices?

The US DoD's definition is the following: Methodologies and tools that consistently yield productivity and quality results when implemented in a minimum of 10 organizations and 50 software projects and are asserted by those who use it to have been benefic in all or most of the projects (DACS 2003).

Fogle provided the following criteria for the identification of best practices (Fogle 2001):

- **Existence** — The practice must have been observed in at least one organization.

- **Importance** — In the benchmarking team's opinion, the practice is important to an effective process.
- **Effectiveness** — In the benchmarking team's opinion, the practice appears to work well where it is used.
    - **Tangible benefit.** In the benchmarking team's opinion, there is a tangible benefit to the organization that performs this practice.
    - **Innovation.** Where appropriate, the practice makes use of innovation, such as automation, in place of manual methods for accomplishing a task.
    - **High value perceived by users of the practice**. The practice must be rated highly on a validation survey of users representing benchmarked organizations.

Turner warns readers about the search for best practices: 'While there might be a few universally good practices (such as peer reviews), and there are certainly required practices (such as configuration and requirements management), most practices have hidden assumptions and conditions for use, and there is little empirical data to support evaluation and selection.' (Turner, 2003). Industry has been working very hard at selecting practices since the early days of SE in the 60s. Many of these practices have been codified in standards. The issues surrounding standards are presented and discussed below.

2.4 What is software engineering?

At a NATO conference in 1968, the term 'software engineering' was first registered (Naur et al. 1968). ISO (ISO/IEC 2009) defines software engineering as follows:

1. The systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software. ISO/IEC 2382-1:1993 Information technology--Vocabulary--Part 1: Fundamental terms.

2. The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. ISO/IEC 24765, Systems and Software Engineering Vocabulary.

2.5 What is a standard?

The British Standards Institute (BSI) defines a standard as follows:

> *A technical specifications or other document available to the public, drawn up with the cooperation and consensus or general approval of all interests affected by it, based on consolidated results of science, technology and experience aimed at the promotion of optimum community benefits (BSI 1981).*

The ISO definition is:

> *Mandatory requirements employed and enforced to prescribe a disciplined uniform approach to software development, that is, mandatory conventions and practices are in fact standards (ISO 2009f).*

Some attributes need to be added to these definitions, in the interests of the development of good standards (adapted from Pfleeger, et al. 1994):

- Be able to determine whether or not the standard is being complied with by an organization/project
- Be able to determine how the standard is being complied with
- Be able to determine what is affected when using the standard
- What attributes of the product/process are improved using the standard
- Be able to determine the cost of using the standard

2.5.1 What is a software engineering standard?

Before presenting the definition of an SE standard, it is interesting to note that, unlike other engineering disciplines such as mechanical or chemical engineering, SE is not based on the physical sciences. Our discipline has evolved mainly based on experience gathered when developing more and more complex software systems. Practitioners have gradually identified and documented 'what to do', 'when to do it', and 'how to do it' as practices. ISO (ISO 2009f) defines practices as follows:

3. An activity that contributes to the purpose or outcomes of a process, or enhances the capability of a process. ISO/IEC 15504-1:2004 Information technology -- Process assessment -- Part 1: Concepts and vocabulary. 3.27.

4. Requirements employed to prescribe a disciplined uniform approach to the software development process. ISO/IEC 24765, Systems and Software Engineering Vocabulary.

5. A specific type of professional or management activity that contributes to the execution of a process and that may employ one or more techniques and tools. A Guide to the Project Management Body of Knowledge (PMBOK® Guide) -- Third Edition. See also: conventions, standards.

As those practices are published and applied by a community of users, consensus is obtained, and some regulatory organizations publish them as standards or mandate them.

2.5.2 The Evolution of Software Engineering Standards

Below, the evolution of SE standards is described. This will help the reader better understand the need to develop standards adapted to the context of VSEs.

The first software quality-related standard, MIL-S 52779, was published in 1974 by the US Army (Schulmeyer et al. 1999). Between 1974 and 1989, of the 32 standards and plan guidelines published in the US, 22 were developed by the US military and only 5 by a professional organization: the IEEE Computer Society.

It is very interesting to look at the evolution of standards, since it allows us to better understand the evolution of SE since the NATO conference. Figure 2-8 illustrates the evolution of some SE standards since the publication of the US military standards in 1988 (Sheard 2001). On the right-hand side, the emphasis is on the US military standards, while on the left-hand side, the emphasis is on the Capability Maturity Model (CMM) (Paulk et al. 1993) developed by the SEI.

I participated in the development of software using the first generation of military standards, i.e. DoD-Std 2167A titled Defense Systems Software Development, in the late 1980s. The standard was perceived to be biased toward the use of the waterfall model. It was criticized for requiring the development of hundreds of pages of support documentation and requiring expensive formal reviews and audits to be carried out. This standard was ‘improved’ and replaced by MIL-STD-498 (DoD 1994). One complaint of the industry was that they had to comply with the standards required by each customer. For example, an aircraft manufacturer had to comply with US Department of Transport regulations for aircraft sold in certain countries, in order to comply with military standards when developing subsystems for military aircraft, and to aerospace standards when producing components for space systems. The contractor had to produce an enormous quantity of documents to satisfy the content and format requirements of each standard. A decision was then made by some government agencies to wait for the publication by a professional society of an acceptable set of SE standards and discontinue use of the military standards. This has led to the adoption of the IEEE 12207 software standard by the US military.

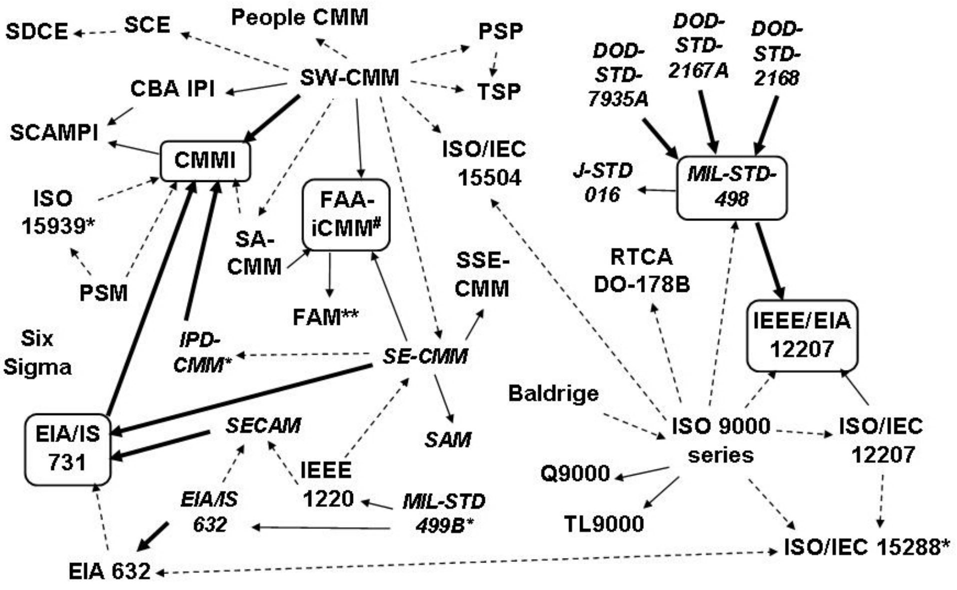


Figure 2-8 Evolution of software engineering Standards (Sheard 2001)

2.5.3 The US Department of Defence moved to commercial standards

Historically, the US DoD has developed and maintained its own set of software engineering standards. Extensive government resources were required to develop and maintain them, and developers incurred additional costs to comply with both US DoD and commercial standards. These factors combined to drive up the cost of sustaining systems across their life cycle. In the mid-1990s, the US DoD decided to shift the responsibility for selecting and maintaining standards to the developer community.

The US DoD anticipated the following benefits from this decision (DACS 2003):

- Improved affordability of systems over their life cycle by:
  - Allowing programs to leverage commercially funded or developed technology
  - Achieving economies of scale that were previously unattainable
  - Taking advantage of increased competition
  - Providing a lower cost path for insertion of new technologies into existing platforms
- Greater adaptability to evolving requirements and threats
- Mitigation of the risk associated with technology obsolescence
  - Systems supported by a wide range of readily available products

- Reduction in development cycle time
  - Faster upgrade of legacy systems with less complexity and cost
  - Development of subsystems and components that are testable
  - Application of the “Product Line Approach” to acquisition of weapons systems enabled
- Achievement of higher levels of performance
  - Contribution to ensuring interoperability among systems without major modification to existing components

2.6 The Software Capability Maturity Model

The SEI was asked by the US DoD to define a way to identify the U.S. Air Force’s most competent software vendors (Humphrey 2008). A task force studied 17 major projects, and force found that the average schedule overrun for these projects was 75% and that not one of the projects was completed on time or within the original cost target. The principal causes of project failure were not technical, but management. The task force listed the practices in the form of questions, and constructed a questionnaire composed of 85 such questions to evaluate organizations. Since many DoD acquisitions had many vendors bidding for a project, it was not easy to rank them using the results of the questionnaire. For example, if 10 companies completed the questionnaire, this resulted in the evaluation of 850 answers. The SEI task force was asked to develop an easier ranking system, a maturity model such as the one devised by Crosby (Crosby 1979). This resulted in the development of the first SEI Capability Maturity Model (CMM).

The Software Capability Maturity Model (SW-CMM) (Paulk et al. 1993) is a model of the key practices to be implemented in any organization wishing to develop or increment software aimed at high quality and significant productivity. Based on the concepts of total quality and continuous improvement, the SW-CMM was the object of a wide consensus on the part of the software community. It quickly became a de facto standard, first in the US and then worldwide, for assessing the maturity of an organization’s software. It can be used as is, as a manual of good practices, but also as a reference for audits or assessments within an organization specializing in software development and maintenance. Table 2-5 presents the main features for each of the five levels of maturity. To be eligible to attain a level of maturity, an organization must have implemented all key sectors of that level, as well as those of the lower levels.

1. Initial
The initial development process is characterized by its ad hoc, improvisational nature. Typically, there are no formalized procedures, cost estimates, or project planning. The quality of the personnel thus becomes a determining factor, and the success of this approach rests largely on the people performing it.

2. Repeatable
At this level, project management processes are introduced to control costs and schedules, and to ensure that the project runs smoothly. Minimal controls are in place to provide an overview, ensure quality control, and follow up on modifications. The company can thus react predictably when carrying out projects similar to those already successfully completed.

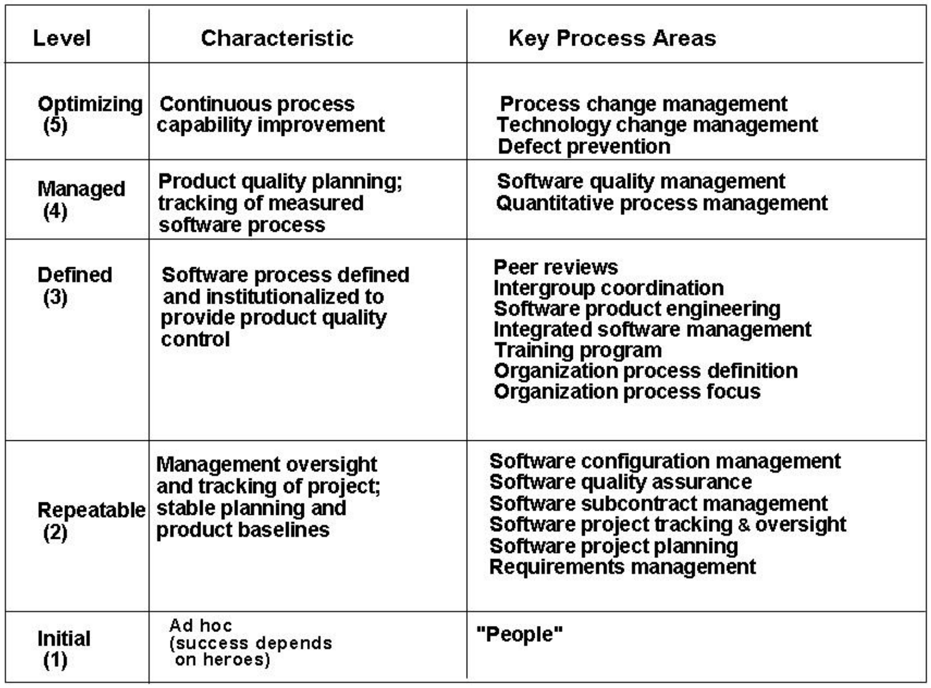
3. Defined
The company has defined the mechanisms for examining the process and the means needed to improve it. It has identified and documented a software development and maintenance process, which is applied to all projects.

4. Managed
To advance to this level, the company must introduce the process measurements needed to ensure that the process is applied systematically. This approach requires a central data bank containing all the relevant quantifiable data.

5. Optimizing
At this level, the procedure for gathering data to measure and analyze internal processes is automated. This ensures statistical control of the overall process, which prevents modification-related problems. The company uses these data to constantly refine the overall process.

Table 2-5 The SW-CMM Maturity Model

Using this model as a base, the SEI has developed a number of assessment methods for an organization that develops or maintains software. One of them, called CBA IPI (Capability Maturity Model-based appraisal for internal process improvement), is designed not to audit a software supplier, but to enable an internal examination by that organization of its practices with a view to deriving an improvement plan to be applied to its own software organization. Table 2-6 lists the main components of the SW-CMM, its maturity levels and their associated practices or process areas.



Level	Characteristic	Key Process Areas
Optimizing (5)	Continuous process capability improvement	Process change management Technology change management Defect prevention
Managed (4)	Product quality planning; tracking of measured software process	Software quality management Quantitative process management
Defined (3)	Software process defined and institutionalized to provide product quality control	Peer reviews Intergroup coordination Software product engineering Integrated software management Training program Organization process definition Organization process focus
Repeatable (2)	Management oversight and tracking of project; stable planning and product baselines	Software configuration management Software quality assurance Software subcontract management Software project tracking & oversight Software project planning Requirements management
Initial (1)	Ad hoc (success depends on heroes)	"People"

Table 2-6 Software Capability Maturity Model (Paulk et al. 1993)

2.6.1 *Difficulties in using the CMM*

Many organizations find the CMM hard to use. It has been designed with a large-scale development environment in mind, along with its associated set of roles, responsibilities, and organizational structure. As reported by an American consortium: *The key practices of the CMM were written from the point of view of large, mission-critical software development projects being performed on behalf of a government agency. Thus, the orientation of the CMM, and especially the key practices, reflects the typical large-scale software development environment of a government contractor* (SPC 1998). Figure 2-9 lists the challenges of adapting the CMM to small organizations with 2 to 10 practitioners. There are similar issues when trying to adapt the CMM for short duration projects, i.e. projects with a schedule of less than 6 months.

In a subsequent chapter, the strategy used to make it possible to provide small projects and small organizations with processes adapted to their size are presented.

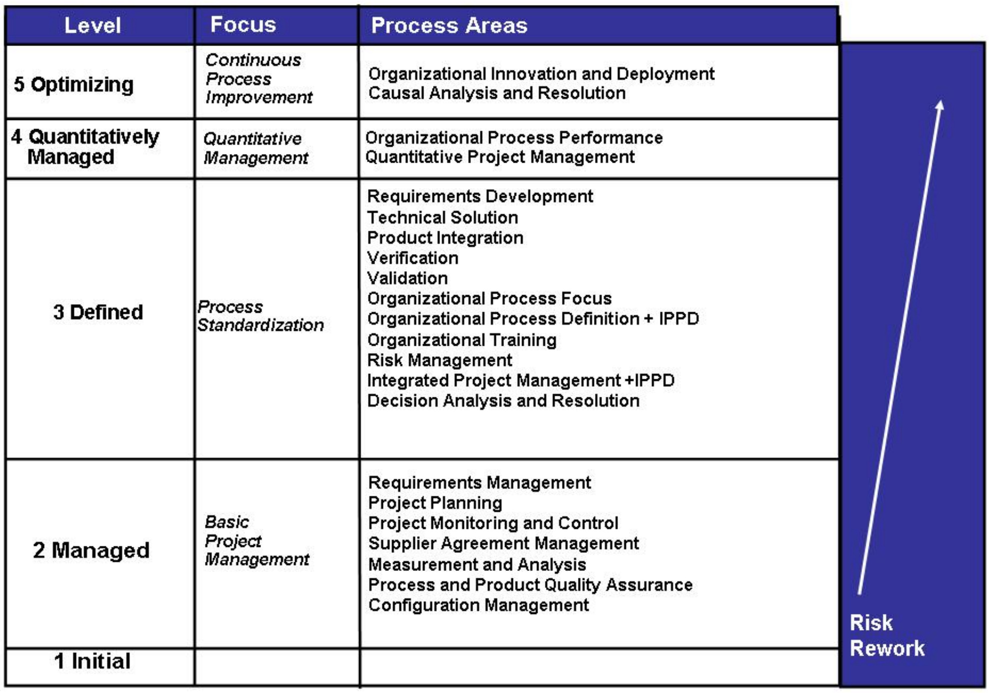
- **Defining Project.** The CMM's use of the word 'project' may or may not match the way the term is defined in a VSE.
- **CMM Roles.** Often team member roles do not seem to closely match the roles described in the CMM, which describes key practices using terminology related to project and organizational roles. An important step in understanding the relevance of the CMM is to map the roles it uses to the roles commonly used by the projects. On many small projects, roles and responsibilities are understood, but not documented.
- **Senior Management Involvement.** Senior management reviews and other responsibilities may not be performed sufficiently to meet the requirements of the CMM for small projects.

- **Configuration Control Board (CCB).** Small projects often do not have enough team members to create a traditional CCB with the authority to manage the project's software baselines.
- **Team Training.** It is difficult to provide training to small project team members at the right time.
- **Process Definition.** Defining the process can be overwhelming for a small project. Many activities must be performed 'according to a documented procedure.'
- **Peer Reviews.** The team may be too small to perform Fagan-style peer reviews with team members. Various techniques, such as a desk check, depending on the level of risk associated with the work product, can be used by teams.
- **Tailoring of a Organizational Standard Process.** Tailoring a large organizational standard process can become burdensome for small projects. Often, VSEs do not even have a documented process. The process is 'in the head' of the practitioners.

Figure 2-9 Challenges in adapting the CMM to small organizations (adapted from SPC 1998)

2.7 The Capability Maturity Model Integration

Many other Capability Maturity Models, such as the Systems Engineering CMM illustrated in Figure 2-10, the Acquisition CMM, the People CMM, and the Integrated Product Development CMM, have been developed by the SEI. This led industry and some government agencies to request, from the SEI, the development of a common architecture for all CMMs and a single model for the system and software engineering of large systems. The Capability Maturity Model Integration (CMMI) is the result of that request.



Level	Focus	Process Areas
5 Optimizing	*Continuous Process Improvement*	Organizational Innovation and Deployment Causal Analysis and Resolution
4 Quantitatively Managed	*Quantitative Management*	Organizational Process Performance Quantitative Project Management
3 Defined	*Process Standardization*	Requirements Development Technical Solution Product Integration Verification Validation Organizational Process Focus Organizational Process Definition + IPPD Organizational Training Risk Management Integrated Project Management +IPPD Decision Analysis and Resolution
2 Managed	*Basic Project Management*	Requirements Management Project Planning Project Monitoring and Control Supplier Agreement Management Measurement and Analysis Process and Product Quality Assurance Configuration Management
1 Initial		

Figure 2-10 Capability Maturity Model Integration (SEI 2006)

Figure 2-11 lists the desirable characteristics of an ideal organization implementing the CMMI (Garcia et al. 2007). It is clear that most VSEs are far from this ideal. This means that considerable effort has to be invested in trying to 'fit' the CMMI to VSEs.

- **Strategy**
  - To increasingly make organizational effectiveness a priority
  - To improve the effectiveness of processes to achieve better performance
- **Reward system**
  - The organization rewards participation in overall efficiency over individual department efficiency.
  - The organization rewards improvement in skills related to process management and support.
  - The organization rewards fire prevention more than firefighting.
  - The reward system will support incentives for high-performance teams (if integrated product and process development is to be implemented).
- **Sponsorship**
  - Consistent support for improving old practices or implementing new ones is exhibited by the organization's leadership.
  - Penalties for avoiding the new practices are consistently applied.
  - Sponsorship is sufficiently broad to include all processes and activities to be improved.
- **Values**
  - Measurements are used to improve the organization's performance, not punish individuals.
  - Participatory management is encouraged.
  - Mistakes are tolerated, as long as they lead to improved processes/performance.
  - Long-term improvement is worth short-term effort, and the ROI comes later.

- **Skills**
  - Project planning and management skills (enough to manage a process improvement project) are available.
  - Organizational change management skills are available.
- **Structure**
  - Clear definition of roles/ responsibilities/authorities exists.
  - Management is a role that is responsible for effectiveness of the processes in use within the organization.
  - Activities can be rationalized and organized around the concept of projects.

Figure 2-11 Desirable characteristics of an ideal organization implementing the CMMI (Garcia et al. 2007)

2.7.1 Difficulties in using the CMMI

Like the issues surrounding the use of the SW-CMM listed above, the CMMI is perceived as a difficult framework for VSEs, most of which are facing some of the following issues (adapted from *(*Kulpa et al. 2008) and (Staples et al. 2007)):

- Product quality assurance versus process quality assurance (Staples et al. 2007):
  - Many of the CMMI process areas and goals are concerned with process quality and quality assurance. VSEs perceive them as costly and time-consuming lower-value activities.
  - VSEs have a more specific interest in product quality (the product working well) than product quality assurance (evidence of the product working well).
- The CMMI tends to suit the requirements of large organizations (Staples et al. 2007):
  - VSEs need the CMMI to be tailored to suit their needs.
  - Tailoring creates additional burdens for them, since VSEs are likely to have fewer resources and expertise available to do the tailoring.
- VSEs consider themselves too small for the CMMI (Staples et al. 2007):
  - VSEs have concerns either about the perceived applicability of the CMMI or about constrained resources (available budget or time).
- No guidance is given for tailoring the CMMI to VSEs (Kulpa et al. 2008):
  - The CMMI, relying heavily on practices maintained by large organizations, is more difficult to implement in VSEs than the SW-CMM.
  - The tailoring guidelines given are called 'Discipline Amplifications'. They reside in the margins of Process Areas and consist of one or two sentences. These are often not clear or detailed enough for VSEs.

- The CMMI is too big for VSEs to handle (Kulpa et al. 2008):
    - VSEs are intimidated by the size of the framework. The CMMI has over 600 pages.
    - The CMMI contains more practices than the SW-CMM, and most of those are not detailed enough to be understood in such a way as to promote consistent application across projects.
    - Writing procedures is a long and difficult process. The time, resources, and costs associated with implementing the CMMI appear to have expanded, compared to the already major investment required by the CMM.
- The Return on Investment (ROI) from the CMMI has not been validated, especially as it relates to VSEs (Kulpa et al. 2008):
    - Some organizations have reported that it took three to five years to realize its ROI from following application of the SW-CMM. VSEs do not have the budgets and overhead of the larger organizations, and must worry about meeting payroll every two weeks.
    - No real studies are available (at this time) that can help VSEs calculate ROI from using the CMMI.
- VSEs may perceive that the CMMI emphasizes systems engineering over SE (Kulpa et al. 2008):
    - Systems engineering may not be part of their work, and may not apply.
- The CMMI is too prescriptive for VSEs (Kulpa et al. 2008):
    - The CMMI is structured so similarly to large, bureaucratically controlled efforts that there seems to be little room to manoeuvre. For example, Verification and Validation may be too rigorous for VSEs.
-

As explained in more detail in a next chapter, a category of VSEs, called 'startup' VSEs, have a set of characteristics which makes it very difficult for them to think about using the CMMI. As mentioned by Sutton, startups often lack the foundations required for success with the CMM—a historical record of experience, infrastructure for process instantiation and improvement, and repeatable and predictable practices (Sutton 2000).

These maturity models are introduced in the section on standards, since the CMM and CMMI have achieved such worldwide acceptance and usage that they are called de facto standards. I have included them, since they are going to be used in subsequent chapters to illustrate their use for process assessment and process improvement activities. Finally, since many industrial and governmental organizations are either using the CMMI in their operations or in contracted projects, or imposing it, the author has proposed, as explained in more detail later, the mapping of the CMMI practices to the future ISO Standards and Technical Reports that will be used by VSEs. This will put VSEs in a position to claim to potential customers that they have some level of conformity to either ISO standards, such as ISO 12207 or ISO 9000, or to the CMMI model.

2.8 Benefits of standards

Standards do not guarantee defect-free software, but they increase the likelihood of producing the required software on time and within budget. Figure 2-12 lists some of the benefits of using standards. It has to be noted that these benefits have been reported by SMEs, not VSEs, and, as explained in a subsequent chapter, most VSEs do not see the benefits of using standards.

- Establish uniform requirements and a vocabulary for development and documentation
- Define a common framework for software life cycle processes
- Clarify the roles and interfaces of participants
- Clarify the types and contents of documentation
- Identify the tasks, phases, baselines, reviews, and documents needed
- Follow the lessons learned and proven (best) practices of the industry
- Avoid the pitfalls and problems of the past
- Save time and money by not reinventing the wheel
- Select a supplier or a developer (e.g. ISO 9000)
- Impose requirements in a contract
- Impose an invisible trade barrier
- Use in a court of law by a customer to prosecute a faulty supplier

Figure 2-12 Advantages of standards

2.9 Very Small Enterprises[19]

The definitions of small and very small enterprises are challengingly ambiguous, as there is no commonly accepted definition of either. For example, the participants in the 1995 CMM tailoring workshop (Ginsberg et al. 1995) could not agree on what 'small' really means. Subsequently, the 1998 SEPG Conference Panel on the CMM and small projects (Hadden, 1998), the latter was described as lasting '3-4 months in duration with 5 or fewer staff'.

[19] This section is adapted from an article originally published in *(Laporte et al. 2008c)*

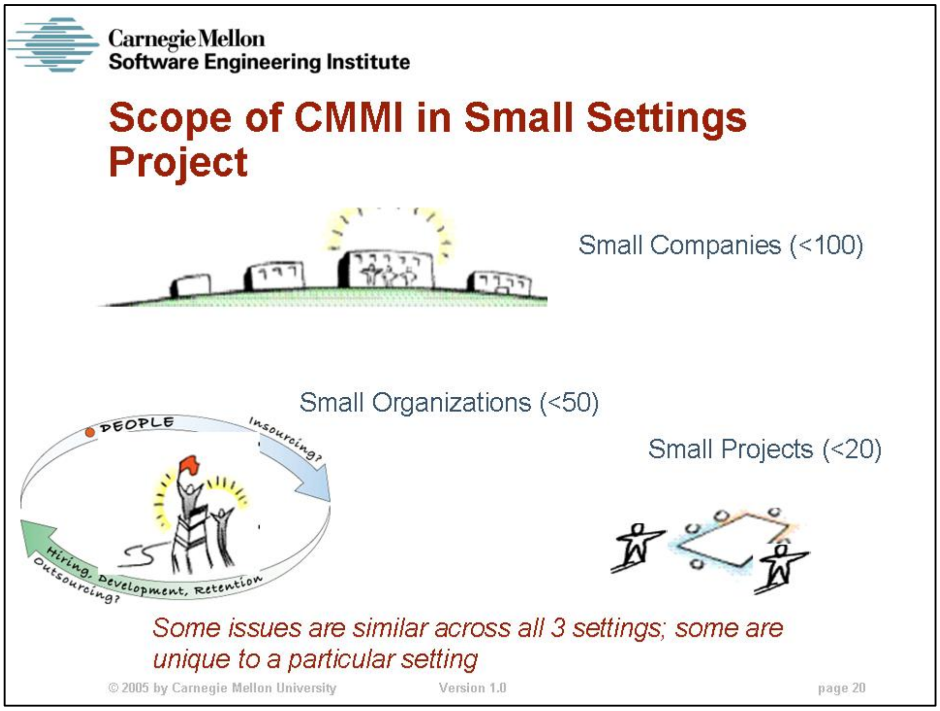
As defined by the US Small Business Administration, depending on the type of product called for by the contract, a company may qualify as a small business as long as it, or its affiliates, does not employ more than a specified number of employees (i.e. usually not more than 500, 750, or 1,000) (Estrin et al. 2003).

Johnson and Brodman (Johnson et al. 1998) defined a small organization as having 'fewer than 50 software developers' and a small project as having 'fewer than 20 software developers'.

From a legalistic perspective, the European Commission (EC 2005) defined three levels of small to medium-sized enterprises (SMEs) as follows:

> *Medium-sized —"employ fewer than 250 persons and which have an annual turnover not exceeding 50 million Euro, and/or an annual balance sheet total not exceeding 43 million Euro"; Small—"which employ fewer than 50 persons, and whose annual turnover or annual balance sheet total does not exceed 10 million Euro"; and Micro—"which employ fewer than 10 persons and whose annual turnover does not exceed 2 million Euro."*

The SEI initiated a project on the use of the CMMM in small settings. In a presentation made by Garcia to SPIN in Montréal in 2005 (see Figure 2-13), titled Thoughts on Applying CMMI in Small Settings, three different sizes can be used to qualify a project as a 'small setting'.



Carnegie Mellon
Software Engineering Institute

Scope of CMMI in Small Settings Project

Small Companies (<100)

Small Organizations (<50)

Small Projects (<20)

PEOPLE
Insourcing?
Hiring, Development, Retention
Outsourcing?

Some issues are similar across all 3 settings; some are unique to a particular setting

© 2005 by Carnegie Mellon University Version 1.0 page 20

Figure 2-13 Definition of small settings (Garcia 2005)

In a more recent SEI document, titled Prototype for a Field Guide for Improving Processes in Small Settings (Garcia et al. 2008), the following question was raised: What is small anyway? The document discusses the criteria that could be used to define a small setting, such as: number of employees, sales, investment level, and total assets. The SEI selected an investment level in process improvement as a criterion for a small setting, as follows:

> *From a practical perspective, we consider small to be any business unit for which the typical activities and approaches of process improvement could cost more than ten percent of the annual operating budget. Ten percent is something of an arbitrary number, but in talking to many organizations about their overall costs related to process improvement, this seems to be a "break point" beyond which it is difficult for an organization to justify the expense without a strong, usually external, incentive.*

This criterion is totally out of range for the small enterprises targeted by the ISO working group, since most of them supported by centers like CETIC in Belgium cannot afford even 25% of the process improvement investment proposed by the SEI.

To better understand the dichotomy between the definitions above, it is necessary to examine the size of software companies operating in the market today. In Europe, for instance, 85% of the IT sector's companies have 1 to 10 employees[20]. In the context of indigenous Irish software firms, only 10 companies out of a total of 630 (1.9%) employed more than 100 people, while 61% of the total employed 10 or fewer, with the average size of indigenous Irish software firms being about 16 employees (Coleman et al. 2008).

Therefore, for the purposes of this thesis and based on the above discussions, I am adopting the definition of a VSE introduced in (Laporte et al. 2006b), which is 'an entity (enterprises, organizations, departments, or projects) having up to 25 people'.

2.10 Characteristics of VSEs

Even though various definitions of terms appear in the literature, such as 'small', 'medium', and 'very small', I and the members of the ISO working group have assembled a list of the characteristics that describe a typical VSE (see Figure 2-14).

- Not necessarily software centric
    - Software development may or may not be a core competency.
    - VSEs struggle to understand how software can be leveraged for a business' benefit and how to be good software users.
- Small staff
    - Single IT person (maybe)
    - Typically very busy
    - Reticent to add additional IT staff
- Lack of appropriate internal expertise
- Micro companies lag in using Standards (NORMAPME 2003)
    - Failure to understand the value of the CMMI
    - Lack of human resources
    - Lack of funds
- Small companies have a flat structure.
- Buying expertise is too expensive.
- Problems regarding standards and standardization (EC 2006)
    - Lack of information on which standards have to be met
    - Difficulties in applying standards correctly
    - Difficulties in obtaining certification of compliance with standards
    - Lack of the possibility of participation in the development of new standards
- Paperwork is a productivity killer and an endless chore (ISO 2005).
- Limited leadership time
- SMEs consider standards as a burden, made by and for large enterprises, and do not involve them (EC 2006).
- Dynamic and flexible company (Hofer 2002)
- Employees often work overtime (Hofer 2002).
- Extremely responsive and flexible (competitive advantage) (Richardson et al. 2007)
- Lack of explicit understanding of work practices
- SMEs are aware of the importance of standards, but often find them complex, lack of knowledge about them, and have difficulty implementing them (NORMAPME 2003).
- Tasks are not continuously documented (Hofer 2002).
- SMEs are often too busy to engage in standardization (ISO 2006b).
- The CMMI is too prescriptive for VSEs (Kulpa et al. 2008).
- SMEs lack sufficient information to make effective use of the standards available to them

[20] http://www.esi.es/en/main/iitmark.html

(7).

- No guidance is given for tailoring the CMMI for VSEs (Kulpa et al. 2008).
- Micro enterprises receive less information about standards (Zoetermeer 2006).
- SMEs risk not being properly taken into account in the resulting standards, which they only learn about after their publication as national standards (Zoetermeer 2006).
- Small organization remains unaware of these methodologies (CMMI, ISO15504) (Richardson et al. 2007).
- VSEs consider themselves too small for the CMMI (Staples et al. 2007).
- Startup VSEs have a set of characteristics which make it very difficult for them to consider using the CMMI (Sutton 2000).
- SMEs and craft enterprises are often not aware of what is going on in standardization and of its importance. (Zoetermeer 2006).
- SMEs consider standards as a burden, made by and for large enterprises, and do not concern them (Zoetermeer 2006).
- SMEs often lack personnel, financial, and time resources (Zoetermeer 2006).
- Large and medium-sized enterprises are generally more active in offering training to employees than micro and small enterprises (Europe 2000).
- The majority of micro and small businesses and sole proprietorships (70%) would prefer to use support services that are specifically targeted toward enterprises of their respective size class (EC 2002).

Figure 2-14 List of characteristics of small organizations or projects.

The unique characteristics of small entrepreneurial businesses, as well as the uniqueness of their situations, necessarily make their style of business different (Anacleto et al. 2004). Some of the unique differences between the behavior of very small and large businesses are given in Table 2-7.

Characteristic	**Small firm**	**Large firm**
Planning orientation	Unstructured/operational	Structured/strategic
Flexibility	High	Structured/strategic
Risk orientation	High	Medium
Managerial process	Informal	Low
Learning and knowledge absorption capacity	Limited	High
Impact of negative market effects	More profound	More manageable
Competitive advantage	Human capital-centered	Organizational capital-centered

Table 2-7 Characteristic differences between large firms and small firms (Laporte et al. 2008d)

Software VSEs are subject to a number of distinctive and intrinsic characteristics that make them different from their larger counterparts, which affects the contents, the nature, and the extent of their activities. We classify VSE characteristics in four main categories: Finance, Customer, Internal Business Processes, and Learning and Growth:

- VSEs are economically vulnerable, as they are driven by cash-flow and depend on project profits, so they need to complete projects within budget. They tend to have low budgets, which has many impacts, such as: lack of funds for corrective post delivery maintenance; few resources for training; little or no budget for quality assurance activities; no budget for software reuse processes; limited budget for responding to risks; and limited budget for process improvement and/or obtaining an assessment or certification.

- The VSE's product typically has a single customer, who is in charge of the management of the system, and of software integration, installation, and operation. It is normal practice for the customer not to define quantitative quality requirements, and for customer satisfaction to depend on the fulfillment of specific requirements which may change during the project. A close relationship among all the project members involved, including the customer, shows that software development in small and very small companies is strongly human-oriented, and communication among them is important. For example, in contrast to small companies, very small companies often do not have regularly scheduled project meetings (Hofer 2002).
- The internal business processes of VSEs are usually focused on developing custom software systems, where the software product is elaborated progressively, and typically do not have strong relationships with other projects. Typically, most management processes (such as human resources and infrastructure management) are performed through informal mechanisms, with the majority of communication, decision making, and problem resolution being performed face to face.
- The learning and growth characteristics of VSEs are typified by a lack of knowledge (or acceptance) of software process assessment and improvement, and a lack of the human resources required to engage in standardization. It is usual for them to have a negative perception of standards as being made by and for large enterprises (Coleman et al. 2008).

Additional characteristics are presented in the chapter on the survey of VSEs.

2.11 Why develop software engineering standards for VSEs?

As explained above, most SE standards have been instigated or developed by large organizations or communities, such as the military, the nuclear industry, and the aerospace industry.

Today, the ability of an organization to compete, adapt, and survive depends increasingly on software. As pointed out in chapter 1, by 2010, it is estimated that a cellular phone will contain 20 million lines of code, and an automobile manufacturer has estimated that the software in its cars will contain up to 100 million lines of code by that time (Charette 2005).

However, because of the increasing size of modern systems, a single supplier cannot develop all the software required in an acceptable time period. As a result, many of the subsystems have to be developed by SMEs, which in turn will subcontract components to VSEs.

A manufacturing chain of large mass market products often has a pyramidal structure, as illustrated in Figure 2-15. For example, a large Japanese mass product manufacturer integrated a part into one of its products that had been fabricated by one of its 6,000 lower-level producers which contained an undetected software error. This defective part resulted in a loss of over $200 million by the mass product manufacturer (Shintani 2006).

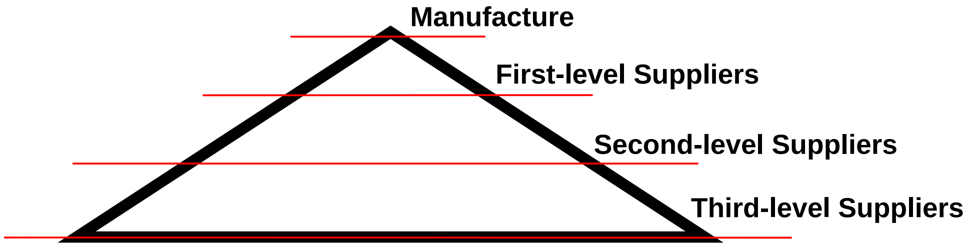


Figure 2-15 A supply chain of a large manufacturer (adapted from Shintani 2006)

In another example, Boeing spent roughly $800 million for the development of the software for the 777's 1,280 onboard processors and more than 4 million lines of Ada code. Boeing might spend five times that amount for the software of the new Boeing 787 (Long 2008). A quick calculation shows the number of staff-years required to develop the software for the 787 using the benchmarking data from Reifer (Reifer 2004):

- $800 million (for the Boeing 777) X 5 (for the Boeing 787) = $4 billion
- Number of lines of code (for the airborne domain) = $4 billion/$200 per line = 20,000,000 lines of code
- Number of staff-months = 20,000,000 lines/100 lines per staff-month = 200,000 staff-months
- Number of staff-years (based on 10 months of work per year) = 200,000/10 = 20,000 staff-years

I do not think that Boeing can develop this quantity of software in-house within a reasonable time frame. The company will have to ask many suppliers to develop the software. I think it is very likely that many VSEs will be providing subsystems or components to Boeing or its main suppliers, who will integrate the components developed by a VSE to integrate them into a higher level component or subsystem.

For a modern airplane, in addition to the standard propulsion, navigation, and communication systems, software needs to be developed for the following components: seats, lavatory control systems, entertainment systems, galley inventory control systems, lighting, weather radar, communication-surveillance, audio, cargo smoke detection, gust/wind shear detection, weight, and balance. SMEs and VSEs are often involved in the development and maintenance of these components. Failure of a component may cause serious performance degradation and even failure of a critical system. The manufacturer must ensure that the components supplied by VSEs are defect-free (i.e. bug-free) and provide the functionalities and performance specified in the requirements. Standards can help the manufacturer by imposing requirements on the development and maintenance processes of VSEs. Standards do not guarantee defect-free software, but increase the likelihood of producing the required software on time and within budget.

2.12 Conclusion

Software engineering standards are indispensable for the development of modern systems. But, to be able to benefit from the production of the multitude of VSEs worldwide, we must find a way to provide them with standards that can be readily used and understood. In the next chapter, the author describes the centers and initiatives focusing their activities mostly on Small and Very Small Enterprises in the software domain.

Chapter 3

Overview of Process Improvement Initiatives for Small Entities

3 Chapter 3 – Overview of Process Improvement Initiatives for Small Entities

3.1 Introduction

Industry recognizes the value of Information Technology VSEs in contributing valuable products and services. In Europe, for instance, 85% of the IT sector's companies have 1 to 10 employees[21]. In Canada, a survey of the Montreal area revealed, as illustrated in Table 3-1, that 78% of software development enterprises have fewer than 25 employees and 50% have fewer than 10 employees (Laporte et al. 2005a). In Brazil, small IT companies represent about 70% of the total number of companies (Anacleto et al. 2004).

Size (number of employees)	**Software Enterprises**		**Jobs**	
	Number	**%**	**Number**	**%**
1 to 25	540	78%	5,105	29%
26 to 100	127	18%	6,221	36%
over 100	26	4%	6,056	35%
TOTAL	**693**	**100%**	**17,382**	**100%**

Table 3-1 Size of software development companies in the Montreal area (Laporte et al. 2005a)

Figure 3-1 illustrates the spectrum of software development enterprises in terms of the cost of their labor, their quality, and their productivity. In order to help software enterprises move to the lower-right quadrant, many countries have decided to develop incentive programs of various kinds.

[21] http://www.esi.es/en/main/iitmark.html

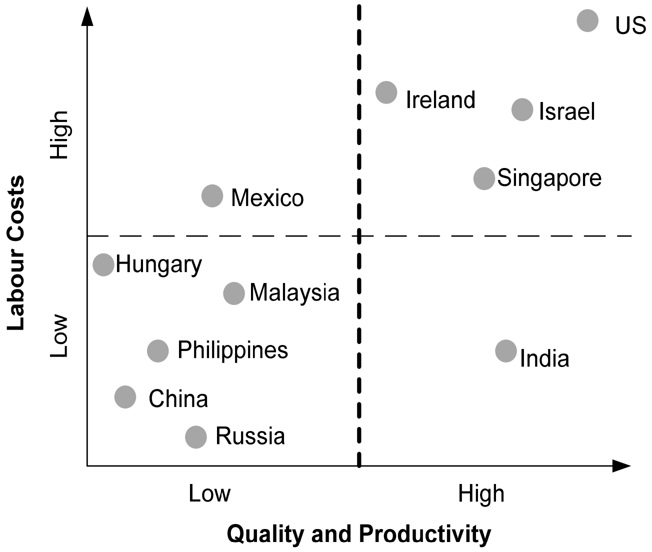


Figure 3-1 Labor costs, quality, and productivity (McCaffery et al. 2006)

In phase 2 of the 6-phase model (Rogers 2003) illustrated in Figure 3-2, titled Basic and Applied Research, a series of initiatives targeted at very small, small, and medium-sized IT Enterprises (SMEs) and VSEs were investigated and are presented below.

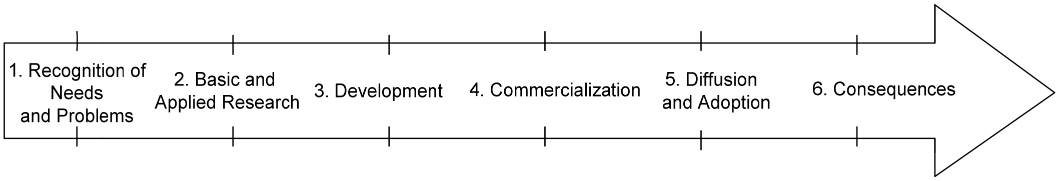

1. Recognition of Needs and Problems
2. Basic and Applied Research
3. Development
4. Commercialization
5. Diffusion and Adoption
6. Consequences

Figure 3-2 Phases of the Innovation Process (Rogers 2003)

3.2 Centers and initiatives focusing on small and very small software enterprises[22]

3.2.1 Introduction

In this section, we describe the work performed by a few centers and initiatives which focus their activities on small enterprises and VSEs. Most software engineering research centers, such as the SEI, dedicate their resources mainly to large organizations. Even though there seems to be an awareness of the need for VSE solutions, they are still for the most part unusable by companies with 25 or fewer employees.

We discuss the objectives and accomplishments of these activities in helping VSEs become more competitive, since WG24 will try to benefit from the experience gained by these centers. The centers and initiatives described in this chapter are grouped in four regions: Europe, Asia, Australia, and the United States.

[22] This section is adapted from an article published in (Laporte et al. 2008c)

3.3 Centers and Initiative in Europe

3.3.1 Irish Software Engineering Research Centre (LERO)[23]

The Irish Software Engineering Research Centre (LERO) was established in November 2005 with support from Science Foundation Ireland's CSET (Centre for Science, Engineering and Technology) program. LERO focuses on specific domains, especially those where reliability is crucial, including the automotive, medical device, telecommunications, and financial services domains. They develop models, methods, and tools that make it cheaper, faster, or easier to produce this critical software. The vision, mission, and goals of LERO are listed in Table 3-2. Two LERO researchers are members of WG24, and are actively participating in conducting the pilot projects described in a subsequent chapter.

LERO's vision is of Ireland as a world leader in the development of high-quality software-intensive systems of high economic and social value.
LERO's mission is to be a world leader in the development of research in software engineering, with a special emphasis on Evolving Critical Systems. LERO will enhance the quality and competitiveness of the Irish software industry through shared projects, knowledge transfer, and education
LERO's goals: **Establish a sustainable national centre** 1. Establish the LERO brand in Ireland and abroad 2. Operate the centre efficiently, effectively, and transparently 3. Develop and implement a long-term business plan for the centre 4. Build up and maintain strong national university-industry research links 5. Have an impact on national software engineering education and training **World-class research** 6. Produce internationally recognized research outputs 7. Establish close links with international research institutions **Focus on strategic industrial domains** 8. Tackle research problems of industrial relevance 9. Validate and improve research results with industry 10. Help Ireland attract software engineering R&D

Table 3-2 LERO's vision, mission, and goals

3.3.2 Centre for Software Process Technologies of Ireland

The Centre for Software Process Technologies (CSPT)[24] is a research and knowledge transfer organization hosted by the Faculty of Engineering at the University of Ulster. Its activities cover a wide range of areas affecting the quality and effectiveness of both software development processes and products, from process measurement, through business process co-evolution, to object-oriented software complexity metrics.

The CSPT recently published the results of its first six assessments in small and medium-sized enterprises (SMEs) using its Express Process Appraisal (EPA) method (Wilkie et al.

[23] Adapted from http://www.lero.ie/

[24] http://www.infc.ulst.ac.uk/informatics/cspt/

2005). EPA is a class C method, which complies with the appraisal requirements for the CMMI® (SEI 2002). The EPA model assesses six of the seven process areas at maturity level 2: Requirements Management, Configuration Management, Project Planning, Project Monitoring & Control, Measurement & Analysis, and Process & Product Quality Assurance. The authors report that the EPA method requires approximately 45 person-hours of the appraised organization's time and 42 person-hours of the CSPT appraisal team's time over a two-week period.

The CSPT also published a paper (McFall et al. 2003) in which McFall presents the various perceived priorities and concern areas for different sizes of organization. As illustrated in Figure 3-3, the priorities and concerns of organizations with fewer than 20 employees are quite different from those of larger organizations. As an example, for small organizations, managing risk is of great concern, while for larger organizations risk management only ranks as priority number 8. Conversely, for small organizations, consistency across teams is less of a concern, while for larger organizations it is the top-priority issue.

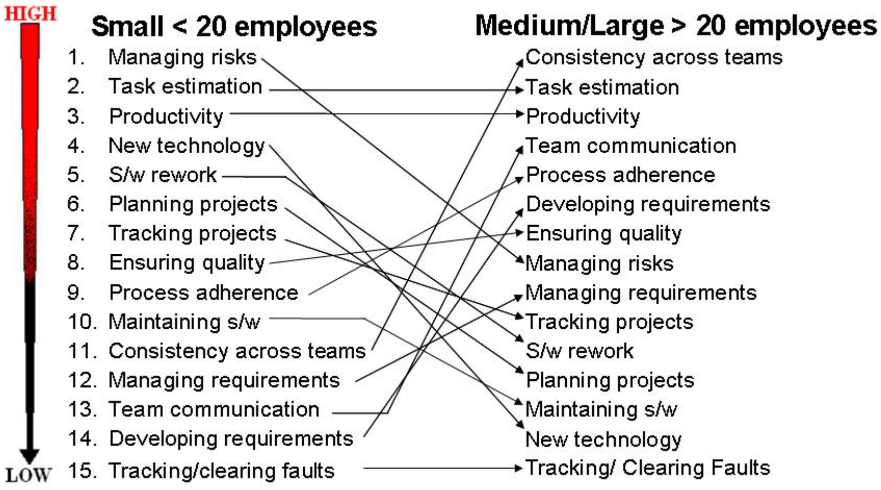


Figure 3-3 Priority and concern differences based on organization size (McFall et al. 2003)

3.3.3 The British "Toward Software Excellence" Initiative

Toward Software Excellence (TSE)[25] provides a self-assessment "health check" facility and corresponding guidance on best practices (the Route Map) and is based on ISO/IEC 15504.

It proposes an interesting mix of functionalities and characteristics that can explain a small organization's success: it uses business language and addresses the business perspective of

[25] http://www.softwareexcellence.co.uk/

the process issue, and is aimed at solving business problems first and highlighting the importance of customer relationships.

TSE is, in fact, much more than an assessment tool, as it helps to explain issues to people using a language they can understand. This tool (with the Belgian Gradual Framework) is probably a good example of how to make tough topics accessible to all.

3.3.4 The European Software Institute – IT Mark

The European Software Institute (ESI) is a technological center whose aim is to contribute to developing the Information Society and to increasing industry competitiveness by means of knowledge, innovation, continuous improvement, and the promotion and dissemination of IT.

The ESI commercially promotes and delivers the ESI's products and services to the European and Latin-American market in the first phase, and worldwide in the second phase. It has established a network of partners, called ESI@net, with companies whose activities are related to SPI (Software Process Improvement) and IT in general.

The ESI Centre Alliance has launched IT Mark Certification[26] worldwide, which is aimed at certifying the quality and maturity of the processes in SMEs (up to 250 workers) that develop and maintain IT Systems. IT Mark assesses and certifies the quality of SMEs in three main areas: Business Management (overall management: strategic, commercial, financial, marketing, etc.); Information Security Management; and Software & Systems Processes (maturity).

In matters relating to Business Management, the reference used is the 10^2 model that was developed to assess Venture Capital applications. For Information Security Management, the reference model used is ISO 17799 (ISO 2005), and for Software & Systems Processes a lightweight version of the CMMI® (SEI 2006) is used.

3.3.5 Finland's SataSPIN

SataSPIN (Varkoi et al. 1999) is not a methodology or a solution for assessing or improving software processes, but rather a regional network for software SMEs wanting to make improvements. The authors' main goal is to set up an SPI program in each of the participating companies with a view to establishing a network of companies promoting good software practices in a region.

The project provides the participating companies with training and consultation on subjects related to software processes. An essential part of the project is process assessment.

[26] http://www.esi.es/en/main/iitmark.html

Companies can also obtain assistance in planning and implementing the improvements. Training activities within the project are targeted to supporting the improvement of software processes and the enhancement of personnel skills. All the activities are tailored separately for each company to ensure flexibility in participation and alignment with business goals.

The core of the SataSPIN project is to help software SMEs to develop their operations using international software process models. The project uses the ISO/IEC 15504 standard as the software process assessment tool and improvement framework (which requires public funding as an enabler). The project is based on the cooperation of the participating enterprises, and offers a wide variety of courses and seminars in the area of software engineering and management.

SataSPIN[27], located at Tampere University of Technology of Finland, offers the following services:

- training and expert services in software process improvement;
- process assessments, improvement planning, results evaluation;
- training and expertise in software management, methods, and technologies.

Two SataSPIN projects have involved 20 software SMEs and over 400 IT professionals.

3.3.6 NORMAPME

NORMAPME[28] is the European Office of Crafts, Trades and SMEs for Standardization. It is an international non-profit association created in 1996 with the support of the European Commission, and the only European organization focusing on small enterprise interests in the European standardization system. Its members represent over 11 million enterprises in all European countries, including all EU and EFTA member states, and its mission is to defend the interests of all of them. This mission is of crucial interest, as SMEs represent over 90% of European companies and they employ nearly 81 million people, which is 66% of Europe's total work force.

Standards are essential for SMEs today, as they are for any company operating in an internal market. The adoption of standards guarantees them several advantages, such as enlarging the potential market for products, facilitating product acceptance, lowering transaction costs, achieving economies of scale, reducing external effects (like environmental impact), interoperability, improving management systems, and so on. Thus,

[27] http://www.tut.fi/public/

[28] http://www.normapme.com/

standards definition cannot be a privilege enjoyed by large companies alone. SMEs must be represented. However, SMEs lack knowledge with respect to standards and standardization, and they need some support to help them implement existing standards, as well as have a voice in the standardization process.

The European Commission has supported NORMAPME during its first years of operation. Currently, NORMAPME is party to an EC contract offering standardization services to SMEs.

The principal and most important activity of NORMAPME is participation in the standardization process: experts recommended by member SME organizations participate in the work of technical committees associated with the European standardization organizations (CEN, CENELEC, ETSI) and at the ISO.

Secondly, NORMAPME collects information on new directives, directives under review, and standardization works. Essential parts of this information are published in simple language by means of newsletters, specific circulars, a Web site, seminars, and the like. All publications are translated into six languages: English, French, German, Spanish, Italian, and Polish, to ensure accessibility to the largest number of Europeans.

NORMAPME members, and all SMEs and their organizations, have the opportunity to formulate proposals for the improvement of standards and directives. These opinions are debated in the expert groups in order to draft SME representative positions. Once these positions are finalized, they are promoted in the standards organizations, in European institutions, and through the media by publishing articles and through the press.

3.3.7 The ESPRIT – ESPINODE Initiative

An assessment methodology has been developed by ESPINODE for Central Italy, with the aim of using rapid software process assessment as a way to promote innovation for SMEs (Cignoni 1999). The methodology is based on a two-part questionnaire compiled by experts who interview representatives of the enterprise. Part 1 is conducted by phone and Part 2 is completed in a direct audit meeting.

Rapid-assessment meetings allow enterprises to "taste" SPI, and awareness and training events are used as a way to establish the very first contact with the enterprises and to present the opportunity to take advantage of a rapid software process assessment as a free service. The specific goals of the subsequent assessment program are the following:

- to stimulate interest in software process assessment and improvement;
- to contribute to the definition of specific improvement plans;

- to collect data and statistics about software process maturity.

This methodology is quick, but it is also, consequently, approximate. Due to time constraints, the scope and accuracy of the assessment are sacrificed, since the assessment meeting is limited to half a day, including time for discussion. In particular, a very general assessment is made of the 35 processes, and some more accurate questions are formulated on just three processes in two of the five SPICE process categories. Moreover, the accuracy of the assessment is limited to the answers given by the enterprises, and those answers are neither cross-checked nor validated.

The rapid assessment procedure offered through awareness and training events shows that, in very many cases, identifiable benefits can be achieved via focused SPI projects.

The offer of a free (Rapid) assessment is a way to both diffuse process quality concepts and propose actual improvement paths to enterprises.

3.3.8 Centre d'Excellence en Technologies de l'Information et de la Communication of Belgium

The Centre d 'excellence en technologies de l'information et de la communication (CETIC)[29], located in Wallonia (Belgium), focuses on applied research and technology transfer in the field of software engineering and electronic systems. CETIC is a connecting agent between academic research and industrial companies. At the University of Namur, a software process improvement approach dedicated to small development structures has been developed. The method, called Micro-Evaluation, has been used and improved in collaboration with CETIC and the Department of Software and IT Engineering at the École de technologie supérieure (ÉTS) (Québec, Canada)[30].

Gradual framework[31]

In the first stage, a very simplified questionnaire, called the Micro-Evaluation, is used to collect information about the current software practices in small structures and to make people sensitive to the importance of software quality aspects. The questionnaire was mostly designed based on the Software Capability Maturity Model (Paulk et al. 1993) and on the ISO/IEC 15504 (SPICE) reference model, and uses an interview method. It covers six key axes selected on the basis of former experience with SME and VSE evaluation as the most pertinent and the most important to the targeted organizations. These axes are: quality management, customer management, subcontractor management, development and

[29] http://www.cetic.be

[30] http://profs.logti.etsmtl.ca/jmdeshar/SiteWQ/index.html

[31] This section is adapted from: Habra et al. 2008, Desharnais et al 2007 and Laporte et al. 2006a.

project management, product management, and training and human resources management.

The Micro-Evaluation was first tested on a sample of twenty organizations in Wallonia (Laporte et al. 2005b). Figure 3-4 shows the global maturity profile of the small enterprises involved in the first Micro-Evaluation round. Subsequently, seven of the twenty companies re-evaluated their practices and one performed a third Micro-Evaluation.

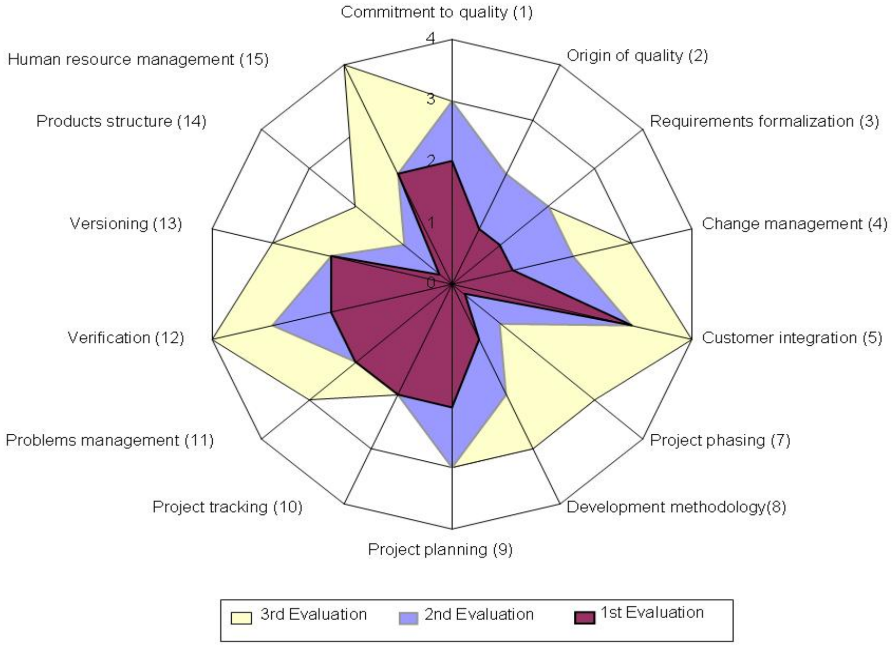


Figure 3-4 Evolution of profile over three Micro-Evaluations

In 2004, twenty-three micro-evaluations were performed in Québec, Canada. The average number of employees in the companies involved was about 13, and the average number of years the companies had been producing software was about 12. Figure 3-5 shows that small organizations were performing requirement formalization, project planning, problem management, and verification and versioning activities, with a score of about 3 out of a maximum of 4. A number of weaknesses can also be noted: very low scores on commitment to quality, change management, product structure, human resources management (i.e. training), and project tracking. It is also interesting to note that project planning scored significantly higher (3.0) than tracking. It seems that VSEs develop a plan, which, once in development, is forgotten while the “fire” of the day is put out.

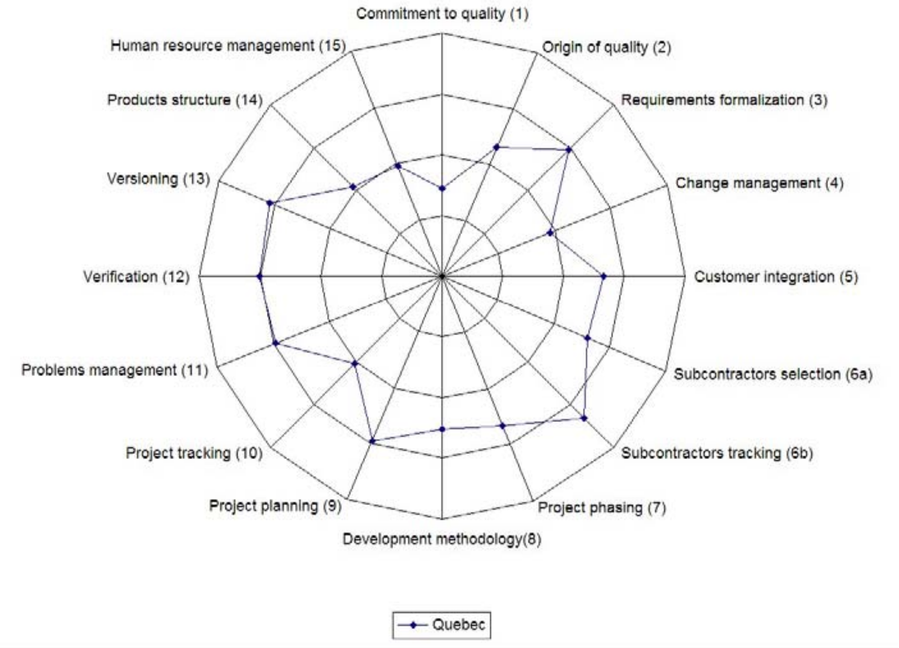


Figure 3-5 First Micro-Evaluation round in Québec

The ÉTS is currently conducting experiments with some of its graduate software engineering students. As part of their academic courses (Software Quality Assurance and The Case Study), they are required to perform evaluations, identify one or two practices to improve, and transfer the practice(s) to the organization. Since some of the students already work for VSEs, and so it is easy for them to sell their management on the idea of a small team of two or three students investing a few hundred hours, of their own time, into improving an area of the VSE development process.

The second step of the gradual approach of the OWPL (Observatoire Wallon des Pratiques Logicielles) is the assessment based on a light reference model adapted from the SW CMM. The OWPL model has been designed with respect to the particular context of small businesses, to help them improve their software practices. The structure of the OWPL model involves processes, practices, and success factors. It defines 10 processes (requirements management, project planning, project tracking and oversight, development, documentation, testing, configuration management, subcontractors management, quality management, and experience capitalization), each of which is decomposed into a number of practices (from 3 to 12). It is also supported by success factors. Each of the above processes is assigned a general goal in accordance with the organization's defined objectives, which involves a number of practices and is supported by a number of success factors. Each practice is defined by its goal, its inputs and outputs, the resources assigned to support it, and its weight. This last attribute is an indicator of the importance of the practice to improving the process as a whole.

3.4 Center and initiative in Australia

3.4.1 The Software Quality Institute of Australia

The Software Quality Institute, Griffith University (Australia), developed the **R**apid **A**ssessment for **P**rocess **I**mprovement for software **D**evelopment (RAPID) method in conformity with ISO/IEC 15504 (Rout et al. 2000). RAPID was developed for SMEs with limited investment of time and resources. The model includes eight ISO/IEC 15504 processes: Requirements Gathering, Software Development, Project Management, Configuration Management, Quality Assurance, Problem Resolution, Risk Management, and Process Establishment.

The scope of the model is limited to Levels 1, 2, and 3, although capability ratings at Levels 4 and 5 are possible. The organizations assessed in Queensland ranged in size from 3 to 120 employees, with an average size of 10 to 12 employees.

3.5 Centers and initiatives in the Americas

3.5.1 The Brazilian Approach[32]

An initiative, started in 2003 to improve software process, mainly in Brazilian SMEs, is titled MPS.BR Program[33]. It was an effort of the software industry and research institutions. The goal was to develop and disseminate a software process model aimed at reducing the SPI implementation costs and providing benefits in a short time frame. The model was based on international standards, such as ISO/IEC 12207 and ISO/IEC 15504, internationally recognized models, such as the CMMI®, and on Brazilian software industry business needs.

The MPS Model comprises three components:

- The MPS reference Model: composed of 3 guides: the MPS general guide, the MPS acquisition guide and the MPS implementation guide.
  - The process dimension of the model is made up of 7 groups of processes that correspond to 7 maturity levels.
- The MPS Assessment Method;
- The MPS business Model.

[32] This section is adapted from Montoni et al. 2008.

[33] MPS.BR is the acronym for the Portuguese *Melhoria de Processo do Software Brasileiro*.

Figure 3-6 presents the components of the model and the elements that constitute each component. The biggest difference between this model and other models, such as the CMMI®, is that an organization can escalate maturity levels more easily. In the MPS Model, few processes, compared to the CMMI®, need to be implemented at each maturity level. As an example, the first maturity level is composed of only 2 processes: requirement management and project management.

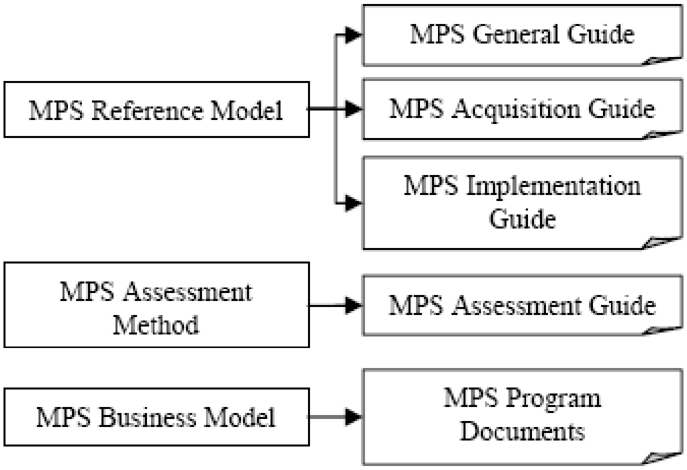


Figure 3-6 MPS model components (Montoni et al. 2008)

In 2007, it was reported that over 2,600 people attended a course on the model and 93 organizations were implementing it in 13 cities. A bank, the Inter-American Development Bank, provided 50% of the implementation and assessment costs on condition that the organization concluded the implementation within 12 months and conducted an assessment 3 months after that. By 2007, more than 17 Brazilian organizations had executed MPS model-based assessments in 9 cities (Rocha et al. 2007).

3.5.2 The Mexican Approach

In Mexico, it was felt that standards such as ISO/IEC 12207, or models such as the CMMI®, were either too general or too costly for Mexican enterprises. A Mexican standard was therefore developed at the request of the Ministry of the Economy. It provides the software industry there with a model based on international practices and on the following characteristics:

- It is easy to understand;
- It is easy to apply;
- It is economical to adopt.

It provides the basis on which to achieve successful evaluations with other standards or models, such as ISO 9000:2000 or the CMMI®.

The Mexican standard (NMX 2005) is divided into four parts: Part 1: Definition of Concepts and Products; Part 2: Process Requirements (MoProSoft); Part 3: Guidelines for Process Implementation; and Part 4: Guidelines for Process Assessment (EvalProSoft).

The process model

The process model MoProSoft uses ISO/IEC 12207 as a general framework. It was developed considering the integration of software processes and business processes, and borrows practices from ISO 9000:2000 and the CMMI®. It also incorporates practices from the Project Management Body of Knowledge (PMBOK) (PMI 2008) and the Software Engineering Body of Knowledge (SWEBOK) (ISO 2005a). In addition, MoProSoft addresses the Process Model requirements of ISO/IEC 15504-2 (ISO 2003). The percentage of coverage by MoProSoft with respect to these practices is as follows:

• ISO 9001:2000	92%
• ISO/IEC 12207 (Amendments 1 & 2)	95%
• CMMI® level 2	77%

MoProSoft focuses on processes, and considers three basic organizational or structural levels under which processes are organized: Top Management, Management, and Operations.

- The Top Management category contains the Business Management process. Its purpose is to establish the reason for the existence of an organization, its goals, and the conditions required to achieve them.
- The Management category consists of Process Management, Project Portfolio Management, and Resource Management.
- The Operations category consists of Specific Project Management and Software Development and Maintenance.

In addition, MoProSoft highlights informative data, added to the normative part, and proposes tailoring guides for each process. This is a very helpful feature and one requested by VSEs in the survey.

The Assessment Method

The Mexican standard also proposes Guidelines for Process Assessment, EvalProSoft, based on ISO/IEC 15504-2. The process assessment model defines five levels of capability and their associated attributes, as illustrated in Figure 3-7. For VSEs, WG24 will develop profiles, guides, and templates for capability levels 1 and 2. After reaching level 2, a VSE should be mature enough to make appropriate decisions about future improvement activities.

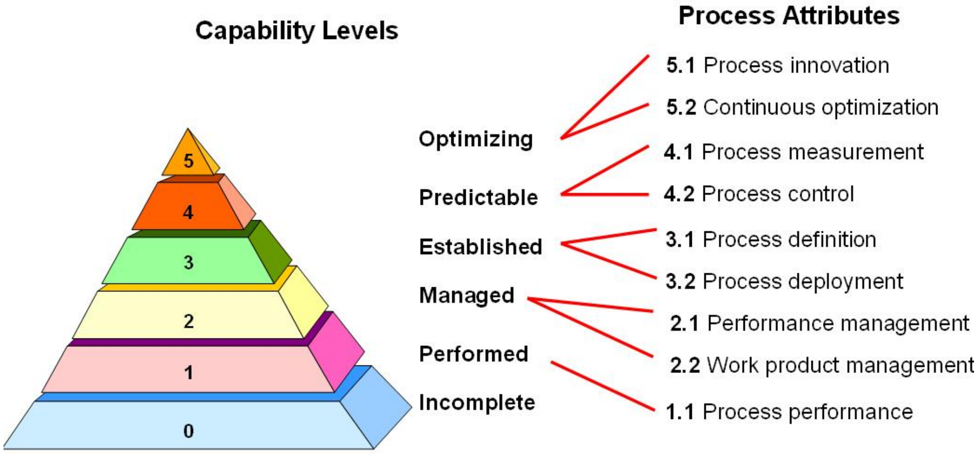
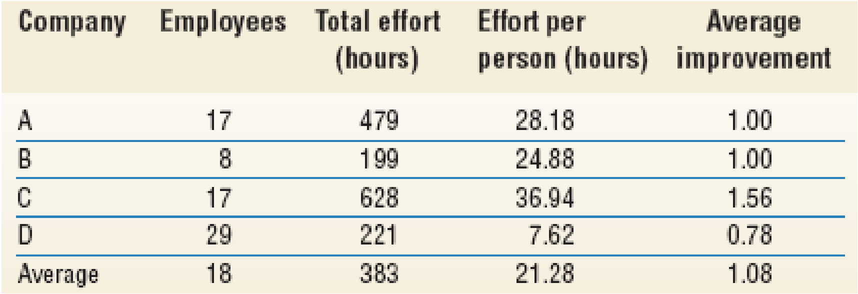


Figure 3-7 Capability dimensions of ISO/IEC 15504-2

In 2004, trials were run in four small Mexican software companies (Oktaba et al. 2007). An initial assessment showed that the firms were between level 0 and 1. After six months of coaching, the companies were assessed again. As illustrated in the last column of Table 3-3, the enterprises achieved an average increase of 1.08 in the capacity level of all their processes.



Company	Employees	Total effort (hours)	Effort per person (hours)	Average improvement
A	17	479	28.18	1.00
B	8	199	24.88	1.00
C	17	628	36.94	1.56
D	29	221	7.62	0.78
Average	18	383	21.28	1.08

Table 3-3 Improvement experience using MoProSoft (Oktaba et al. 2007)

3.5.3 The COMPETISOFT Project

Small and medium-sized enterprises (SMEs) account for about 90% of Latin America's businesses, and they generate about half of all employment (Oktaba et al. 2007). The COMPETISOFT project was initiated in 2005 by researchers and practitioners from many Latin-American countries, as well as from Spain and Portugal. The project was set up to provide the Latin American software industry with a framework for software process improvement and certification. Different initiatives were used as inputs, as illustrated in Figure 3-8, such as:

- the Mexican MoProSoft process model and EvalProSoft evaluation method;
- the Brazilian Process Improvement Model;
- the Columbian agile software process improvement;
- the Spanish Ministry of Public Administration's Métrica v3;
- the Ideal model and the CMMI® of the SEI;
- the 12207 and 15504 ISO standards.

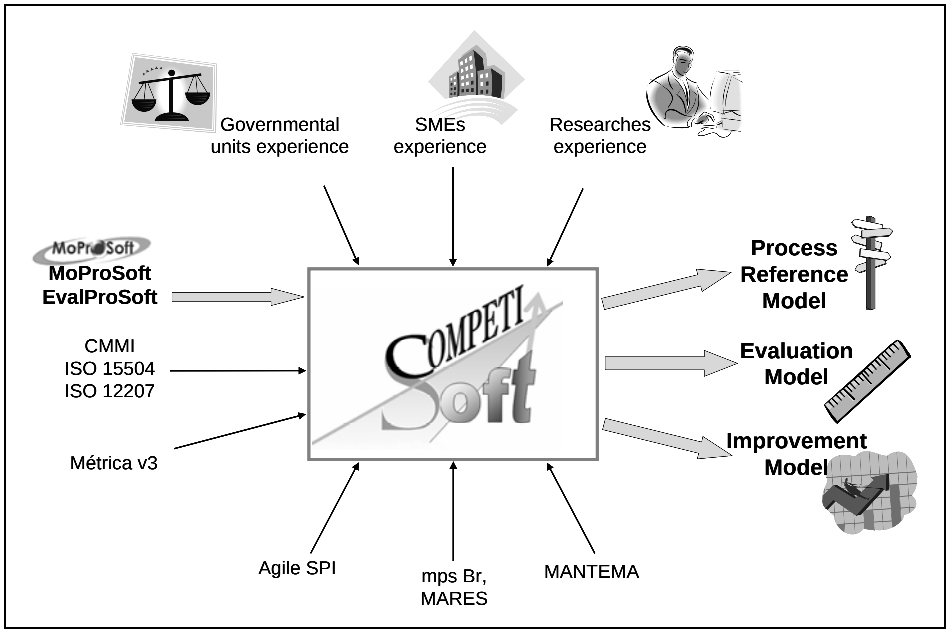


Figure 3-8 Overview of the COMPETISOFT project (Oktaba et al. 2007)

An objective of the COMPETISOFT project was to develop a new Process Reference model, a new Evaluation model, and a new Process Improvement model based on the frameworks listed above.

They have defined a process for managing and leading the improvement process called PmCOMPETISOFT. The process is based on the following principles (Pino et al. 2009):

- Early and continuous achievement of improvements,
- Continuous and rapid process diagnosis,
- Elemental process measurement,
- Effective group collaboration and communication,
- Continuous learning.

This process was influenced by the ISO 15504 standard, the IDEAL model of the SEI, and the SCRUM agile method. PmCOMPETISOFT is described in terms of purpose, objectives, roles, activity diagram, activities, work products, and tool support.

3.5.4 Software Engineering Institute of the USA

As stated in an earlier chapter, the Capability Maturity Models, such as the SW-CMM and the CMMI®, developed by the SEI were perceived as not being useful for most VSEs. In a draft guide (Garcia et al. 2008), the SEI described this situation as follows:

> *Because larger rather than smaller organizations have had more resources and finances to work with the SEI over the years, these technologies have inherently favoured the organizational infrastructures of large organizations. As a result, small organizations have complained that these technologies are too big, too complex, too costly, and too time consuming for the small business, small project, and small organizational unit, and they've asked the SEI for help.*

This situation has prompted the SEI to launch a project titled "Improving Processes in Small Settings (IPSS)"[34]. The IPSS project assembled small businesses, governments, large businesses, advocacy organizations, universities, and industry associations from around the world to jointly explore the unique challenges and opportunities of applying process improvement strategies in small businesses. The SEI sought to achieve the following objectives:

- increase awareness of process excellence as an enabler of global competitiveness,
- demonstrate effective approaches to process improvement for the small business,
- provide tools for process improvement that are easily applied by small businesses.

We should remember that the definition of a small business is 'any business unit for which the typical activities and approaches of process improvement could cost more than ten percent of the annual operating budget.' As stated previously, this definition does not correspond to the size of entities addressed by the ISO working group (i.e. fewer than 25 people).

The author became a member of this project in 2008,. The main output of this project was the 'Prototype for a Field Guide for Improving Processes in Small Settings' (Garcia et al. 2008). The guide, published as a draft version in 2008, is a collection of how-to guides, examples, templates, and checklists. I contributed to this document as a reviewer.

3.5.5 The ParqueSoft Organization of Columbia

The Software Technology Park Foundation[35] (Fundación Parque Tecnológico del Software), ParqueSoft, is a not-for-profit organization established in 1999 for the purpose of creating and developing enterprises providing goods and services to the IT market. ParqueSoft is consolidating south western Colombia's Science and Technology Corridor, by integrating 14 Software Technology Parks located in the following cities: Cali, Popayán, Pasto, Buga, Tuluá, Palmira, Buenaventura, Armenia, Manizales, Ibague, Villavicencio, Medellin, Sincelejo, and Pereira.

[34] http://www.sei.cmu.edu/iprc/ipss.html

[35] www.parquesoft.com

To date, ParqueSoft and its network of Parks house more than 250 VSEs where more than 1,000 software engineering professionals specializing in the industries' latest technologies, along with 200 other professionals, provide support in administrative and business development processes. These VSEs have, on average, 6 employees each.

ParqueSoft's goal for the year 2010 is to develop 400 competitive and productive IT enterprises which will export their software products and services to international markets, and create 4,000 jobs in an innovative science and technology sector contributing to the regional economy by more than US $100 Million annually (Arenas 2007).

ParqueSoft has created an innovative support model encompassing five macro objectives supported by 16 synergistic strategies to promote enterprise development and research and development (R&D). The five macro objectives and their corresponding strategies are listed in Figure 3-9.

1. To Provide Infrastructure for Business Development and Support
By supplying all the companies with logistical support to facilitate the development of their businesses. Some of these are:
Competitive physical and technological infrastructure infrastructure
Technological Support (Telco, Networking, Videoconference, Data Center)
Effective Communications (Internet, Intranet, and Media)
Intellectual Property and Legal Support

2. To Develop the Best People in the Industry
By developing a program for training and qualifying personnel, since world-class talent specializing and certified in specific areas of work is essential for the sector to be competitive in worldwide markets.

3. To Develop Financial Strength
By providing financial support to facilitate development of all the companies, some of these being: Entrepreneurship Promotion Funds, Risk Capital Funds, and Savings Funds.

4. To Support Enterprise Development
By providing support tools to facilitate the development of all the companies, some of these being: Market Intelligence, Creative Marketing, Business Knowledge, Business Development, and Business Support.

5. To Produce More Innovative, Reliable, and Competitive Products
By applying a Strategy Quality Program for creating and improving the software process improvement model and quality assurance set of practices for small ParqueSoft organizations, which must be simple, based as much as possible on common sense, fast, and not too expensive to implement – a model which truly fits the needs of VSEs.

Figure 3-9 ParqueSoft's five macro objectives and their corresponding strategies (Arenas 2007)

ParqueSoft has completed the implementation of its Quality Management System based on ISO 9001. This certification turns ParqueSoft into the first enterprise incubator in Colombia to certify its quality processes. This is also being achieved by 17 of its VSEs. The next goal is to certify all ParqueSoft VSEs within the next 4 years. The delegate from Columbia on WG24 works at the ParqueSoft Foundation as Quality Assurance Director, ParqueSoft Colombia.

3.6 Centers and initiative in Asia

3.6.1 The Association of Thai Software Industry

The Association of Thai Software Industry (ATSI) developed the Thai Quality Software (TQS) standard[36] to provide Thai VSEs with a way to improve their process quality using a standard as a reference model. TQS is a staged implementation of ISO 12207, where different processes are implemented at each of 5 capability levels, and each level has different requirements (L1 = records, L2 = procedures, plans, and L3, L4, L5 = more processes). TQS was developed to respond to the following issues:

- Thai SMEs are not ready to implement the entire ISO 12207 standard.
- Not all ISO 12207 activities are suitable for SME operations.
- There is no assessment model for the ISO/IEC 12207 standard.
- Most software developers are not document-oriented.

To address those issues, the ATSI proposed the following guidelines for the creation of a framework:

- Break down the ISO 12207 standard into stages or levels in order to fit all sizes of SME;
- Modify ISO 12207 activities to suit SME operations – product and project based on the type of business;
- Develop a set of checklists for use by assessors;
- Provide templates and examples.

The TQS standard has the following characteristics:

  - It has been adapted from the ISO/IEC 12207 Software Life Cycle Standard.
  - It is divided into 5 stages.
    - Each stage ensures that software organizations use international standards for producing software.
    - Software organizations are assessed for certification at each stage.
  - It comprises 3 main processes:
    - Primary Life Cycle Process,
    - Supporting Life Cycle Process

[36] http://www.atsi.or.th/atsi_th

- Organizational Life Cycle Process

Table 3-4 illustrates the breakdown, from level I to level V, of the development process.

12207	12207 Processes & Activities	Level I	Level II	Level III	Level IV	Level V
5.	*Primary life cycle processes*					
5.3	Development process	Process implementation	Process implementation	Process implementation	Process implementation	Process implementation
		Software requirements analysis	Systems requirements analysis	Systems requirements analysis	Systems requirements analysis	Systems requirements analysis
		Software architectural design	System architectural design	System architectural design	System architectural design	System architectural design
		Software coding and testing	Software requirements analysis	Software requirements analysis	Software requirements analysis	Software requirements analysis
		Software acceptance and support	Software architectural design	Software architectural design	Software architectural design	Software architectural design
			Software coding and testing	Software detailed design	Software detailed design	Software detailed design
			Software installation	Software coding and testing	Software coding and testing	Software coding and testing
			Software acceptance and support	Software integration	Software integration	Software integration

Table 3-4 Breakdown of the development process

By March 2005, 43 Thai software organizations had already been certified at TQS level 1, and 11 software organizations had been certified at TQS level 2.

However, in spite of the effort made to stage the standard and make it a step-by-step approach, most companies (VSEs) still found it too complicated and too difficult to implement, and few of them managed to do so.

3.6.2 Other initiatives in Asia

Malaysia

The model in Malaysia is to provide extra funding to cover consulting, appraisal, and training costs for companies that embark on a CMM-based PI effort; 40 companies had been supported by this project by 2006 (Garcia et al. 2006).

India

The Indian government provides tax incentives for companies that demonstrate improvement against a number of accepted models, rather than providing funding up front. This approach is more successful than providing extra funding for training, etc. (Garcia et al. 2006).

Hong Kong

This project was organized by the Hong Kong Computer Society and funded by the SME Development Fund of the Trade and Industry Department, HKSAR Government. The main objectives of this project are (Garcia et al. 2006):

- to provide an alternative means for Hong Kong Software SMEs to reach development capability assessed at Level 2 (Repeatable) or 3 (Defined) of the CMMI®;
- to provide avenues for increased collaboration between software SMEs in Hong Kong and partnering software companies located in the Pearl River Delta region;
- to foster relationships while encouraging cooperation, collaboration, and communication among IT professionals, the software industry in Hong Kong, and international experts.

The project developed a toolkit to expedite understanding and adoption of the CMMI® by a model group of small and medium-sized software organizations (Hareton et al. 2006). The government funded 15 companies (out of 600 who applied) and provided expertise and training to support them in pursuing model-based improvement. Five of these companies achieved CMMI® maturity level 2 during the course of the project (May 2003 to December 2004) (Garcia et al. 2006).

3.7 Conclusion

A few countries have opted for the 'survival of the fittest' strategy for their VSEs, i.e. an approach where a government does not intervene in the marketplace and lets the market decide which VSEs will survive. At the same time, a large number of government agencies, universities, research centers, and associations have worked to determine how to help VSEs. The initiatives presented share some of the following assumptions about the needs of VSEs:

- VSEs require low-cost solutions;
- VSEs require additional effort in communications and in standardizing vocabulary;
- VSEs require a staged approach;

- VSEs require ways to identify potential quick wins;
- VSEs require guidance in the selection and implementation of software practices.

The approach taken by the ISO working group, described in more detail in a subsequent chapter, corresponds to the mixed economy approach, where the intent is to help VSEs succeed in business by providing them with a set of software engineering practices tailored to their needs, in the form of international standards and technical reports.

In addition, knowing that most VSEs do not have the expertise or the resources to develop useful processes from published standards, the working group decided to develop guides, templates, checklists, and examples to facilitate and accelerate the transfer of technology from the developers of the standard to VSEs. At that point, it is expected that some VSEs will use the technology developed on their own, other VSEs will get some help from government organizations, such as training or coaching, and some large organizations will impose the new standards on the VSEs that supply components for their products.

There was also a decision made by the working group to provide a roadmap for VSEs to improve their capabilities up to a level corresponding to a point where they would be autonomous and capable of selecting the appropriate technologies to either sustain or grow their business.

The next chapter is divided into two parts. In the first part, the ISO's process for developing engineering standards is described. In the second part, the official accomplishments of WG24, i.e. according to the official mandate approved by the ISO, are presented, along with some of the unofficial activities performed, i.e. activities performed outside the official mandate of the working group.

This page is intentionally left blank

Chapter 4

The Development of International Standards

4 Chapter 4 – The Development Process for International Standards

4.1 Introduction

An organization wanting to develop a successful standard has two main objectives: develop a usable standard and convince the targeted users, e.g. government agencies, manufacturers, and developers, to purchase it, use it, or impose it. But a critical success factor is that practitioners actively participate in the development of standards, otherwise there is a risk that the standard may be either not usable or not used.

Although their use is generally voluntary, standards are, in fact, quasi-legal documents, and evidence associated with a standard imposed by a customer on its suppliers may be used to substantiate or refute points raised in a court of law. Moreover, standards often become legal requirements when they are adopted by governments and regulatory agencies. It is very important, therefore, that they are developed using a process based on a solid consensus and participation on the part of a broad range of interested parties.

The development of a standard normally takes 3 to 5 years, as explained in more detail below. But the publication of a standard does not imply widespread use, as entities, especially VSEs, may not be aware of the availability and benefits of standards. It is very important, therefore, to address the issue of technology transfer/diffusion to VSEs, otherwise the standards will collect dust at the International Organization for Standardization (ISO) in Geneva.

The ISO standardization process requires that standards be developed by a committee representing a group of countries participating in the ISO and representing broad cultural diversity. When a standard is developed by a professional society, such as the IEEE, the range of cultural interests is narrower, because the contributors often reflect their own perspectives (Ferris 2006). Nevertheless, even the development of an ISO standard involves cultural issues. This topic will be discussed in the second part of this chapter.

It is worth mentioning that, for ISO and IEEE organizations, the practitioners who participate in the development of standards are volunteers, i.e. they are not paid by for their time and travel. One of the reasons why VSEs are not involved in the development of standards is that they cannot afford to attend the meetings of the working groups that formulate them.

This chapter is divided into three parts: in the first part, on the development of international standards, the development process for ISO standards is described. The author also briefly touches on the involvement of the IEEE Computer Society, since it has developed close relationships with the ISO on the development and harmonization of software and systems engineering standards, and the two organizations are currently collaborating on the development of a common portfolio of standards.

In the second part, on technology transfer, the author presents the process of technology transfer and some cultural issues that have to be considered when an international standard is being developed. As mentioned above, the publication of a standard does not necessarily mean that it will be widely used, especially in the case of VSEs, who may not be aware of the availability and benefits of standards. This part will help explain the activities that the working group described in part 3 undertook, over and above the official ISO activities, to ensure that the new standards would respond to the needs of VSEs and would be supported once they are published by the ISO.

In the third part, on the history leading to the establishment of an ISO/IEC JTC 1/SC7 working group for VSEs and recent achievements, a chronology of the events leading to the creation of ISO/IEC SC7 Working Group 24, as well as its accomplishments, are presented.

Although part 2 introduces the topic of technology transfer, this chapter covers mainly phase 3 of the 6-phase innovation process illustrated in Figure 4-1, titled Development.

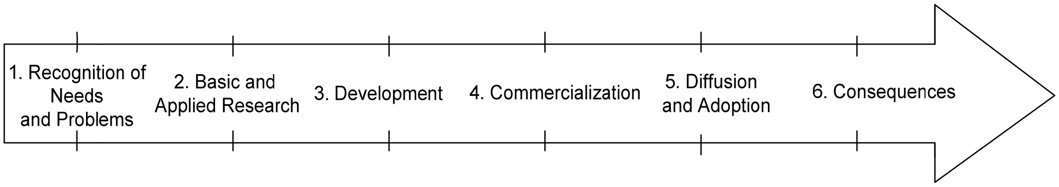

1. Recognition of Needs and Problems
2. Basic and Applied Research
3. Development
4. Commercialization
5. Diffusion and Adoption
6. Consequences

Figure 4-1 Phases of the Innovation Process (adapted from Rogers 2003)

Phase 5, Diffusion and Adoption, is discussed in more detail later, mainly in chapter 7 and to some extent in chapter 8.

4.2 Part 1 – The development of international standards

In this section, the author briefly describes the ISO subcommittee responsible for the development of software standards and its activities.

4.2.1 Description of the International Organization for Standardization (ISO)

The ISO was founded in 1947. Its mission is to promote the development of standardization and related activities around the world with a view to facilitating the international exchange of goods and services, and to developing cooperation in the spheres of intellectual, scientific, technological, and economic activity. The organization has a membership of 161 national standards institutes from countries in all regions of the world, and a portfolio of more than 17,800 standards[37]. The International Electromechanical Commission (IEC) was founded in 1906. IEC is the organization that prepares and publishes international standards for electrical, electronic, and related technologies. During 1987, the ISO and the IEC joined forces and put in place a joint technical committee, called Joint Technical Committee 1 (ISO/IEC JTC1), with the following mandate (Coallier 2003):

> *Standardization in the Field of Information Technology: Information technology includes the specification, design, and development of systems and tools dealing with the capture, representation, processing, security, transfer, interchange, presentation, management, organization, storage, and retrieval of information.*

Figure 4-2 illustrates the structure of the standardization organizations, and the position of JTC1 within this structure.

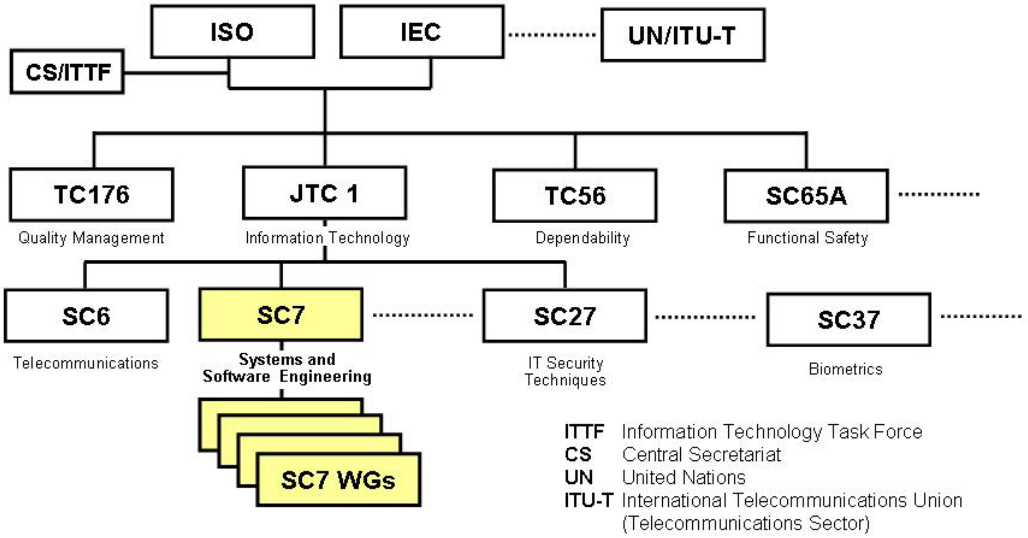
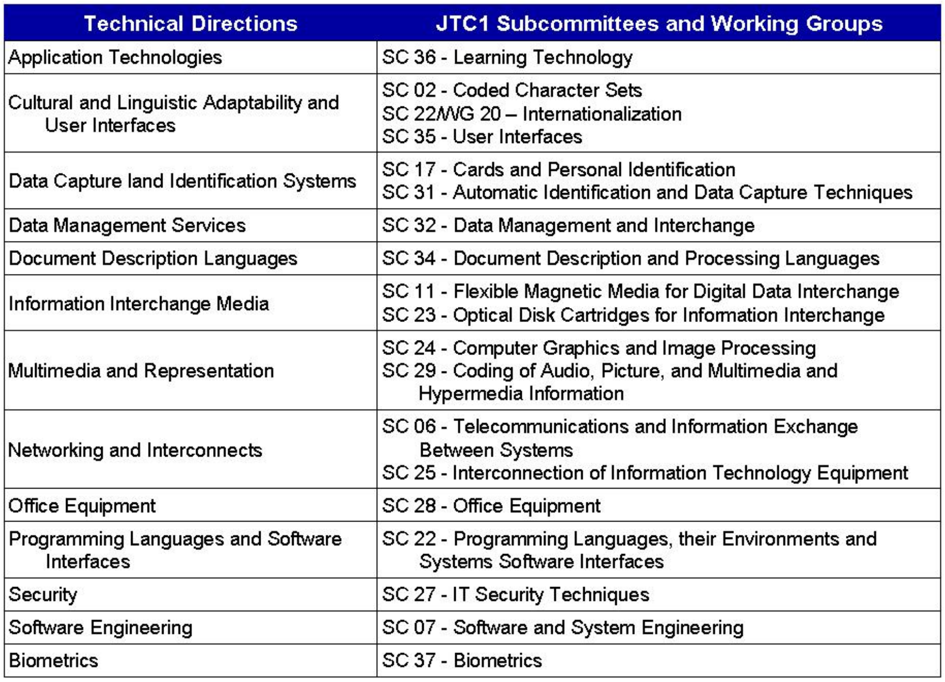


Figure 4-2 Structure of Standards Organizations (ISO 2009h)

One of the subcommittees (SC) of JTC1 is subcommittee 7 (SC7[38]), as illustrated in Figure 4-2 and Table 4-1.

[37] www.iso.ch

[38] ISO/IEC/JTC1 SC7

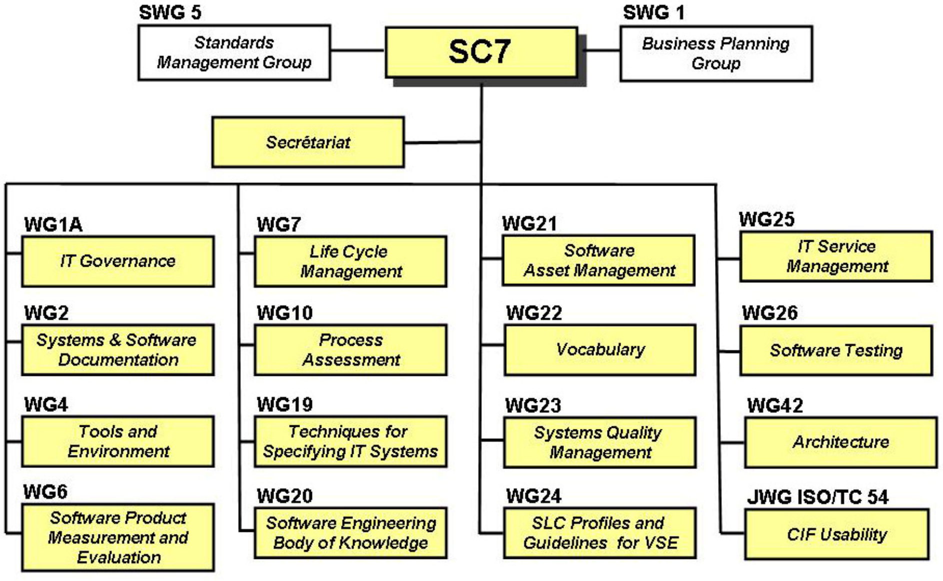


Technical Directions	JTC1 Subcommittees and Working Groups
Application Technologies	SC 36 - Learning Technology
Cultural and Linguistic Adaptability and User Interfaces	SC 02 - Coded Character Sets SC 22/WG 20 – Internationalization SC 35 - User Interfaces
Data Capture land Identification Systems	SC 17 - Cards and Personal Identification SC 31 - Automatic Identification and Data Capture Techniques
Data Management Services	SC 32 - Data Management and Interchange
Document Description Languages	SC 34 - Document Description and Processing Languages
Information Interchange Media	SC 11 - Flexible Magnetic Media for Digital Data Interchange SC 23 - Optical Disk Cartridges for Information Interchange
Multimedia and Representation	SC 24 - Computer Graphics and Image Processing SC 29 - Coding of Audio, Picture, and Multimedia and Hypermedia Information
Networking and Interconnects	SC 06 - Telecommunications and Information Exchange Between Systems SC 25 - Interconnection of Information Technology Equipment
Office Equipment	SC 28 - Office Equipment
Programming Languages and Software Interfaces	SC 22 - Programming Languages, their Environments and Systems Software Interfaces
Security	SC 27 - IT Security Techniques
Software Engineering	SC 07 - Software and System Engineering
Biometrics	SC 37 - Biometrics

Table 4-1 JTC1 Subcommittees (adapted from Coallier 2003)

The Terms of Reference of SC7 comprise: "Standardization of processes, supporting tools, and supporting technologies for the engineering of software products and systems." A revised Terms of Reference has been proposed to JTC1 as follows: "Standardization of processes, methods, and supporting technologies for the engineering and management of software and systems throughout their life cycles."

Figure 4-3 illustrates the structure of SC7. Once the mandate of a working group has been fulfilled, the working group is disbanded. This explains why there is a working group 42, but only 16 are displayed in the figure. One of the working groups of this subcommittee is the one responsible for the development of standards for VSEs: Working Group 24 (WG24).


SWG 5
Standards Management Group
SC7
SWG 1
Business Planning Group
Secrétariat
WG1A
IT Governance
WG2
Systems & Software Documentation
WG4
Tools and Environment
WG6
Software Product Measurement and Evaluation
WG7
Life Cycle Management
WG10
Process Assessment
WG19
Techniques for Specifying IT Systems
WG20
Software Engineering Body of Knowledge
WG21
Software Asset Management
WG22
Vocabulary
WG23
Systems Quality Management
WG24
SLC Profiles and Guidelines for VSE
WG25
IT Service Management
WG26
Software Testing
WG42
Architecture
JWG ISO/TC 54
CIF Usability

‡ Adapted from Prof. M. Azuma

Figure 4-3 Structure of subcommittee SC 7 (ISO 2009h)

The ISO has three categories of members: those in the first category, called *P Members*, participate actively in the work. A P Member has an obligation to vote (approve, disapprove, or abstain) on all questions formally submitted for voting within the technical committee or subcommittee, on new work item proposals, enquiry drafts, and final drafts for the international standards, and to contribute to meetings (ISO 2007c). Currently, there are 36 P Members participating in SC7 (see Table 4-2).

Australia (SA)	Korea, Republic of (KATS)
Belgium (NBN)	Luxembourg (SEE)
Brazil (ABNT)	Netherlands (NEN)
Canada (SCC)	New Zealand (SNZ)
China (SAC)	Peru (INDECOPI)
Colombia (ICONTEC)	Poland (PKN)
Czech Republic (CNI)	Portugal (IPQ)
Denmark (DS)	Romania (ASRO)
Finland (SFS)	Slovakia (SUTN)
France (AFNOR)	South Africa (SABS)
Germany (DIN)	Spain (AENOR)
India (BIS)	Sweden (SIS)
Iran, Islamic Republic of (ISIRI)	Switzerland (SNV)
Ireland (NSAI)	Thailand (TISI)
Israel (SII)	USA (ANSI)
Italy (UNI)	Ukraine (DSSU)
Japan (JISC)	United Kingdom (BSI)
Kazakhstan (KAZMEMST)	Venezuela (FONDONORMA)

Table 4-2 List of Countries Participating in SC7[39] (ISO 2009h)

The second category is made up of observer members, called *O Members*. They follow the work as observers and receive documents. An O Member has the right to submit

[39] The information in parentheses identifies national body, e.g. Canada (SCC), where SCC stands for Standards Council of Canada.

comments and attend meetings (ISO 2007a). Currently, 19 countries are O Members of SC7 (see Table 4-3).

Argentina (IRAM)
Austria (ON)
Cuba (NC)
Dominican Republic (DIGENOR)
Estonia (EVS)
Ethiopia (QSAE)
Hong Kong, China (ITCHKSAR)
Hungary (MSZT)
Iceland (IST)
Indonesia (BSN)
Kenya (KEBS)
Malasia (DSM)
México (DGN)
Norway (SN)
Philippines (BPS)
Russian Federation (GOST R)
Serbia (ISS)
Turkey (TSE)
Uruguay (UNIT)

Table 4-3 List of Observer Countries in SC7 (ISO 2009h)

The third category is made up of liaison members, called *L Members*. Those are organizations that can contribute to the work of JTC1, such as the IEEE Computer Society and the International Council on System Engineering (INCOSE). An L member has no power of vote, but has some options in terms of attending meetings and receiving documents. There 3 types of liaison members (ISO 2007a):

- Category A. Organizations that make effective contributions to, and participate actively in, the work of JTC 1 or its SCs for most of the questions dealt with by the committee.
- Category B. Organizations that have indicated a wish to be kept informed of the work of JTC 1 or any of its SCs.
- Category C. Organizations that make effective technical contributions and participate actively at the WG or project level.

Once a year, all working groups move to a common location for a one-week meeting in or around the month of May for the purpose of collaboration and coordination among working groups. Figure 4-4 illustrates the attendance at the annual plenary meetings in various countries. In 2003, attendance was at a ten-year low, as delegates were not allowed to travel to Canada because of Severe Acute Respiratory Syndrome (SARS). Also, an annual interim meeting of the working groups is scheduled. The working groups meet in or around the month of October at the location of their choice for a one-week meeting. Finally, working groups and editor teams may also conduct additional meetings. The frequency of meetings is the reason why some people humorously call the ISO the "International Sightseeing Organization"!

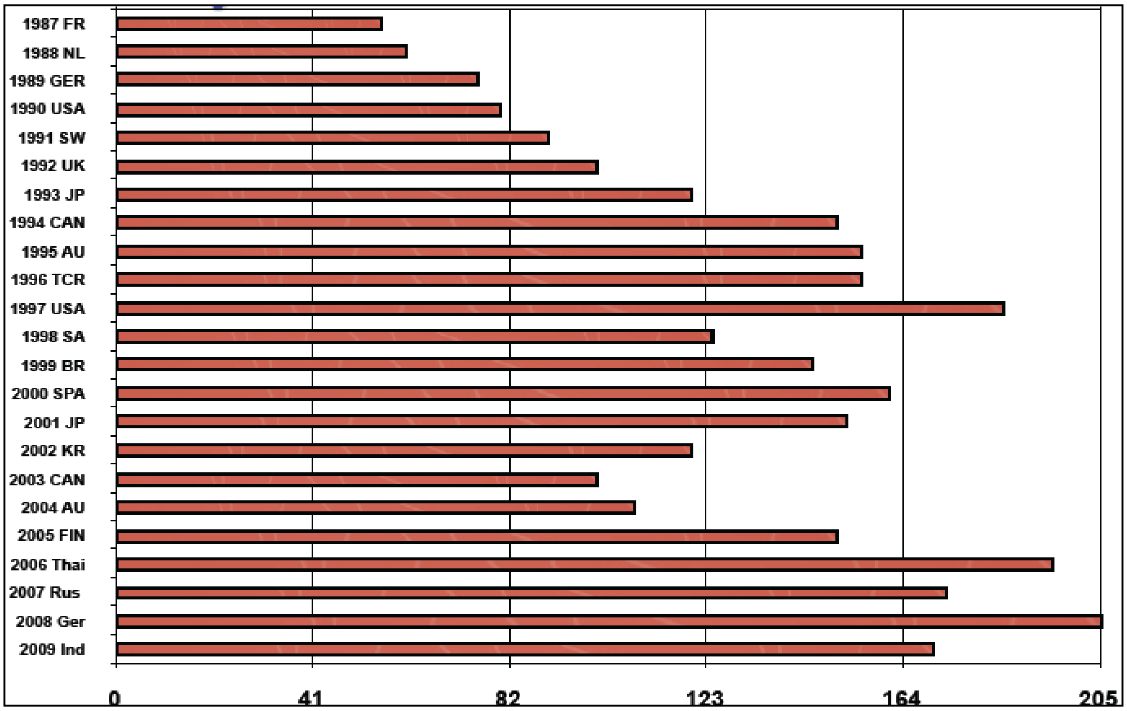


Figure 4-4 Participation in Plenary Meetings (ISO 2009h)

Integration and harmonization of the collection of engineering standards

For many years, the IEEE Computer Society and ISO SC7 had been developing their own collection of software engineering standards. At one point, they began to share standards to fill the gaps in their own collections. In 2004, they adopted a procedure for the development and harmonization of standards (SC7 2004). The principles of this relationship are as follows:

- The collections of SC7 and SESC should be consistent and complementary – harmonized. Users should able to select and apply standards from both collections without contradiction.
- The SC7 collection, taken as a whole, should be at a higher level of abstraction than the SESC collection. Typically, SC7 standards would describe principles, while SESC standards would provide more detailed treatments of selected subjects.
- Both organizations should respect the consensus achieved by the other organization and avoid creating multiple variants of the documents.
- Whenever possible, coordination of a standard should commence by one organization adopting a standard of the other organization, so that coordination begins with a shared baseline.

- Maintenance/revision of adopted documents should be accomplished through a coordinated process so both organizations have the same standard.

Figure 4-5 illustrates the evolution of the ISO/IEC JTC1 standards that are maintained and published under the responsibility of SC7. The growth, from about 5 standards in the 1980s to over 100 standards in 2009, is explained mainly by the fact that more software engineering practices have matured and gained a broad consensus since the end of the 1980s.

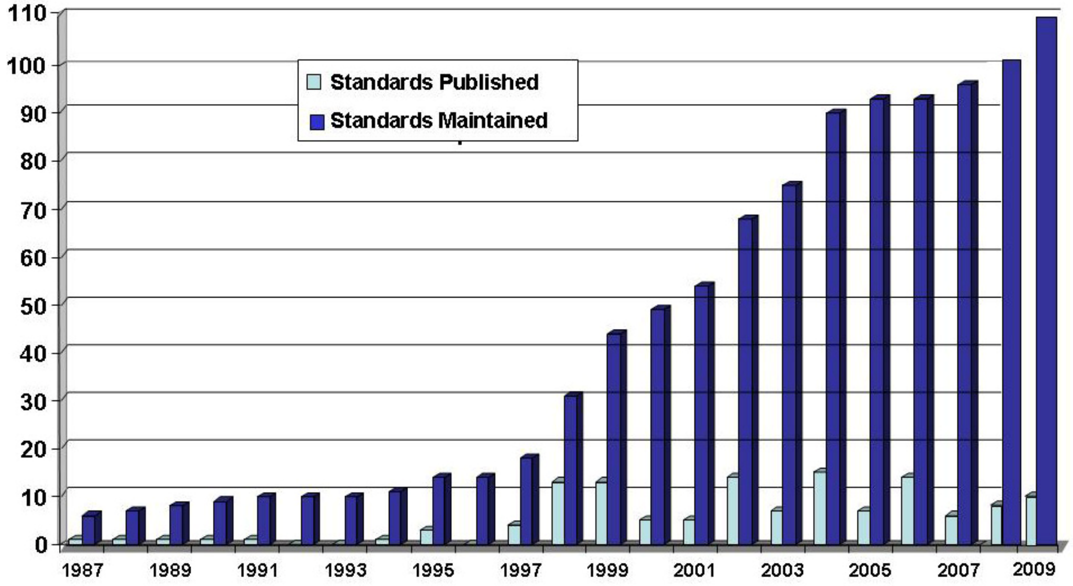


Figure 4-5 Evolution of published ISO/IEC JTC1 SC7 Software and Systems Engineering Standards (ISO 2009h)

The portfolio of SC7 standards, listed in Table 4-4, is divided into areas of work, such as body of knowledge or process standards.

Areas of Work	Identification of the Standard or Technical Report
Software Engineering Body of Knowledge: establish the appropriate sets of criteria and norms for the professional practice of software engineering upon which industrial decisions, professional certification, and educational curricula can be based	• ISO/IEC TR 19759:2005, Guide to the Software Engineering Body of Knowledge • ISO/IEC 24773:2008, Software engineering — Certification of software engineering professionals • ISO/IEC 24765:2009 Systems and Software Engineering Vocabulary
Software and Systems Engineering Processes Standards: describe software and systems engineering practices and assess organizational software and systems engineering practices	• ISO/IEC 12207 Software Life Cycle Processes; • ISO/IEC 15288 Systems Life Cycle Processes; • ISO/IEC 15504 Software Process Assessment series. • ISO/IEC 90003 – Guidelines for the Application of ISO 9001 to computer software • ISO/IEC TR 90005:2008 – Systems engineering -- Guidelines for the application of ISO 9001 to system life cycle processes

Areas of Work	**Identification of the Standard or Technical Report**
	• ISO/IEC TR 15271 – Guide to ISO/IEC 12207 • ISO/IEC 14764 – Software Maintenance • ISO/IEC TR 15846 – Configuration Management • ISO/IEC 15910 – Software User Documentation Process • ISO/IEC 15939 – Software Measurement Process • ISO/IEC TR 16326 – Guide for the Application of ISO/IEC 12207 to Project Management • ISO/IEC TR19760 – Guide for ISO/IEC 15288 • ISO/IEC 29110 – Lifecycle Profiles for Very Small Entities series • ISO/IEC 16085 – Risk Management
Software Systems Products Standards: allow developers, purchasers, and buyers to size and document software products, as well as to express, measure, and evaluate the quality of the software that is produced and its contribution to the final product or application systems	• ISO/IEC 14598 – Information technology – Software product evaluation series • ISO/IEC 14143 – Information technology – Software measurement -- Functional size measurement series • ISO/IEC 19761:2003 – Software engineering – COSMIC-FFP -- A functional size measurement method • ISO/IEC 15026:1998 – Information technology – System and software integrity levels • ISO/IEC 25000:2005 – Software Engineering – Software product Quality Requirements and Evaluation (SQuaRE) series • ISO/IEC 20926:2003 – Software engineering – IFPUG 4.1 Unadjusted functional size measurement method • ISO/IEC 20968:2002 – Software engineering – Mk II Function Point Analysis -- Counting Practices Manual • ISO/IEC 24570:2005 – Software engineering – NESMA functional size measurement method version 2.1 – Definitions and counting guidelines for the application of Function Point Analysis • ISO/IEC 15289 – Systems and software engineering – Content of systems and software life cycle process information products (Documentation)
Enterprise Architecture Standards: integrate IT and business systems definitions and provide the software and systems engineering tools to implement enterprise information systems	• ISO/IEC 13235 – Information technology – Open Distributed Processing -- Trading function, series • ISO/IEC 10746 – Information technology – Open Distributed Processing. Series • ISO/IEC 14752:2000 – Information technology – Open Distributed Processing – Protocol support for computational interactions • ISO/IEC 14753:1999 – Information technology – Open Distributed Processing – Interface references and binding • ISO/IEC 14750:1999 – Information technology – Open Distributed Processing – Interface Definition Language

Areas of Work	Identification of the Standard or Technical Report
Software Engineering Environment Standards: make it easier to use software-engineering environments and to reuse and redeploy the data contained in them	• ISO/IEC 14102:2008 – Information technology -- Guideline for the evaluation and selection of CASE tools • ISO/IEC TR 14471:2007 – Information technology – Software engineering – Guidelines for the adoption of CASE tools
Software and Systems Engineering Formalisms Standard: formal representations and modeling of software and systems	• ISO/IEC DIS 19500 – Information technology -- Open distributed processing – Common Object Request Broker Architecture (CORBA) specification
Management of Software Assets Standards: describe the basic requirements of a software asset management environment	• ISO/IEC 19770 – Information technology -- Software asset management, series
Legacy Standards: legacy information processing standards	• ISO 3535:1977 – Forms design sheet and layout chart ISO/IEC TR 14759:1999 – Software engineering – Mock-up and prototype -- A categorization of software mock-up and prototype models and their use
Governance Standards: describe the requirements for governance	• ISO/IEC 20000 – Information technology -- Service management, series • ISO/IEC 38500:2008 – Corporate governance of information technology
Testing Standards: describe the requirements for testing	• ISO/IEC FCD 26513 – Systems and software engineering – Requirements for testers and assessors of user documentation
Miscellaneous	• ISO/IEC 42010:2007 – Systems and software engineering -- Recommended practice for architectural description of software-intensive systems • ISO/IEC CD 29118 – Information Technology – Tools and methods of requirements engineering and management for product lines • ISO/IEC 26514:2008 – Systems and software engineering – Requirements for designers and developers of user documentation • ISO/IEC 6592:2000 – Information technology -- Guidelines for the documentation of computer-based application systems • ISO/IEC TR 9294:2005 – Information technology -- Guidelines for the management of software documentation

Table 4-4 Portfolio of SC7 Standards (adapted from Coallier 2003)

Figure 4-6 illustrates the structure and relationships between ISO standards:

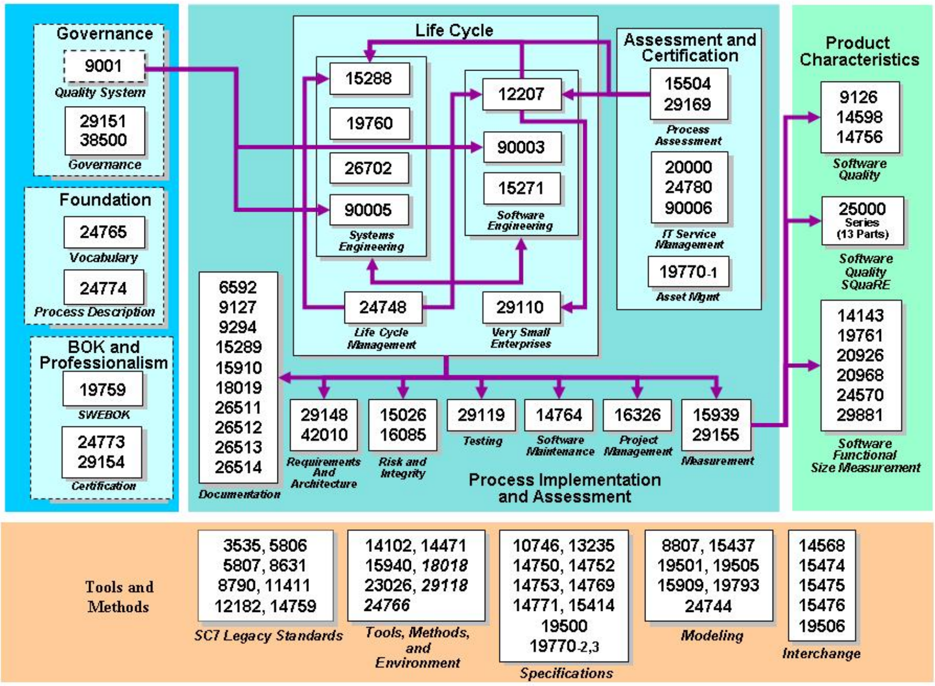


Figure 4-6 Portfolio of SC7 Standards (ISO 2009h)

The relationships between key ISO and SC7 standards, i.e. 12207, 15504, and 9001, and the new standards for VSEs, are illustrated in Figure 4-7. Although these standards are well known in large software and systems engineering organizations, the current SC7 Life Cycle standards are a challenge to use in VSEs. Compliance with them is difficult to achieve, if not impossible, for VSEs. Consequently, VSEs have few, or a very limited number of ways to be recognized as organizations producing quality software systems. In a later chapter, the standards that meet the needs of VSEs are described.

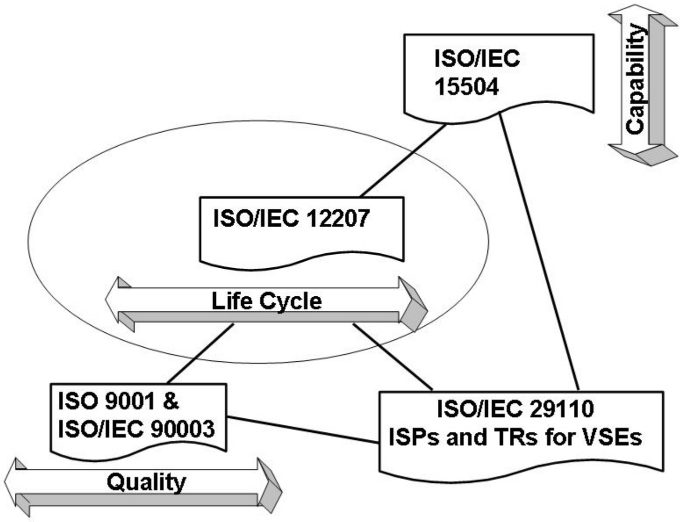


Figure 4-7 Relationships between key SC7 standards (adapted from Coallier 2003)

Once a standard is published, the official duties of a working group are completed and the working group is usually disbanded. In the author's opinion, it is an illusion to believe that once a standard for VSEs is published, VSEs will spontaneously be aware of that fact and apply the standard. In the next part, the concepts of technology transfer/diffusion are presented. In a later chapter, the author presents a detailed account of what is being done to encourage VSEs all over the world to apply the standards in their daily software management and implementation activities.

4.2.2 *Roles and responsibilities of working groups and their members*

Here, the roles and responsibilities of working groups, including convenors, editors, and members, and delegates are described (adapted from ISO 2007a and CAN 2006).

Administrative responsibilities of working groups (WGs)

The administrative responsibilities of working groups include:

- Maintenance of a document distribution list;
- Maintenance of lists of members and liaisons, the convenors or secretariats of WGs ensuring that the JTC 1 Secretariat receives the necessary information;
- Maintenance of a document register;
- Preparations for WG meetings in consultation with hosts;
- Timely distribution of documents;
- Preparation and distribution of meeting agendas in accordance with guidelines;

- Preparation of meeting reports, which shall include the following:
    - List of attendees, including their nominating organization (a national body (NB) or liaison organization) and employer;
    - Actions taken relative to assigned projects;
    - Problems and issues highlighted;
    - Target date updates;
    - Resolutions;
    - Forwarding of the meeting report and resolutions to the parent body secretariat for distribution to the parent body for action as appropriate;
    - Maintenance of progress reports (includes updates to JTC 1 database).

Roles and responsibilities of the convenor

A WG convenor is nominated for a three-year term ending at the next plenary session of the parent body following that term. A convener may be reappointed for additional three-year terms.

The convenor is responsible for monitoring, reporting, and ensuring active progress of the work, and bringing this work to an early and satisfactory conclusion. These tasks are carried out as far as possible by correspondence. This individual is responsible for ensuring that ISO/IEC Directives and the decisions of the ISO Technical Committee (TC) or Sub-Committee (SC) are followed, as well as acting in an expert capacity and without expressing a national point of view.

The individual appointed as convenor shall (adapted from CAN 2006):

- organize the work of the group;
- ensure that all necessary aspects of the subject are dealt with and that duplication or contradiction of the work of other technical committees or sub-committees is avoided;
- maintain a list of names and addresses of all working group members, which should be copied to the secretariat of the parent technical committee or sub-committee;
- ensure that the documents of the working group are properly identified, archived, and made available to working group members;

- number, prepare for circulation, and circulate working documents to members, and edit the final version;
- prepare drafts for distribution to the parent committee;
- send copies of all working group correspondence to:
  - individual appointed members
  - the secretariat of the parent committee
  - the ISO Central Secretariat or JTC1 Secretariat;
  - accredited liaison member bodies;
- set target dates for completion of the work assigned to the group, in consultation with the parent committee and plan accordingly;
- arrange, schedule, and convene meetings if questions cannot be handled by correspondence;
- prepare reports to the parent committee on the progress of the work.

Roles and responsibilities of project editors

The project editor is identified as early as possible for each standard or other document under development. This individual is appointed by the SC and follows the editing instructions given by the ISO.

The project editor is responsible for maintaining the document throughout the stages of technical work, i.e. until publication, ensuring that the foreword of the final text of the standard indicates the JTC 1 SC responsible for the standard. After publication, this individual maintains an updated document incorporating all approved technical corrigenda and amendments so that a revision may be published with minimum delay when appropriate.

Roles and responsibilities of delegates from Canada

A Canadian delegate on an ISO committee does not represent a professional or technical society or trade association, but rather the Standards Council of Canada (SCC), which is the Canadian member of the ISO. Delegates to the ISO technical committees and subcommittees have the task of ensuring that the views and positions of the particular national member body that they represent are known and understood by the committee. Delegates participate in negotiations and consultations intended to lead to the development of an international consensus which considers the view of the delegate's country position in the outcome.

Active participation on the part of Canadian delegates includes (CAN 2004):

- voting on ballots within specified time frames;
- making comments on work under development within specified time frames;
- participating in CAC meetings (either in person or by other means);
- use of the SCC's electronic forum work area.

When representing Canada at an international meeting, delegates state the Canadian position and comment on behalf of the Canadian member body, i.e. the SCC, to the ISO, and in the third person. A Canadian delegate does not represent himself or his employer during the technical sessions or associated social events. Canadian delegates do not express personal opinions contrary to the official Canadian position established at the briefing meeting (CAN 2004).

Since WG documents are circulated only to the members of the group, the WG member's report is the primary source of information about the activity. After each WG meeting, the Canadian expert submits a progress report to the CAC Chair and recommends continuance in this activity if warranted.

The meeting report should contain minimum technical detail, except where items of particular concern to Canada are discussed. The WG meeting report should include the following:

- Location and date of meeting;
- List of experts who attended;
- Main items of business;
- Progress made in completing the task and anticipated work schedule;
- Indication of any item of particular concern to Canada;
- Evaluation of the importance of the work to Canada;
- Recommendation concerning continued Canadian participation.

Voting principles for Canadian delegates

Delegates from Canada follow a set of principles when voting (CAN 2004):

- In determining an affirmative voting position, the following principles should be considered:
    - that the proposal is technically sound and acceptable for use in Canada;
    - that the proposal is technically sound, although it is currently not relevant for use in Canada;
    - that the proposal is in accordance with Canadian practice (as laid down in a Canadian standard or other standard, or embodied in basic design criteria used in Canada);
    - that the proposal is not in conflict with Canadian cultural and/or jurisdictional imperatives.
- In determining a negative voting position, the following principles should be considered:
    - that the proposal is not technically sound;
    - that the proposal is not acceptable for use in Canada;
    - that the proposal will be in conflict with published Canadian standards, related standards, or basic design criteria used in Canada;
    - that the proposal is in conflict with Canadian cultural and/or jurisdictional imperatives.
- In determining an abstention from voting, the following principle should be considered:
    - that the CAC has not participated in developing the document and considers the subject to have no current or expected future relevance for Canada.

Head of the Canadian delegation

When the Canadian delegation consists of more than one individual, the responsible Canadian Advisory Committee (CAC) Chair appoints a head of delegation, who acts as spokesperson and may call on other delegates to speak on specific agenda items. The author has been appointed Head of Delegation, for Canada for JTC1 SC7 Working Group 24 (Software Life Cycle Profiles and Guidelines for use in Very Small Entities (VSE)) and Working Group 20 (Software and Systems Bodies of Knowledge and Professionalization).

4.2.3 International standard and technical report development process

In this section, the process for developing new international standards (IS) or technical reports (TR) and the roles and responsibilities of participants in a working group are explained (ISO 2007a). The author has been involved in every step of the process of development of the ISO standards and ISO TR for VSEs.

The basic principle in developing a standard in order to obtain broad user commitment and adoption is by consensus. Consensus is defined by the ISO as follows (ISO 2007a):

> *General agreement, characterized by the absence of sustained opposition to substantial issues by any important part of the concerned interests and by a process that involves seeking to take into account the views of all parties concerned and to reconcile any conflicting arguments.*

Consensus means the following (Coallier 2003):

- All the parties involved were able to voice their views.
- The best effort was made to take into account all the above views and resolve all issues (i.e. all comments submitted during a ballot).

However, consensus does not mean unanimity, the minimum support required for IS adoption at the ISO being a two-thirds majority of the participating member countries voting. An ISO TR is adopted with a simple majority only, since the impact of a TR is much less than that of an IS.

IS, TR, and ISP (International Standardized Profiles) are developed in a six-stage process that normally takes between three and five years to complete, from initiation to publication by the ISO. The six stages of development are defined as illustrated in Table 4-5.

Stage	**Standard**	**Technical Report**	**ISP**
Stage 0 (preliminary stage)	A study period is under way.	idem	idem
Stage 1 (proposal stage)	A New Work Item Proposal (NP) is under consideration.	idem	idem
Stage 2 (preparatory stage)	A Working Draft (WD) is under consideration.	idem	idem
Stage 3 (committee stage)	A Committee Draft/Final Committee Draft (CD/FCD) is under consideration.	Proposed Draft Technical Report (PDTR)	• Proposed Draft International Standardized Profile (PDISP) • Final Proposed Draft International Standardized

Stage	Standard	Technical Report	ISP
			Profile (FPDISP)
Stage 4 (approval stage)	A Final Draft International Standard (FDIS) is under consideration.	Draft Technical Report (DTR)	Final Draft International Standardized Profile (FDISP)
Stage 5 (publication stage)	An International Standard (IS) is being prepared for publication.	TR	ISP

Table 4-5 Stages of Development of ISO Standards (ISO 2007a)

- Proposal Stage (Stage 1) – A New Work Item Proposal (NP) may be submitted by any number of sources. A decision to add the item to the work program can be taken either by correspondence or at a meeting. Approval by a simple majority of participating members (i.e. those having a vote and defined duties) is required, plus a commitment by at least five NBs to participate actively in the work.

- Preparatory Stage (Stage 2) – Working Draft(s) (WD) – Initial documents circulated by the secretariat of the technical committee or subcommittee, or by the convenor of a working group, for consideration by its members.

- Committee Stage (Stage 3) – Committee Draft(s) (CD) – Working draft for an IS registered at the ISO Central Secretariat or IEC Central Office and submitted to members of a technical committee or subcommittee for consideration.

- Approval Stage (Stage 4) – Draft International Standard(s) (DIS) – A CD that has received consensus approval from the participating members of a technical committee or subcommittee for registration as a DIS. A two-thirds majority is deemed sufficient, with every effort being made to resolve negative votes. Consensus is judged by the chair in consultation with the secretariat of the committee and, if necessary, the project leader.

- Publication Stage (Stage 5) – International Standard(s) (IS) are approved for publication if:
    - a two-thirds majority of the votes cast by the participating members of the technical committee or subcommittee are in favor, and
    - not more than one-quarter of the total number of ISO or IEC member body votes cast are negative.

Every IS is reviewed every five years to determine if it should be confirmed, revised, or withdrawn. Figure 4-8 illustrates the life cycle of ISO documents, as well as the organizations that have control of a document at each step of the life cycle.

It is also possible to develop a standard using a faster process, called *Fast track*, which takes about two years. Our working group did not use this process because none of the documents we considered met the criteria for that process.

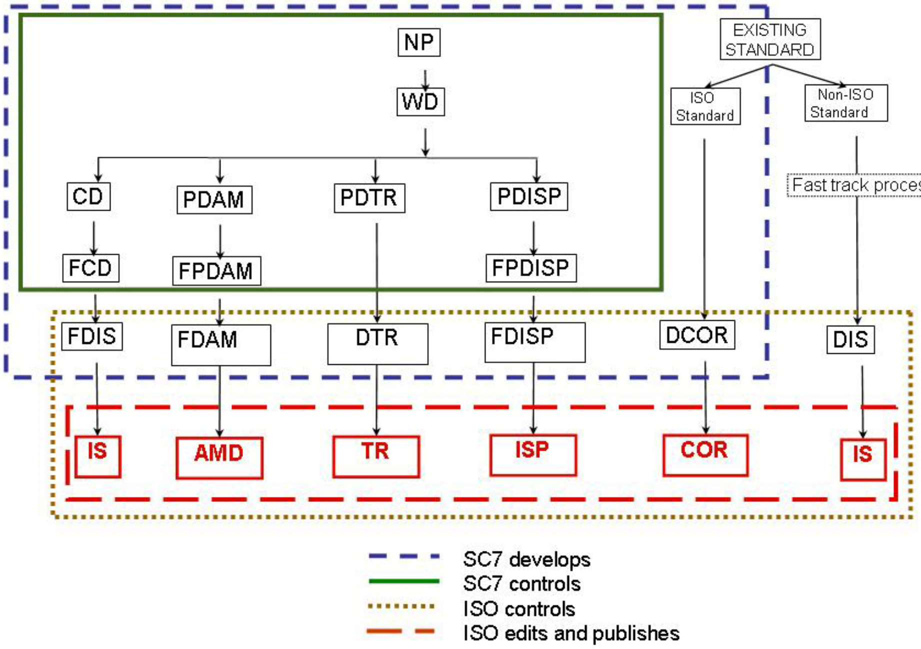


Figure 4-8 Document life cycle (adapted from ISO 2009h)

ISP and TR, the documents developed by WG 24, should be ready to enter stage 4 after the round of comments is submitted at the Fall 2009 meeting in Peru and be ready for publication by the ISO in 2010.

4.2.4 The processing of comments by SC7

When the national bodies receive a document for approval, they are asked to comment on it using the five categories listed in Table 4-6. The *Technical High* (TH) category is the most important of these, while *Editorial* (E) comments are the least important. A TH comment covers a technical problem which exists in many places in the document, which gives this category the greatest impact. It is preferable to use specific individual references for each case instead of this category if at all possible.

Technical High (TH) Major problems. For example, definitions, architectural (stage, process level), activity/task deletion or addition. Proposed new text is mandatory for additions and strongly recommended for modifications.

Technical Low (TL)
Minor problems. For example, earlier clauses (1, 2, 3), activity/task modification or relocation. Readability (including localization) problems should be TL. These problems are often of a more significant nature than a simple editorial change.
Proposed new text is mandatory for additions and strongly recommended for modifications.

General Editorial (GE) or General Technical (GT)
Avoid use of this category if at all possible.
If the comment is GE, it will greatly assist the project editor if each occurrence is identified. If it is GT, then it will be important for every contributing item to be addressed (by individual comments).

Editorial (E)
Covers grammar, missing or duplicated text, missing or incorrect cross-references, and spelling and punctuation problems.
These are matters that should be able to be resolved offline by the project editor.
Terminology changes are not E, e.g. enterprise versus organization.
Proposed new text should be provided to facilitate document updating.

Table 4-6 Categories of Comments (ISO 2009i)

When a document is sent to an NB, an Excel spreadsheet is included for the NB to register their comments. Table 4-7 illustrates the heading of the spreadsheet, which is completed as follows:

- The *NB* column records the code name of the country making the comment. For example, the code for Canada is CAN.
- The *No.* column indicates the number of the comment, starting with 1.
- The *Category c*olumn records the category (the options are listed in Table 4-6).
- The *Clause*, *Sub Clause*, *Paragraph*, *Figure*, and *Table* columns identify the specific area in the document targeted by the comment.
- The *Comment and rationale* column describes the issue.
- The *Proposed New Text* column provides the text to the delegates.

NB (ISO 3166)	No.	Category	Clause, Sub-clause	Paragraph, Figure, Table	Comment and rationale	Proposed new text

Table 4-7 Spreadsheet Headings

When a comment is analyzed by the delegates, it is given a response from the list of six categories in Table 4-8.

Agreed (A)
The comment correctly identifies a problem, and any proposed text or action can be implemented exactly as provided to resolve that problem.
Text may be provided either to ensure that there is no confusion in understanding the decision, or to separate alternatives.

Agreed in Principal (AIP)

The comment correctly identifies a problem, but any proposed text or other solution does not completely or correctly resolve that problem.
The accompanying text must state the solution as well as the reason for using the modified solution. In general, it is unacceptable to leave this step up to the editor, especially where large numbers of comments are concerned.

Rejected (R)
The comment does not identify a problem, propose changes that have previously been disallowed, or recommend resolution which was not provided. The disposition text must clearly state why the comment was rejected.

Withdrawn (W)
The NB voluntarily withdrew the comment from consideration. Accompanying text should be included to ensure that interested parties can understand the reason for the withdrawal.

Overtaken by events (OBE)
The comment relates to an item that was the subject of another comment. The other comment was A or AIP, and took an opposing or conflicting view, or resulted in deletion of the text that was the subject of this comment.
The accompanying text must clearly describe how and why the comment was superseded. Refer specifically to the other comment.
Note: If one comment is similar to another, it should be assigned to the same category as the other comment, i.e. not categorized as OBE.

Deferred (D)
Deferred for consideration until the next revision cycle, possibly after publication.

Table 4-8 Categories of Comments (ISO 2009i)

To illustrate the disposition of a comment, Table 4-9 shows the relevant columns for a comment sent by an NB: this is a TL comment submitted against clause 5.4. The reviewer indicates that there are too many roles, i.e. 12, for VSEs. This comment was analyzed by the working group and accepted as illustrated in the *Disposition* column. In the *Notes* column, the document editor noted a suggestion to reduce the number of roles for VSEs.

NB (ISO 3166)	No.	Category	Clause, Sub-clause	Paragraph, Figure, Table	Comment and rationale	Proposed new text	Notes	Disposition
XXX	18	TL	5.4		12 roles are too much to implement actual project for VSEs.	Reorganize and redefine them.	Multiple roles could be executed by the same individual. Eliminate RE and incorporate their activities to other roles	Accepted

Table 4-9 Example of the Disposition of a Comment

4.3 Part 2 – technology transfer

4.3.1 Introduction

Since the term *technology transfer* tends to mean different things to different people, a few definitions are provided (Pfleeger et al. 2003):

- an idea, practice, or object resulting from research, as well as an embodiment of the technology;

- the movement of technology into a setting where it can improve a product or process in some way;
- an entire process involving facilitators at various stages, including those who create the technology, those who incorporate it into a useful product, service, tool, or practice, and those who further develop the technology for commercialization and use.

Many organizations have invested in technologies, such as methods, tools, or paradigms, which were useful to some extent and sometimes not used at all. 'The transfer of new software engineering techniques and tools to common practice has had much more limited success; sometimes new ideas take hold immediately, but more often a new, proven idea takes many years to become accepted as standard practice' (DACS 1998). A study has reported that once a technology has been developed, it takes an average of 7.5 years for it to become widely available (Redwine et al. 1985). Weinberg has published data about limited success in software tool utilization (Weinberg 1997):

- 70% of tools purchased by the organizations surveyed are never used, other than perhaps in an initial trial;
- 25% of tools are used by only one team or person within an organization;
- 5% are widely used, but not to capacity, with perhaps only 10% of the capacity of the tool being used.

As stated by Krasner, successful technology transfer occurs when a perceived problem meets a technologically based solution within an appropriate interval of time and when there exists "commitment on both sides of the producer-consumer relationship at all organizational levels" (Krasner 1995). But technology transfer is successful only if the technology supports the business' success, because it is an innovation to an existing development process that helps reduce the development lead-time (shorter time-to-market), improving the quality of software in products, and/or reducing the development cost (Punter et al. 2009).

Since the ultimate objective of the development of any new standard is rapid and wide adoption and utilization, it is worthwhile discussing this topic at some length. Particularly because the ISO standard is being developed for VSEs, it is critical to understand technology transfer practices. As pointed out in a later chapter, VSEs are not very interested in standards. They mostly perceive them as expensive, not very useful, and 'bureaucratic' documents.

It is the experience of the author, who was involved for many years in software process improvement, that technology transfer is nearly as difficult as technology development. There are a number of reasons for this (Zelkowitz 1995):

- Infusion mechanisms do not address software engineering technologies as well as they do other technologies. This may be the result of software engineering's process orientation, where the focus is more on producing than on transferring a product.
- Technology transfer requires far more than simply understanding a new technology.
- Quantitative data are important for understanding how and why the new technology will fit into or replace the existing processes.
- Technology infusion is not cost-free.
- Most software professionals are resistant to change.
- Personal contact is essential for change.
- Technology transfer knowledge and skills are very different from those of technology development (e.g. *soft* skills versus *hard* skills).
- Timing is important.

The search for a technology

A business need prompts an organization to ask if there is a technology that might address that need. It is at this point that an organization starts looking for a technology. There are three possibilities: the technology exists and has been used before, the technology exists but has not been proven in practice, or the technology does not exist.

If the technology exists, it has to be selected. Once selected, the organization should search for evidence that the technology has worked in similar projects or organizations. If that evidence exists, the organization should be able to answer the following question: Is there enough evidence to justify selecting it and using it. If there is enough evidence and there is some support for it, such as a tool or training, the next step is to use the technology and evaluate it to confirm that it meets the business need that prompted this process in the first place.

Technology producers are organizations, research teams, or individuals who develop a technology such as a method, tool, or paradigm (e.g. object-oriented technology), and technology consumers are enterprises, departments, or projects that could be the users of a technology. Currently, for the ISO standard development project, the technology producers are the members of the ISO working group and the main consumers, or technology adopters, are VSEs worldwide. Eventually, if the standard is successful, technology producers will emerge from for-profit and not-for-profit organizations, such as organizations, governmental organizations, and consulting firms.

4.3.2 Technology transfer process

Pfleeger suggests a five-step process to successfully transfer technology, as illustrated in Figure 4-9. The notation, based on SADT, is as follows: the arrows on the left are inputs, the arrows on the right are outputs, the arrows from above are constraints, and the arrows from below are mechanisms for accomplishing the activity represented by the box. The five-step process is described below.

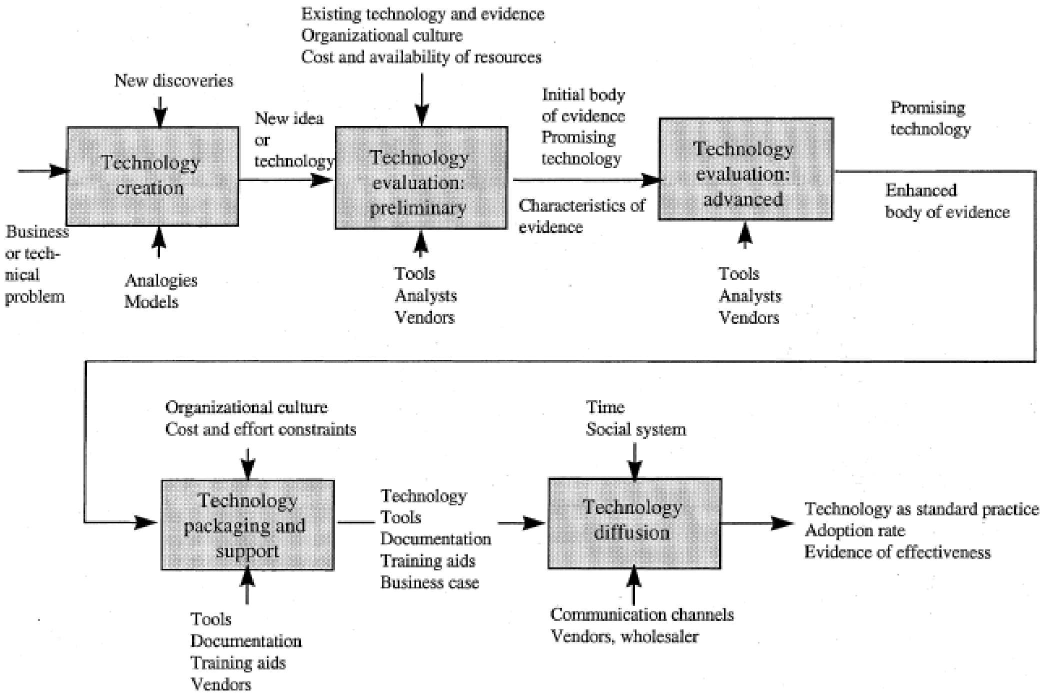


Figure 4-9 The Five-Step Technology Transfer Process (Pfleeger 1998)

The five-step technology transfer process (Pfleeger 1999)

STEP 1 – TECHNOLOGY CREATION

A technology is created as a result of a perceived need or a problem. Once the technology becomes visible, a potential user asks questions such as: What problem can the technology solve for me? Does it work properly? Does it replace/expand/enhance an existing technology? Does it fit easily into the existing development or maintenance process, without major disruption to established and effective activities? Is it easy to understand? Is it easy to learn? Is it easy to use? Is it cost-effective?

STEP 2 – PRELIMINARY TECHNOLOGY EVALUATION

At this stage, a potential user finds the technology interesting. The user must then evaluate the technology relative to the technologies already in use in the organization. A potential user asks questions such as:

- Are there any benefits to this technology over existing technology?
- Are the benefits greater than or less than those of the existing technology?
- To what degree is it easy to understand and use?

The potential user is looking for evidence of benefits from sources such as case studies or field studies. Unlike researchers, potential users rarely look for evidence in sources such as theoretical proof or simulations. If the evidence is not convincing, a potential user will likely stop evaluating the technology. If the evidence is convincing and the benefits are greater, the user may proceed to a more exhaustive evaluation.

STEP 3 – ADVANCED TECHNOLOGY EVALUATION

At this stage, the potential user examines the evidence to determine under what conditions the technology is likely to work best. A potential user asks questions such as:

- Is the evidence credible?
- Are the providers of the evidence credible?
- Is some of the evidence contradictory?
- If the evaluation is successful, the potential user will look for artifacts to facilitate or accelerate the utilization of the new technology.

STEP 4 – TECHNOLOGY PACKAGING AND SUPPORT

At this stage, the potential user will look for artifacts, such as documentation or tools, to facilitate or accelerate the adoption and utilization of the new technology. A potential user asks questions such as:

- Has the technology been commercialized and marketed?
- Is the technology used outside the group that developed it?
- If the appropriate artifacts are available, is the technology ready for diffusion?

STEP 5 – TECHNOLOGY DIFFUSION

At this stage, the technology is ready to be transferred to a wider audience. Rogers defined five categories of technology adopters (Rogers 2003). These are defined below. The topic of technology diffusion is also explained in more detail below.

4.3.3 Technology adoption and the steps towards institutionalization

The five-step model does not adequately explain the activities required for the utilization and institutionalization of a new technology. The topic is discussed in greater detail in chapters 7 and 8. The technology transfer is not completed until the technology is embedded in an organization, which means that it has become part of the organization's 'normal' way of working.

Table 4-10 presents the different degrees of involvement of the main actors in the technology transfer: the developers of the technology, in our case ISO WG24, the potential adopters/users of the technology, i.e. VSEs and the actors who will transfer the technology, which are the technology transfer centers, the consultants, and the educators.

In this table, the technology creation step is completed once the new standard being developed by WG24 is published by the ISO and becomes available for purchase by various actors, such as VSEs, technology transfer centers, consultants, government organizations, and systems integrators and educators. Government organizations and system integrators are organizations that could impose the use of the standards on VSEs. The degree of involvement of the actors is indicated using the following scale (adapted from Punter 2008):

- 0 = no involvement
- 1 = less involvement
- 2 = moderate involvement
- 3 = high involvement

Degree of Involvement	**Technology Creation**	**Preliminary Technology Evaluation**	**Advanced Technology Evaluation**	**Technology Packaging and Support**	**Technology Diffusion**
ISO Working Group 24	3	0	0	0	0
VSEs	0	2	2	2	3
Members of the Network to Support VSEs	1	2	2	3	3

Degree of Involvement	**Technology Creation**	**Preliminary Technology Evaluation**	**Advanced Technology Evaluation**	**Technology Packaging and Support**	**Technology Diffusion**
Technology transfer centers and Consultants	0	2	3	3	3
Government organizations and system integrators	1	2	2	2	3
Educators	0	2	2	2	3

Table 4-10 Degree of Involvement of Actors in Technology Transfer (adapted from Punter 2008)

4.3.4 Technology transfer inhibitors and promoters

Finally, Pfleeger has provided a list of factors (Table 4-11) that may promote or inhibit technology transfer (Pfleeger 1998).

	Technology Inhibitors	Technology Promoters
Technological	• Lack of "packaging" • No clear relationship with the technical problem • Difficult to use • Difficult to understand	• Supporting tools, manuals, classes, help • Clear benefit to the technical problem • Well-understood context for using the technology • Easy to use • Easy to understand
Organizational	• No management support • Cognitive dissonance • Biases and preconceptions • Cross-organizational mandate • Not cost-effective • Heterophyllous	• Gatekeeper • Technology booster • Trial in one organization • Results (especially improvement) visible to others • Perceived advantage • Success in similar organizations • Compatible with values • Compatible with experience • Compatible with needs • Cost-effective • Homophyllous
Evidential	• Conflicting evidence • Lack of evidence • Meaning of evidence unclear • Experiments in toy situations	• Consistent evidence • Case studies and field studies • Credible messenger • Relative advantage • Cause-and-effect evident

Table 4-11 Technology Transfer Inhibitors and Promoters (Pfleeger 1999)

WG24 will have to walk through this list, identify the various inhibitors and promoters, and address them during the development of the technology transfer activities. These activities will be presented in detail in a later chapter.

4.3.5 Categories of ideal technology adopters

Rogers defined five categories of technology adopters using a criterion called *innovativeness,* which means the degree to which an individual or other unit adopts a new idea relatively sooner than other members of a social system (Rogers 2003). The five categories of adopters are: innovators, early adopters, early majority adopters, late majority adopters, and laggards. The main characteristics are (adapted from (Rogers 2003) and (Garcia et al. 2007)):

- **Innovators** – They are the first people to adopt a technology. They have substantial financial resources to absorb possible losses from an unprofitable innovation, and can cope with a high level of uncertainty about an innovation. They comprise about 2.5% of the total number of likely consumers or users. They are driven by a desire to be rash and to do something daring, will spend hours trying to get the technology to work, and do not need good quality documentation. They could also serve as helpful critics.

- **Early adopters** – They are more integrated into an organization's development culture, are respected by their peers, act as role models for those around them, and use new ideas discreetly. They comprise about 13.5% of the total number of likely consumers or users. By making judicious innovation decisions, they decrease the uncertainty of a new idea by adopting it and personally informing colleagues of its success.
- **Early majority adopters** – They are deliberate in their decision-making, thinking for some time before embracing a new technology, although they seldom hold positions of leadership in terms of opinion among those around them. They comprise about 34% of the total number of likely consumers or users. They follow, rather than lead, but are willing to try new things demonstrated to be effective by others.
- **Late majority adopters** – They are more skeptical, and adoption by them may be the result of economic pressure or peer pressure. Most of the uncertainty about the innovation must be removed before they adopt it. They comprise about 34% of the total number of likely consumers or users. Most of the uncertainty about a new idea must be resolved before a late adopter will agree to try it. They will not volunteer for an improvement project.
- **Laggards** – They jump on the bandwagon only when they are certain that a new idea will not fail, or when they are forced to change by mandate from managers or customers. They are suspicious of innovations and change agents, and must be certain that the innovation will not fail. They comprise about 16% of the total number of likely consumers or users. They avoid improvements until they have no other choice.

Figure 4-10 illustrates the approximate percentage of individuals in the five categories. The figure shows that, between the early adopter and early majority adopter categories, there is a discontinuity in the distribution called a *chasm* (Moore 2002). Rogers claims that there is no research to support this claim.

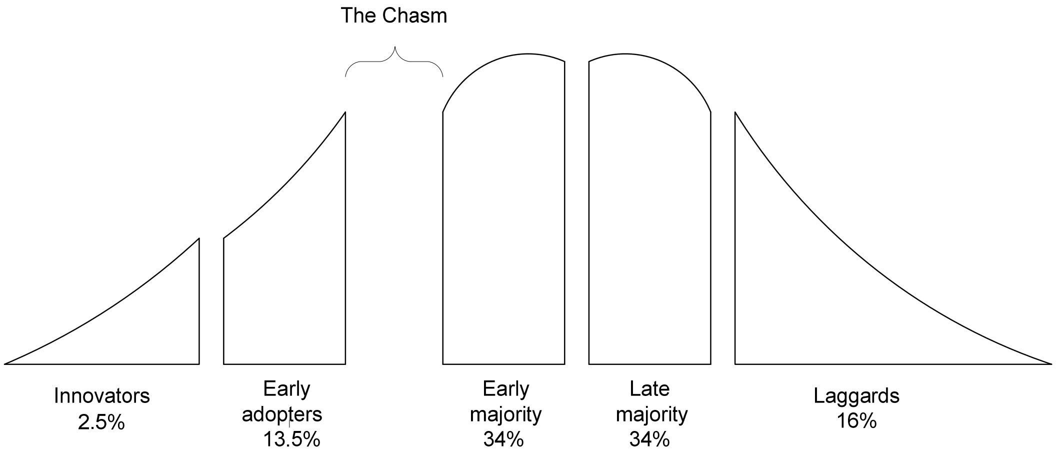


Figure 4-10 Technology Adoption Curve (Rogers 2003) and the Chasm (Moore 2002)

By creating a bandwagon effect, it could be possible to use the results obtained from the first group of adopters, i.e. the innovators, to convince the other groups one after the other. The goal is to achieve a critical mass of users, so that the innovation becomes self-sustaining (Rogers 2003). As an example of this critical mass effect, the Internet was introduced by the US Department of Defence, in the late 1960s. Critical mass was reached in about 1990, as illustrated in Figure 4-11, when the rate of adoption skyrocketed.

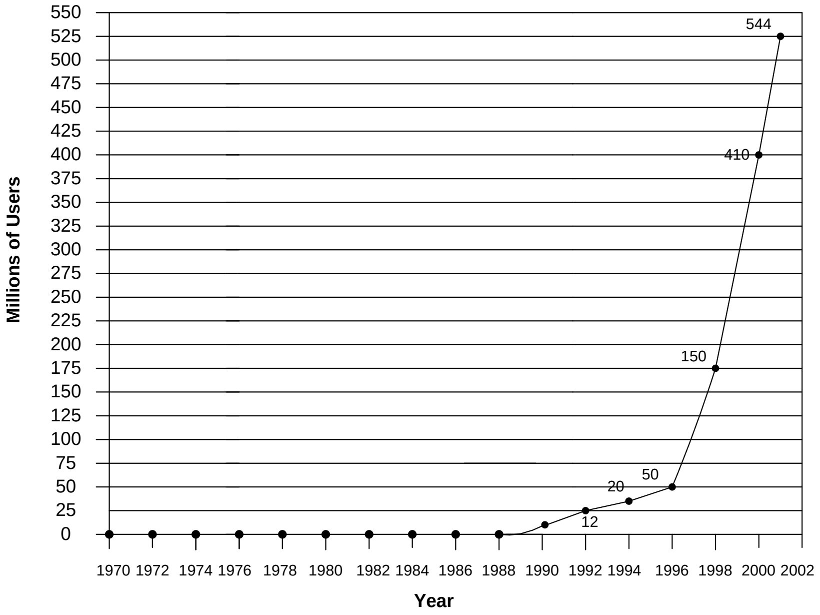


Figure 4-11 Adoption Rate of the Internet (Rogers 2003)

A technology is considered to be embedded in the adopter organization when it has been used for a significant period of time in a number of projects, and is considered to be sustainable when its application no longer depends on the people who originally introduced it, like the so-called *champions* who promote a technology among their colleagues (Punter et al. 2009).

Approaches to technology transfer

Once a technology has been selected, we can choose one or more approaches to encourage the transfer of that technology (adapted from DACS 1998):

- **A people-mover model.** Relying on personal contact between technology producer/developer and consumer/user.
- **A communication model**. A new technology is reported in print and noticed by the potential user.
- **An on-the-shelf model.** Packaging and ease-of-use make the technology appealing to potential users.
- **A vendor model**. An organization relies on its primary vendor to be gatekeeper, promoting new technologies when appropriate.
- **A rule model.** An outside organization imposes use of a technology on the development organization. This model can be invoked from inside, as when an organization imposes a development standard, or from outside, as when the customer mandates a development process or language standard, such as ISO 12207.

The last approach, the rule model, is the one currently used by large government organizations, such as defense and transport departments, or by large enterprises. Finally, a mapping between the categories of adopters, their perceived level of risk and the adoption approaches is presented in Table 4-12.

Adopter category	Level of risk (perceived by the adopter)	Adoption approach
Innovators	Very high	People-mover model
Early adopters	High	Communication model
Early majority adopters	Moderate	On-the-shelf model
Late majority adopters	Low	Vendor model
Laggards	Very low	Rule model

Table 4-12 Relationships Among Adopters, Risk, and Likely Transfer Models (Pfleeger 1999)

The strategy proposed to ISO WG24 to accelerate transfer to VSEs is presented in a later chapter. The reader is reminded that technology transfer activities are not included in the official mandate of an ISO working group. The author convinced the members of WG24, at the first meeting in 2005 in Bangkok, that technology transfer activities were critical to the adoption and utilization of the future standards by VSEs.

4.3.6 The process of commitment toward a technology

Connor has developed a 3-phase, 7-stage technology commitment process: the preparation phase, the acceptance phase, and the commitment phase (Connor 1992). The vertical axis, in Figure 4-12, indicates the level of support or commitment, and the annotations along the time axis indicate the technology transfer failures that can result from not implementing activities to support the adopter through each of the seven stages.

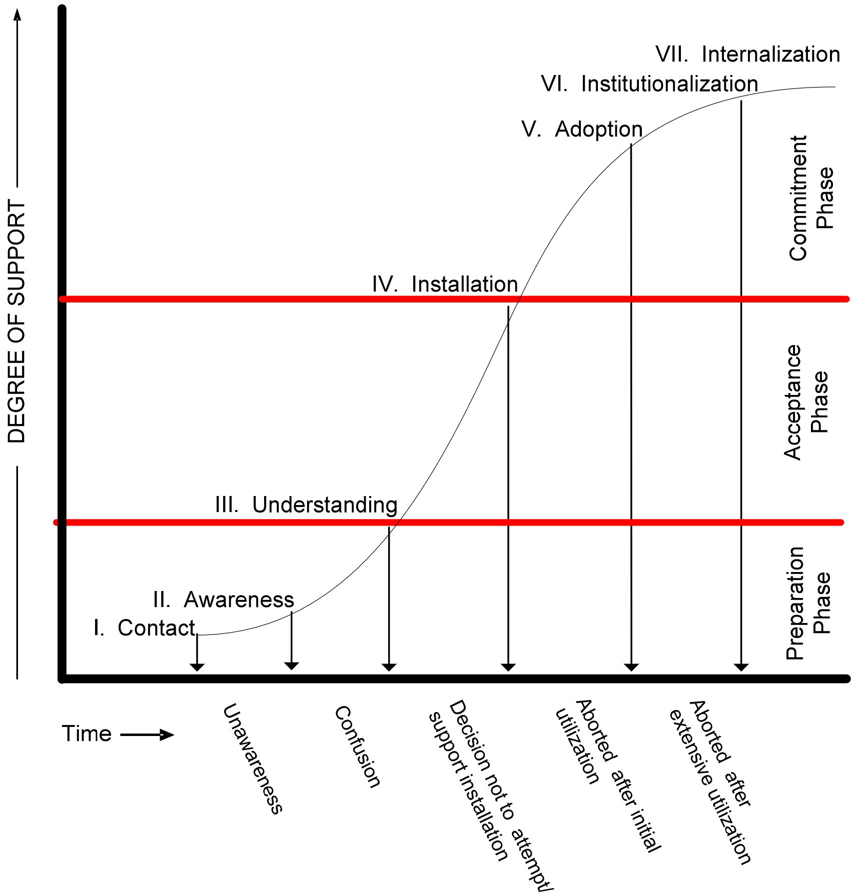


Figure 4-12 Technology Adoption Life Cycle Model (Connor 1992)

An important feature of this model is that, at each stage of the commitment process, the commitment level can completely disappear, even if the adopting organization has passed the adoption level. For example, many organizations bought CASE tools in the 1980s, only to discover months after their installation that they were not being used. In other words, the organization had not reached the institutionalization phase.

Communication activities, such as briefings, serve to help the organization move through the first three stages. Then, implementation activities will help the organization move through the last four stages. (Garcia et al. 2006). Table 4-13 lists typical communication mechanisms for each stage of the commitment process. Some of these mechanisms have already been used to communicate the deployment of the international survey of VSEs, as well as to report the results of the survey (presented in a later chapter). Some of the communication mechanisms illustrated in the table below will be used in a subsequent chapter, when the technology transfer communication plan and action plan are developed.

Commitment Stage	Typical Mechanisms
Contact and Awareness	- "Elevator speech" -Standard 45-minute pitch, road show -Magazine articles -Conference briefings -Web site devoted to the technology, with links and dialog -Successful ROI stories, case studies
Understanding	1-day seminar Detailed case studies Technical brief Champion identified and authorized
Trial use	Pilot programs Defined incentives for pilot participation User group Lessons learned Barriers identified and workarounds devised Technology use start-up and coaching
Adoption	Education: mature courses Repository on business cases and lessons learned Job aids: process guides, start-up guides, coaching
Institutionalization	Curriculum for training different types of users New employee training/orientation

Table 4-13 Typical Transition Mechanisms Categorized by Adoption Commitment Curve Category (Garcia et al. 2006)

Technology diffusion and infusion

When we discuss technology transfer within an organization, we define *technology diffusion* to mean how broadly a new technology has been diffused within an organization, and *technology infusion* to mean how deeply a new technology has penetrated the organizational environment of the intended audience (adapted from Garcia et al. 2007).

For WG24, we can redefine these terms, since we are not focusing on the transfer a technology within an organization, but to VSEs worldwide:

- Technology diffusion – in how many countries have the new standards been used by VSEs?

- Technology infusion – how deeply have the new standards penetrated VSEs worldwide (e.g. percentage of VSEs, number of business domains)?

Table 4-14 lists typical questions asked by participants of a technology transfer project, as well as activities at each step of the technology diffusion process (Garcia et al. 2007).

Stage	Key Question	Successful Exit Criteria	Potential Events to Measure
Contact/ Awareness	What is it (in this case, the model being used for the improvement effort)?	-could concisely answer: What is it? Why should I use it? Who is it for? How is it different from current practice?	-attended awareness meetings -received newsletters with articles on improvement effort
Understanding	What does it mean to us? How will we have to change?	-detailed knowledge of what it will take to use new processes -beginning to think through changes to work practices and the environment needed for new processes to be successful -level-of-use goals have been defined	-participated in working meetings to define work practice changes -end user: attended detailed user course for their role on the new process -PI specialist: attended detailed course on model used as basis for developing improved processes
Trial Use	How will we implement it?	-explicit decision to go forward with widespread adoption	-participated in at least one pilot activity -has tried the new process for at least one cycle of typical tasks
Adoption	Are we really using it?	-initial levels-of-use goals have been achieved (assumes they have been specified) -not using the improved processes is considered an exception to general work practices	-not using the improved processes is noted in reviews/audits -individuals routinely complete their work assignments using the improved processes
Institutionalization	What do we need to keep it in place?	-new processes have long history of worth, durability, continuity -NOT using the new processes has defined negative consequences	-all infrastructure (licenses, etc.) needed to support the improved processes have been provided/kept up to date -use of the improved processes is visible in the relevant policies/procedures -performance metrics include reference to the process use

Table 4-14 Diffusion Event Suggestions (Garcia et al. 2007)

WG24 will use some of these mechanisms to assess the degree of diffusion of the new standards. In the next section, the author discusses the cultural issues that have to be considered when developing a technology targeted toward many cultures.

4.3.7 Cultural issues and the development of International Standards

As stated by Phongpaibul, “Cultural difference is one of the key successes of an adoption of software processes” (Phongpaibul et al. 2005). In this section, the cultural issues related to the development and deployment of IS are described, following an explanation of the model, developed by the Dutch psychologist Geert Hofstede, that will illustrate the author’s ideas.

The Thai culture is used here to demonstrate the impact of culture on the development of software engineering processes and software standards, specifically on the utilization of pair programming, inspections and requirements elicitation, and process improvement. The author had the opportunity to teach software engineering courses to professionals in Bangkok and Chiang Mai between 2005 and 2007, and to undergraduate software engineering students at Chiang Mai University in 2008. The assignment comprised a one-day peer review course (i.e. inspection and walk-through) taught to four groups of software professionals and a four-week software quality assurance course taught to undergraduate software engineering students . The author also attended 3 one-week meetings in Bangkok in 2005 and 2006.

In the last section of this chapter the impact of national culture on the development and deployment of software IS for VSEs is described.

Geert Hofstede’s cultural dimensions

In the late 1960s, Geert Hofstede conducted a comprehensive study of how values in the workplace are influenced by culture. He collected and analyzed data on over 100,000 individuals from forty countries while working at IBM as a psychologist. His model has been criticized for using a single organization and for suggesting that the five dimensions are not enough to describe the many aspects of a culture (Thanasankit et al. 1999). Since the model is very well known, the author used it for the analysis of cultures, especially for the preparation of the worldwide deployment of the VSE standards in 2010.

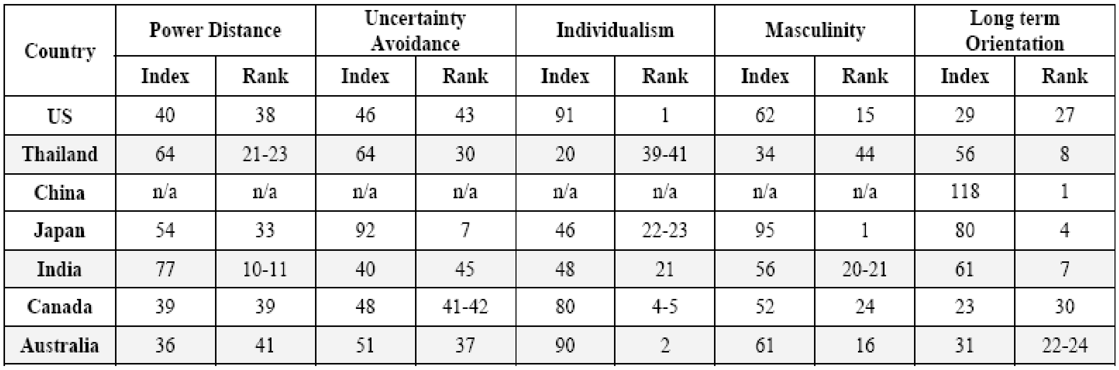
Hofstede has stated that all societies share the same basic problems: equality versus inequality, group solidarity, gender roles, an uncertain future, and need gratification (Hofstede 1997). All societies have developed solutions to these problems, but those solutions differ from one society to another. Hofstede defines culture as the collective programming of the mind (i.e. mental model) which distinguishes the members of one group or category of people from those of another. He developed a model that identifies five primary dimensions to differentiate cultures: power distance, uncertainty avoidance, individualism, masculinity, and long-term orientation.

These five dimensions are described as follows (adapted from (Hofstede 1997), (Hofstede et al. 2002), and ([40])):

- **Power Distance Index** is the extent to which the less powerful members of organizations and institutions (like the family) accept that power is distributed unequally and expect this to be so. This represents inequality (more versus less) in terms of prestige, power, and wealth. It suggests that a society's level of inequality is endorsed by the followers as much as by the leaders. Power and inequality are fundamental facts of life in any society, and anyone with a modicum of international experience will be aware that 'all societies are unequal, but some are more unequal than others' (Hofstede et al. 2002). In high Power Distance Index countries (e.g. Malaysia, Thailand), the less powerful accept power relationships that are more autocratic and paternalistic. In a lower Power Distance Index society, like Canada, a boss and his or her subordinates are more equal.
- **Individualism Index** on the one hand versus its opposite, collectivism, on the other is the degree to which individuals are integrated into groups. On the individualist side, we find societies in which the ties between individuals are loose: everyone is expected to look after himself/herself and his/her immediate family. On the collectivist side, we find societies in which people from birth onwards are integrated into strong, cohesive in-groups, often extended families (including uncles, aunts, and grandparents), which continue protecting one another in exchange for unquestioning loyalty. The word *collectivism* in this sense has no political meaning: it refers to the group, not to the state. Again, the issue addressed by this dimension is an extremely fundamental one, and applies to all societies in the world. The Latin American and Thai cultures rank among the most collectivist in this category, while the US rank highest in terms of individualism.
- **Masculinity Index** versus its opposite, femininity refers to the distribution of roles between the genders, which is another fundamental issue for any society, and for which there is a range of solutions. The IBM studies revealed that: (a) women's values differ less among societies than men's values; and (b) men's values from one country to another are characterized by a dimension ranging from very assertive and competitive and maximally different from women's values on the one hand, to modest and caring and similar to women's values on the other. The assertive pole has been called *masculine* and the modest, caring pole *feminine*. The women in feminine countries (e.g. Thailand) have the same modest, caring values as the men; in the masculine countries (e.g. Japan) they are somewhat assertive and competitive, but not as much as the men, so that these countries show a gap between men's values and women's values.
- **Uncertainty Avoidance Index** deals with a society's tolerance for uncertainty and ambiguity; and ultimately refers to man's search for absolute Truth. It indicates to what extent a culture programs its members to feel either uncomfortable or comfortable in unstructured situations. Uncertainty-avoiding cultures try to minimize the possibility of such situations by strict laws and rules, safety and security measures, and on the philosophical and religious level by a belief in absolute Truth: for them, there can only be one Truth, and they have it. People in uncertainty-avoiding countries are also more emotional, and motivated by inner nervous energy. The opposite type, uncertainty-accepting cultures (e.g. India), are more tolerant of opinions different from those they are used to; they try to have as few rules as possible, and on the philosophical and religious level they are relativist and allow many currents to flow side by side. People within these cultures are more phlegmatic and contemplative, and not expected by their environment to express emotions.
- **Long-Term Orientation Index**, as opposed to short-term orientation. This fifth dimension was found in a study of students in 23 countries around the world, using a questionnaire designed by Chinese scholars. It can be said to deal with Virtue regardless of Truth. Values associated with long-term orientation are thrift and perseverance toward the achievement of slow results, funds available for investment, and willingness to subordinate oneself for a purpose; values associated with short-term orientation are respect for tradition, little money for investment, the expectation of quick results, the fulfillment of social obligations, and saving face.

[40] http://www.geert-hofstede.com/hofstede_canada.shtml

Table 4-15 lists the five-dimensional scores for a few countries, as well as the ranking of those countries in the world. Although each country has only one set of scores, there is sometimes more than one cultural group in that country. For example, in Canada, there are cultural differences between the French-speaking population of the province of Quebec and the English-speaking population of other provinces.



Country	Power Distance		Uncertainty Avoidance		Individualism		Masculinity		Long term Orientation	
	Index	Rank	Index	Rank	Index	Rank	Index	Rank	Index	Rank
US	40	38	46	43	91	1	62	15	29	27
Thailand	64	21-23	64	30	20	39-41	34	44	56	8
China	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	118	1
Japan	54	33	92	7	46	22-23	95	1	80	4
India	77	10-11	40	45	48	21	56	20-21	61	7
Canada	39	39	48	41-42	80	4-5	52	24	23	30
Australia	36	41	51	37	90	2	61	16	31	22-24

Table 4-15 Hofstede's Index Scores and Ranks for Countries
(Phongpaibul et al. 2005)

Cultural impacts in software engineering processes and practice

In this section, some of the cultural differences between the Thai culture and the American culture with regard to the application of software engineering practices are illustrated.

The Power Distance Index of 64 for Thailand, versus 40 for the United States, indicates a high level of inequality of power and wealth within the Thai society. A high Power Distance Index is one of the driving factors in Thai business. Decision-making is dependent on the person with the higher rank. With a high Power Distance Index, Thai managers can force the team to follow the processes. But, since the members of a team wait for the project managers to tell them what to do, they do not try to improve their software engineering capabilities by themselves. In Thailand, if a company has project managers who are younger than the developers, the project managers cannot manage the team because of a high Power Distance Index in terms of age (Phongpaibul et al. 2005). For example, in the first course taught by the author in Bangkok, students did not ask questions in class, even in response to an invitation to do so. Since the mere fact of a student asking a question might imply that the professor did not explain the topic well, they would wait for the break to ask their questions. After the break, the author made it clear that he wanted students to ask questions. The students responded immediately and enjoyed the opportunity to do so right through to the end of the class. A high Power Distance Index is an advantage in the workplace, since employees tend to wait for a manager to tell them how to perform a task without any conflict. At the same time, if employees implement the process without asking for clarification, the process will not be improved even if it is appropriate for the project. They will wait for the manager's approval to make any improvements. Finally, since the cost of training is high, only managers attend courses and possess detailed knowledge the about a process (Phongpaibul et al. 2005).

The Uncertainty Avoidance Index of 64 for Thailand versus 46 for the United States indicates that former society's low level of tolerance for uncertainty. In Thailand, this characteristic eases the process of software implementation and improvement, since the improved process will provide better guidance.

The high Individualism Index of 91 for the United States, versus 20 for Thailand, indicates that Thai society is collectivist. Thai software development businesses are relationship-oriented. A project can only be won as a result of a connection with someone in that company, and so bidding does not require the CMMI. Companies do not need to process improvement for better quality of product to keep their business running. Instead, they can rely more on their collaborators putting in extra effort when needed to preserve good relationships (Phongpaibul et al. 2005).

Thailand has a Masculinity Index of 34, compared to the United States with 62. This lower level is indicative of a society with less assertiveness and competitiveness. The culture of femininity does not affect the adoption of software process models and improvements in Thailand (Phongpaibul et al. 2005).

The Long-Term Orientation Index is the lowest scoring dimension for the United States at 29, compared to the Thai score of 56. However, a practitioner may be promoted to manager because he has been working for the company for a long time, even though he does not have the appropriate management skills (Phongpaibul et al. 2005).

Application of pair programming and inspection practices[41]

A classroom experiment was performed to study the commonalities and differences between pair programming and software inspection in Thailand.

Inspection, as defined in the IEEE Standard for Software Reviews and Audits (IEEE 2008a), is a visual examination of a software product to detect and identify software anomalies, including errors and deviations from standards and specifications. It is the most highly structured process for detecting defects in software artifacts and removing them. The inspection is defined by the IEEE standard as follows:

> *Inspections consist of two to six participants (including the author). An inspection is led by an impartial trained facilitator who is trained in*

[41] Adapted from (Phongpaibul et al. 2006)

> *inspection techniques. Determination of remedial or investigative action for an anomaly is a mandatory element of a software inspection, although the resolution should not occur in the inspection meeting. Collection of data for the purpose of analysis and improvement of software engineering procedures is a mandatory element of software inspections.*

Laurie Williams defined pair programming as follows:

> *Pair programming is a style of programming in which two programmers work side-by-side at one computer, continuously collaborating on the same design, algorithm, code, or test. One of the pair, called the driver, types at the computer or writes down a design. The other partner, called the navigator, has many jobs. One is to observe the work of the driver looking for defects in the work of the driver. The navigator has a much more objective point of view and is the strategic, long-range thinker. Additionally, the driver and the navigator can brainstorm on-demand at any time. An effective pair programming relationship is very active. The driver and the navigator communicate, if only through utterances, at least every 45 to 60 seconds. Periodically, it's also very important to switch roles between the driver and the navigator (Williams et al. 2002).*

During the experiment, the authors measured the Total Development Cost (TDC). TDC is the amount of money that the team spends on producing the system, called *production cost*, plus what the team spends on ensuring the quality of the system, called *CoSQ*. CoSQ is composed of four categories of cost: prevention costs, appraisal costs, internal failure costs, and external failure costs.

The average TDC for the pair development group is 527 man-hours and the average TDC for the inspection group is 695 person-hours. The pair development group expended 24% less effort than the inspection group for the same quality. Development required continuous interaction between the developers, which may not be a practical approach in the United States, where people generally require more personal space (Phongpaibul et al. 2005).

Often, team members became embroiled in arguments during the meeting and the moderator could not stop them since he was one of their classmates. Members preferred to have the project manager present during inspection meeting, since this individual has more power (i.e. a high Power Distance Index) than the moderator to stop an argument and move the inspection forward.

In (IEEE 2008a), (Fagan 1976), and (Gilb et al. 1993) suggested that, during an inspection meeting, the inspectors not be allowed to identify final solutions for the defects, but rather that the defects be corrected by the manager after the inspection meeting. This inspection method, wherein the manager can solve the problem without help from the team, was originally developed in the United States (the US ranks highest for individualism). It differs from the Thai approach, which is collectivist in nature, and where the best solution comes from the team and not from an individual.

IEEE 1028 also states: "Individuals holding management positions over any member of the inspection team shall not participate in the inspection." This is to avoid participants identifying superficial defects in front of a superior. Because Thai culture has a high Power Distance Index, not having a project manager at the inspection meeting results in the meeting lacking energy.

The impact of Thai culture on requirement elicitation[42]

Thanasankit et al. describe the impact of Thai culture on the requirements elicitation process, indicating that, because of the strong power distance between managers and subordinates and the fact that those subordinates are not comfortable making decisions, requirements must be sent up the hierarchical ladder for approval. Most Thai development teams are organized like a steering committee rather than a working team, in contrast to the practice in Western countries. The requirement elicitation process is tedious and slows down the approval of requirements.

Also, because Thai people want to have stable relationships and maintain harmony, they want to work in a smooth, pleasant, polite, and conflict-free environment. A junior requirement analyst may not contradict a manager, even though the analyst may be right. Requirements from the top of the organization have precedence over requirements gathered by low-level managers and analysts.

Since Thai society is a collective one and they express concern among themselves, they may resist requirements that could pose a threat to another employee. They will resist the requirement elicitation process by finding excuses for not being able to attend meetings.

Since Thai society is an uncertainty avoidance society, there is fear in ambiguous situations and unfamiliar risks. Thai managers seek their security and focus on the short term and less on strategic planning (Jirachiefpattana 1996). This cultural trait has an impact on the selection and utilization of elicitation methods such as interviews. For example, a young analyst will be very careful when interviewing a senior member of an organization, to ensure that the senior member is not embarrassed by not knowing an answer to a question. The analyst will also be careful not to make mistakes and avoid being seen to be wrong or to not completely understand the system (Thanasankit et al. 2000).

[42] Adapted from (Thanasankit et al. 2000)

The impact of national culture on the development and deployment of international standards for VSEs

Most current software process models or improvement models are developed by either the American or European standards committees, such as the SEI, which developed the CMM and CMMI frameworks. The culture dimensions provided by Hofstede may explain low software process adoption rates in certain cultures. One example is the low adoption rate by organizations in the more collective/feminine/long-term-focused Thai culture compared to the more individual/masculine/short-term-focused American culture's Software CMM (Phongpaibul et al. 2005).

Unlike other Asian countries, Thailand has only one official language (Jirachiefpattana 1996). This dependence on the one language makes Thai delegates less productive during working group meetings, since they have difficulty expressing their ideas or may not argue with another delegate to preserve harmony and not lose face. Also, a junior delegate at a meeting may not express his views if they contradict those of a senior member of the delegation.

The majority of delegates come from a European culture or the American culture. This may affect the selection of practices that will be imposed by the VSE standard. For example, the imposition of a practice, such as an inspection, may well be correct for the other cultures, but may pose some problems to Thai practitioners, who would prefer a less formal approach, such as a walk-through or desk-check. However, the ISO standard development process does, in fact, have a built-in mechanism to compensate for national culture: all countries may participate in the development and balloting of standards. A country may also submit comments to request modifications to committee drafts before the standard is published. As illustrated in Table 4-15, because of the wide variety of countries and cultures participating in WG24, the resulting standard has more chance of being acceptable to all countries and used by all countries.

With regard to countries with a high Power Distance Index, the adoption and deployment of standards should be greatly facilitated by the authorities in those countries or by the adopting organizations, since they will likely impose the standards on VSEs.

The topic of technology transfer has been introduced here. In the next part, some technology transfer activities are presented which have been conducted since the establishment of the working group. In chapter 7, what has been done to facilitate the adoption and utilization of VSE standards by providing guidance, examples, templates, and checklists is explained in greater detail.

4.4 Part 3 – History leading to the establishment of an ISO/IEC JTC 1/SC7 Working Group for VSEs and recent achievements[43]

Here, a history of the events leading to the creation of the ISO/IEC SC7 Working Group, WG24, is presented, as well as its recent achievements up to the May 2009 meeting in India. The official accomplishments of WG24, i.e. activities accomplished according to the official mandate approved by the ISO, are described, as are some of the unofficial activities performed, i.e. those performed outside the official mandate of the working group such as the conducting of a survey.

Even though most of the activities described in this part cover phase 3 of the innovation process illustrated in Figure 4-1, and reproduced here as Figure 4-13, some activities that cover phase 4 the commercialization phase are presented as well.

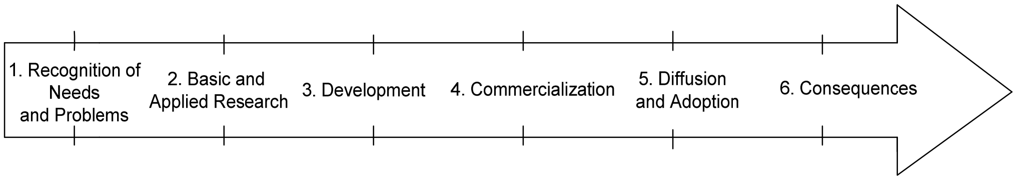

1. Recognition of Needs and Problems
2. Basic and Applied Research
3. Development
4. Commercialization
5. Diffusion and Adoption
6. Consequences

Figure 4-13 Phases of the Innovation Process (adapted from Rogers 2003)

4.4.1 Plenary meeting of ISO/IEC JTC 1/SC7 – Australia, 2004

At the Brisbane meeting of SC7 in 2004, Canada's representatives raised the issue of small enterprises requiring standards adapted to their size and maturity level. The current software engineering standards target (or are perceived as targeting) large organizations. Australian's delegates supported Canada's position in this regard, and the two NBs took action to investigate possible ways forward. A meeting of interested parties was held with delegates from five NBs (Australia, Canada, the Czech Republic, South Africa, and Thailand), at which a consensus was reached on the general objectives:

- To make the current software engineering standards more accessible to VSEs;
- To provide documentation requiring minimal tailoring and adaptation effort;
- To provide harmonized documentation integrating available standards:
  - Process standards
  - Work products and deliverables
  - Assessment and quality modeling and tools

[43] This section has been adapted from an article originally published in (Laporte et al. 2008c)

- To align profiles, if desirable, with the notion of maturity levels presented in ISO/IEC 15504.

It was also decided that a special interest group (SIG) be created to explore these objectives, and to better articulate the priorities and a project plan. The participants felt that it would be possible, during 2004, to draw up:

- A set of requirements;
- An outline of key deliverables, and the associated processes to create them (e.g. how to create profiles);
- A Terms of Reference document for the working group;
- An example of a simple profile.

The participants proposed the calendar of events illustrated in Table 4-16.

Phase 1 (July 31st)
Canada will produce a draft of the first two documents for review for the end of July, so that the work can be then allocated to multiple participants over the next six months.

Phase 2 (December 31st)
Participant will work on an item selected from the initial document.

Phase 3 (February 28th)
Material will be assembled and edited.
If required, draft proposals to SC7 will be prepared and circulated.

Phase 4 (April 30th)
Material will be reviewed and improved by participants.

Phase 5 (May 20th)
If required, final proposals to SC7 will be prepared and tabled at the Advisory Group Meeting of SC7.

Table 4-16 Proposed Calendar of Events

4.4.2 First Special Working Group meeting in Thailand, 2005

In March 2005, the Thailand Industrial Standards Institute (TISI) invited software engineering experts to be part of a Special Working Group (SWG) to advance the work items defined at the Australia meeting. The meeting was attended by participants from the following countries: Australia, Belgium, Canada, the Czech Republic, Finland, South Africa, South Korea, Thailand, and the United States.

Risks perceived by organizations of different sizes

A key topic of discussion was to clearly define the size of VSEs that would be targeted by the working group. The group used a paper published by the Centre for Software Process

Technologies (McFall et al. 2003) to help define the size of small organizations. McFall presented the various perceived priorities and areas of concern for different organization sizes. The author presented the participants with the priorities and concerns of organizations with fewer than 20 employees, as illustrated in Figure 4-14, which are quite different from those of larger organizations. For example, medium and large organizations rank process adherence higher than do small organizations. For the latter, managing risk is of great concern, while for larger organizations this ranks much lower, as priority number 8. Conversely, for small organizations, consistency across teams is less of a concern, while for larger organizations it is a top-priority issue.

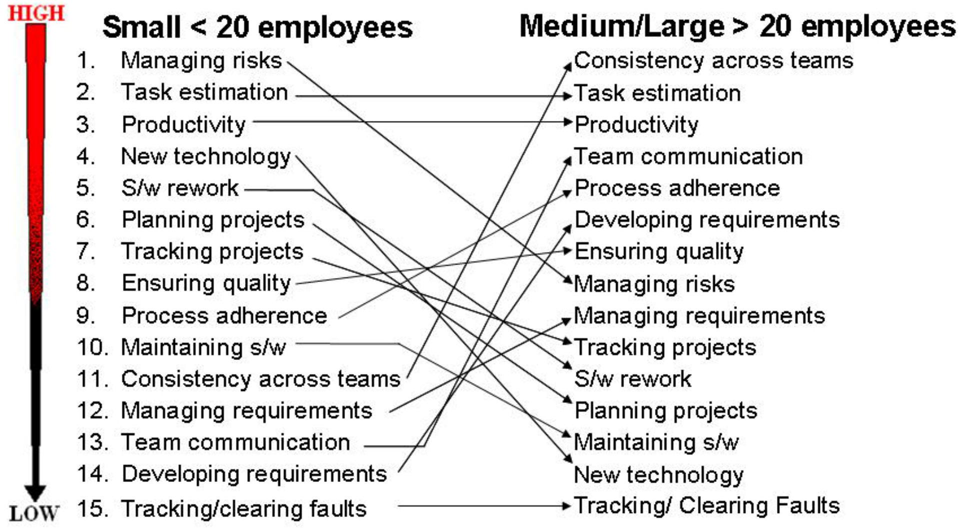


Figure 4-14 Priority and concern differences based on organization size (McFall et al. 2003)

A consensus was achieved by the members of the SWG on defining the targeted VSEs as IT services, organizations, and projects with between 1 and 25 employees.

A list of actions that could be undertaken by a future ISO/IEC SC7 working group was developed at this meeting. The proposed action items are the following:

- validate the work products produced by the working group;
- prepare, conduct, analyze, and communicate survey results;
- search for other centers/organizations focusing on SMEs and VSEs;
- assemble a complete list of characteristics of VSEs and projects;
- generate multiple profiles from the standards mentioned above;
- prepare communication material to inform VSEs about the work performed by the WG;

- develop business cases for the adoption and deployment of work products developed by the WG;
- develop one or more ISO 12207 roadmaps;
- conduct pilots of the roadmaps (using an approach similar to the trials conducted by the members of the working group that developed the ISO/IEC 15504 standard, i.e. the SPICE project).

A work schedule has also been developed for the new working group. As illustrated in Figure 4-15, the top row shows the standard ISO steps for the development and approval of an ISO standard. The lower part of the figure illustrates the unofficial actions, called the *shadow process*, which would need to be performed, as well as their expected date of completion, in order to obtain a CD1 (Committee Draft 1) by the end of 2007.

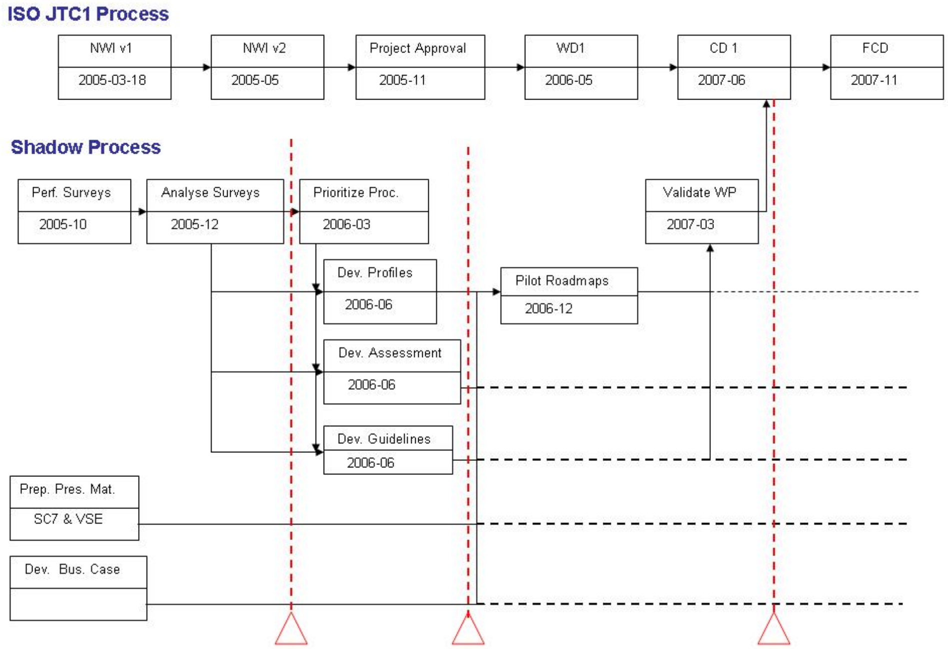


Figure 4-15 Proposed work schedules for the new working group (Laporte et al. 2008c)

The major output of this one-week meeting was a document, titled New Work Item Proposal (NWIP), as described in a previous chapter. This document had been presented and discussed at the Finland meeting held in May 2005 and is presented in the next section.

The reader may have noted that the original schedule was, like most software projects, optimistic. However, all the activities have been conducted in the sequence proposed at the 2005 meeting. The shadow activities are described in more detail in a later chapter.

4.4.3 Plenary meeting of the ISO/IEC JTC 1/SC7 meeting – Finland, May 2005

The New Work Item Proposal (NWIP) developed in Thailand was reviewed at the May 2005 SC7 plenary meeting in Helsinki. At that meeting, a resolution was approved, as follows: 'JTC1/SC7 instructs its Secretariat to distribute for letter ballot an updated version of [the] New Work Item Proposal for the development of Software Life Cycle Profiles and Guidelines for use in Very Small Enterprises (VSE) by 20 June 2005 (ISO 2005b).'

Balloting on this document remained open until September 21, 2005. Over 12 countries voted in favor of the NWIP. The following countries indicated a commitment to participate in the new working group: Belgium, Canada, the Czech Republic, Ireland, Italy, Japan, Korea, Luxemburg, South Africa, Thailand, the UK, and the United States.

As a result of this vote, the project was approved and the new working group, WG24, was established as follows:

- Mr. Tanin Uthayanaka (Thailand), Convener.
- Mr. Claude Y. Laporte (IEEE Computer Society), Project Editor.
- Mr. Jean Bérubé (Canada), Secretary.

Proposed project tabled at ISO/IEC JTC 1/SC7

The document tabled at the SC7 Helsinki plenary meeting describes the scope and purpose of the proposed working group, the justification for it, and its vision statement. Each element of that project is presented below, with text extracted from the document balloted by the ISO (ISO 2005b).

PROJECT SCOPE

- Organizations and projects with fewer than 25 employees;
- Current scope of ISO/IEC 12207 and its amendments, the associated guidance document, and other relevant SC7 Standards (e.g. ISO/IEC 15504, ISO/IEC 90003);
- Production of technical reports (Guides) establishing a common framework for describing assessable life cycle profiles used in VSEs, including small software systems development departments and projects within larger organizations;
- Guides to be based on International Standardized Profiles (ISP) identifying which parts of the existing standards are applicable to VSEs, at a specific level and for a specific domain;

- Guides which can be applied throughout the life cycle for managing and performing software development activities, the ultimate goal being to improve the competitiveness and capacity of VSEs.

PURPOSE AND JUSTIFICATION

The software systems industry as a whole recognizes the value of VSEs in terms of their contribution of valuable products and services. The majority of software organizations fall within the VSE category in terms of size. From the various surveys conducted by some of the NBs that initially contributed to the development of this NWI list, it is clear that the current SC7 Life Cycle Standards (ISO/IEC 12207 and the related Guide) are a challenge to use in these organizations, compliance with them being difficult (if not impossible) to achieve. Consequently, VSEs have few, or a very limited number, of ways to be recognized as organizations producing quality software systems, and therefore they do not have access to some markets. Currently, conformity with software engineering standards requires a critical mass in terms of number of employees, cost, and effort, which VSEs cannot achieve.

This project will attempt to ease the difficulties associated with the use of ISO/IEC 12207 processes and ISO 9001, and reduce the conformance obligations by providing VSE profiles. The project will develop guidance for each process profile and provide a roadmap for compliance with ISO/IEC 12207 and ISO 9001.

It has been reported that VSEs find it difficult to relate ISO/IEC 12207 to their business needs and to justify the application of the international standards in their operations. Most VSEs cannot afford the resources for, or see a net benefit in, establishing software processes as defined by current standards (e.g. ISO/IEC 12207). Liaison will be established between the proposed work and other SC7 work; specifically, the progress of ISO/IEC 12207 will be tracked.

VISION STATEMENT

This project will:

- Provide VSEs with a way to be recognized as producing quality software systems without the initial expense of implementing and maintaining an entire suite of systems and software engineering standards, or performing comprehensive assessments;

- Produce guides which are easy to understand, affordable, and usable by VSEs;

- Produce a set of profiles, which builds on or improves a VSE's existing processes, or provides guidance in establishing those processes;

- Address the market needs of VSEs by allowing domain-specific profiles and levels;

- Provide examples to encourage VSEs to adopt and follow processes that lead to quality software, matching the needs, issues, and risks of their domain;

- Provide a baseline for how multiple VSEs can work together or be assessed as a project team on projects that may be more complex than can be performed by any one VSE;

- Develop scalable profiles and guides so that compliance with ISO/IEC 12207 and/or ISO 9001:2000 and assessment become possible with a minimum of redesign of the VSE's processes.

REFERENCED DOCUMENTS

As illustrated in Figure 4-16, a number of documents have been identified as pertinent inputs to this project: ISO 90003, ISO/IEC 12207, ISO/IEC 15504, Capability Maturity Model Integration (CMMI), and the Software Capability Maturity Model (SW-CMM) developed by the SEI.

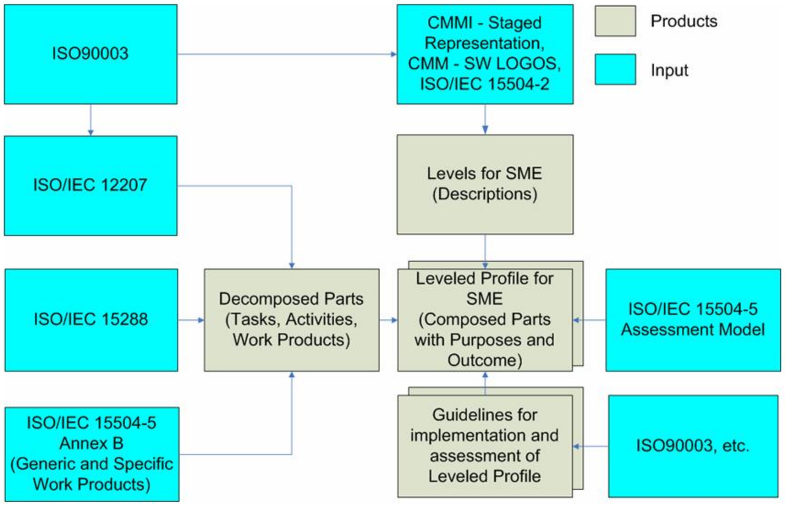

ISO90003
CMMI - Staged Representation, CMM - SW LOGOS, ISO/IEC 15504-2
Products
Input
ISO/IEC 12207
Levels for SME (Descriptions)
ISO/IEC 15288
Decomposed Parts (Tasks, Activities, Work Products)
Leveled Profile for SME (Composed Parts with Purposes and Outcome)
ISO/IEC 15504-5 Assessment Model
ISO/IEC 15504-5 Annex B (Generic and Specific Work Products)
Guidelines for implementation and assessment of Leveled Profile
ISO90003, etc.

Figure 4-16 Referenced documents (ISO 2005b)

4.4.4 Second Special Working Group meeting – Thailand, 2005

In July 2005, the TISI sent out a second invitation to participate in the SWG that was to be held in September 2005 in Bangkok. The meeting was attended by 12 participants from the following 10 countries: Australia, Belgium, Brazil, Canada, Finland, Japan, Luxembourg, South Africa, Thailand, and the United States.

The main objective of the meeting was to prepare material that would be presented to WG24 in order to facilitate the official start-up of the working group at the Italy meeting in October 2005.

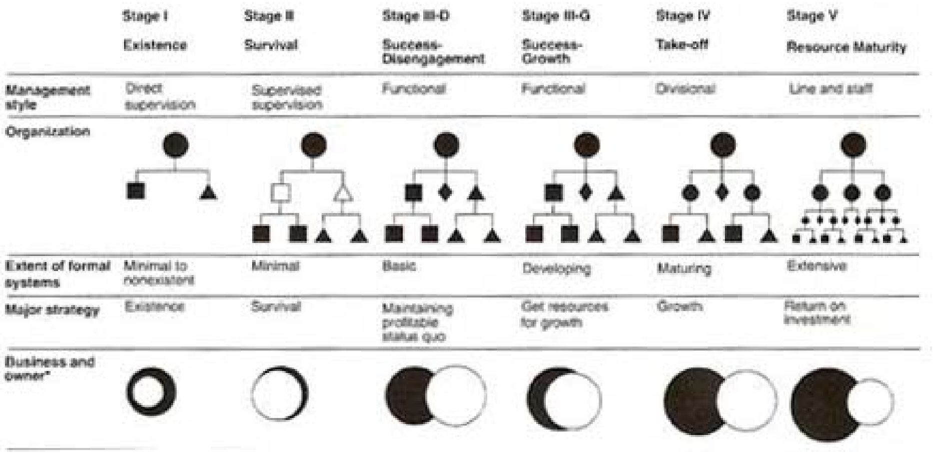
Five stages of small business growth

The author presented a paper describing the five stages of growth of a small and growing business (Churchill et al. 1983), since most standalone VSEs are by definition small and hopefully aiming to grow. The objective of the presentation was to discuss, from a management perspective, how an enterprise grows from a recently created business, or start-up, to a mature business, instead of concentrating only on the technical perspective. This presentation helped the group better understand how the profiles could be in line with the stages of growth of an enterprise. The main assumption is that small businesses (SBs) experience common problems, which arise at similar stages.

- The model developed by Churchill can aid in:
    - assessing challenges at each stage of growth

- anticipating the requirements at each stage
- evaluating the impact of proposed profiles
- diagnosing problems of each stage of growth

As illustrated in Figure 4-17, each stage of growth is characterized by the following factors: size, management style, organizational structure, extent of formal systems, major strategic goals, and degree of involvement of the owner.



	Stage I Existence	Stage II Survival	Stage III-D Success-Disengagement	Stage III-G Success-Growth	Stage IV Take-off	Stage V Resource Maturity
Management style	Direct supervision	Supervised supervision	Functional	Functional	Divisional	Line and staff
Organization						
Extent of formal systems	Minimal to nonexistent	Minimal	Basic	Developing	Maturing	Extensive
Major strategy	Existence	Survival	Maintaining profitable status quo	Get resources for growth	Growth	Return on investment
Business and owner*						

Figure 4-17 Stages of Growth (Churchill et al. 1983)

The five stages are defined as follows:

STAGE I

This stage is titled Existence, since its objective is to attract customers and deliver a product or service. The owner does everything and directly supervises the employees. The three main issues at that stage are:

1. Can we attract enough customers and deliver the product or service to become viable?
2. Is it possible to attract additional customers?
3. Can we make enough money to pay the bills?

If they cannot achieve sufficient customer acceptance or offer an interesting product or service, the business can close. If the enterprise stays in business, it moves on to stage II.

STAGE II

In this stage, titled Survival, the key issue is to generate enough cash to break even or to finance growth. The enterprise has a small number of employees supervised by a manager.

STAGE III

At this stage, titled Success, the owner can either keep the enterprise stable and profitable for long periods if the product or service niche does not allow growth over time, or he can make the decision to grow. The enterprise is large enough to have functional managers and the first professional staff.

STAGE IV

At this stage, titled Take-off, the issue is how to grow rapidly and how to finance this growth. The owner has to delegate to others and to secure enough cash to grow.

STAGE V

At this stage, titled Resource Maturity, the issue is to consolidate and control gains and to retain the advantages of small size.

Churchill lists the factors that change in importance as the business grows: financial resources (cash and borrowing capacity); personal resources (number, depth, quality); systems resources (sophistication of information, planning, and control); and business resources (customer relations, market share, supplier relations, manufacturing and distribution processes, technology, reputation). Similarly, there are factors related to the owner that change in importance as the business grows: the owner's goals for himself or for the enterprise, and the owner's operational, managerial, and strategic abilities. Churchill has defined three levels of importance: critical, important, and modestly irrelevant to the enterprise. Figure 4-18 illustrates the changes of factors at each stage of development. At stages I and II, the owner's ability to delegate is low, as is the formality of systems and controls, since the number of employees is low and the owner is making most of the decisions. Major changes occur, as illustrated in Figure 4-18, when the enterprise moves from stages I and II to stage IV. At stage IV, the importance of these factors is reversed, since the enterprise has more employees, the level of formality is higher, and the owner has to delegate.

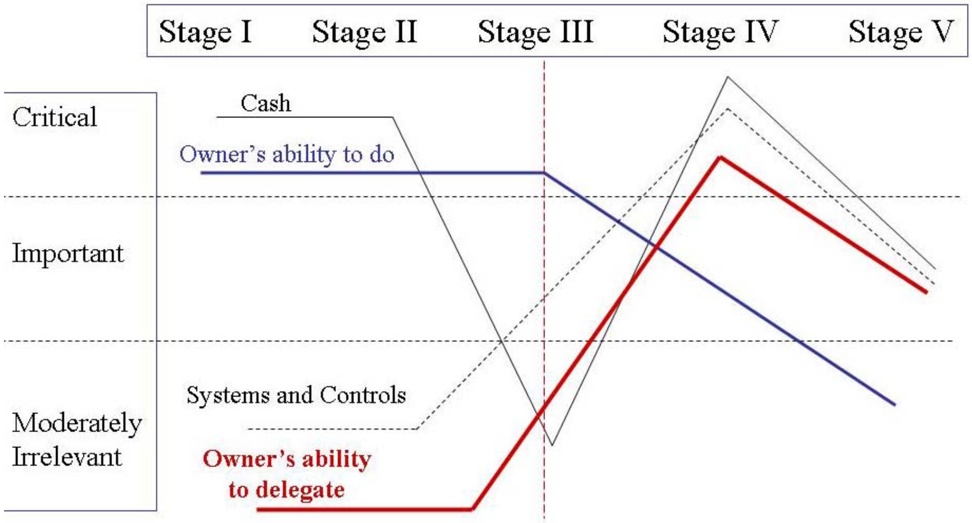


Figure 4-18 Importance of factors at each stage

Business models and the selection of appropriate software practices

The author also presented the delegates with a paper (Iberle 2002) to explain the characteristic business models of organizations developing software, such as different domains, under different levels of criticality, etc. The primary business models described by Iberle are as follows:

- Custom systems written on contract: We make money by selling our services to write other people's software.
- Custom systems written in-house: We make software to make our own business run better.
- Commercial products: We make money by writing and selling software to other businesses.
- Mass-market software: We make money by writing and selling software for consumers.
- Commercial and mass-market firmware: We make money by selling objects which happen to require embedded software.
- Open-Source: We make software for our personal satisfaction.
- Academic software: We make software for use in research – in biology, chemistry, physics, psychology, economics, and many other fields, as well as computer science.

- Internet: There are multiple business models using Internet technology now, and a single practice culture has not yet formed.

Iberle states that each business model has a set of attributes associated with it (as listed in Table 4-17). These attributes influence the choice of software engineering practices.

- **Criticality:** The potential for harming the user's or purchaser's interests varies, depending on the type of product. Some software can kill you when it fails, other software can lose large sums of many people's money, and still other software can do nothing worse than waste the user's time.
- **Uncertainty of users' wants and needs**: The requirements for software that implements a known business process (such as a tax code) are necessarily better known than the requirements for a consumer product that is so new that the end users do not even know that they want it.
- **Range of Environments:** Software written to use in a specific company needs to be compatible only with that company's officially supported computing environments, whereas software sold to the mass market has to work with a wide range of environments. The skill sets of users in the mass market also vary more widely than those of users in a specific company.
- **Cost of Fixing Errors:** Distributing firmware fixes is a great deal more expensive than patching a single website.
- **Regulation:** Regulatory agencies and contract terms can require practices that might not otherwise be adopted, or may set up a situation in which certain practices become useful. Some situations require process audits, which verify that a certain process was followed to make the product. (The term *audit* is also used in some fields for an activity which verifies that the end product works as intended, which is not the same as a process audit.)
- **Project Size:** Multi-year projects with hundreds of developers are common in some businesses, whereas shorter single-term projects are more typical in other businesses.
- **Communication:** There are a number of factors in addition to project size that can increase the amount of person-to-person communication needed, or make accurate communication more difficult. Some of the factors seem to show up more frequently in certain practice cultures, whereas others appear to be randomly distributed.
- **Organizational Culture:** The organization itself will have a culture that defines how people operate such as:
  - **Control –** "Control cultures, like IBM and GE, are motivated by the need for power and security."
  - **Competence –** "A competence culture is driven by the need for achievement: Microsoft is an obvious example."
  - **Collaboration –** "Collaboration cultures, epitomized by Hewlett-Packard, are driven by a need for affiliation."
  - **Cultivation** – "A cultivation culture motivates by self-actualization…and can be illustrated by Silicon Valley start-up companies."

Table 4-17 Attributes Associated with Business Models (adapted from Iberle 2002)

Following a discussion, the group decided to keep the information about business models and their attributes to help define the characteristics of the VSEs that will be targeted for the definition of the first profile standard. The decision was made, for the first profile, to select generic software developers and leave the development of profiles for mission-critical software developers and commercial software developers for the future. In the final chapter, the author presents profiles to address some of the business models listed in this section.

Draft Requirements for the first profile

The group decided to draft the requirements that will be used to develop the first profile (BKK-032 2005). These requirements, a few of which are listed in Figure 4-19, were reviewed and updated at subsequent meetings.

R01 - The collection should be based on the SE needs of the majority of the SMEs (market-driven).
R02 - The collection should initially focus on lower levels of capability.
R03 - The collection should be applicable to small teams or projects (small-scale software development).
R04 - The use of the collection should enable multiple SMEs to work together (teaming arrangements) or work with a prime contractor.
R05 - The guides should be available free of charge.
R06 - The ISPs should be as cheap as possible.
R07 - Implementation of the collection must be affordable (low cost of entry).
R08 - The collection should provide the whole spectrum of documents (from standards to education material).
R09 - The collection should provide upward compatibility to implementation of the full standards.
R10 - The standardized profile should clarify its purpose in relationship to other standards (e.g. 12207).
R11 - The collection should reference standards on how to express artefacts (the product itself).

Figure 4-19 Subset of draft requirements (adapted from BKK-032 2005)

Conducting a survey of needs and problems of VSEs in using standards

The author proposed to the members of the working group that a survey be conducted in order to better define the needs of VSEs regarding the use of standards, and confirm the assumptions of the members of the working group regarding the needs of VSEs. In a later chapter, the details of the survey are presented.

During the previous meetings, the delegates exchanged their points of view regarding the problems and needs of VSEs about the utilization of standards. These were main assumptions of the working group:

- A particular business context requires particular profiles (i.e. ISP).
- The situational factors have an impact on profiling.
- Organizations become more competitive through having higher-quality or more innovative products, faster product delivery, lower product cost, or better customer service.
- Improvements in any of these goals are possible through use of improved processes for product development.

- Use of the standard ISO/IEC 12207 to improve processes can be applied by an enabler to achieve any of these goals through better management of time and resources, and the elimination of the waste associated with either.
- VSEs are limited in both time and resources, which leads to lack of understanding of how to use the standard for their benefit.

The other main outputs of the meeting were:

- approaches to profiling development and architecture;
- business models on how organizations profit from software (Iberle 2002), such as custom systems written on contract, custom systems written in-house, commercial products (mass-market), and consumer software;
- agenda for the first official WG24 meeting;
- draft strategic plan for WG24.

4.4.5 First official Working Group meetings – Italy, October 2005[44]

In October 2005 in Italy, WG24 held its first official working session. Sixteen delegates from 7 countries (Canada, Finland, Ireland, Luxembourg, South Africa, and Thailand) and 2 liaison organizations (IEEE, INCOSE) attended the meeting. Each participant was asked to introduce himself, and, if applicable, present projects or initiatives related to the scope of the new working group:

- Belgium presented the Micro Evaluation method developed by the CETIC and the Université de Namur.
- Thailand presented its Thailand Quality Standard (this topic was presented in a previous chapter)
- Ireland presented the Irish initiatives and introduced the Spire Handbook as an example of packaging standards for consumption by VSEs.
- The SEI initiative for small settings was presented, as well as work in progress in Finland.
- The following topics were covered during the meeting:
  - Project requirements were finalized to constitute the project baseline.

[44] This section is adapted from the minutes of the meeting in Bari (Italy) (BAR-003 2003)

  - The comments received during the balloting of the New Work Item (NWI) were processed.
  - The profile creation strategy was defined.
  - Appropriate situational factors and business models from the Iberle paper (Iberle 2002) were identified.
  - The survey questionnaire to validate project requirements and to collect information from VSEs was reviewed and finalized (the survey is presented in a later chapter).
- The group also discussed the process that will be used to develop the future IS and TR (see Figure 4-20).

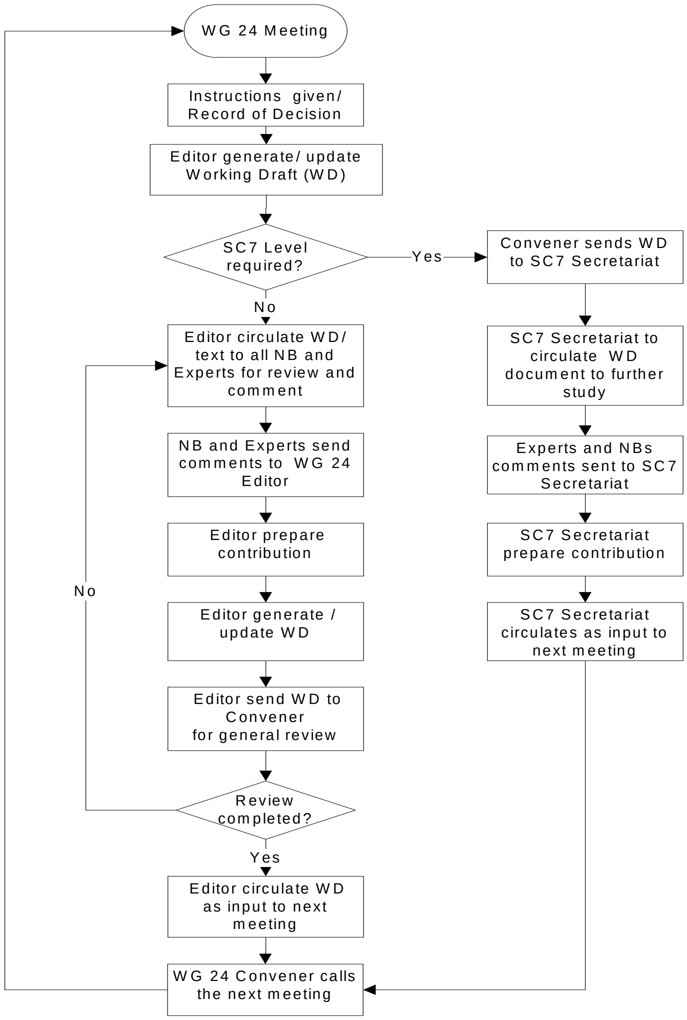

WG 24 Meeting
Instructions given/ Record of Decision
Editor generate/ update Working Draft (WD)
SC7 Level required?
Yes
No
Convener sends WD to SC7 Secretariat
SC7 Secretariat to circulate WD document to further study
Experts and NBs comments sent to SC7 Secretariat
SC7 Secretariat prepare contribution
SC7 Secretariat circulates as input to next meeting
Editor circulate WD/ text to all NB and Experts for review and comment
NB and Experts send comments to WG 24 Editor
Editor prepare contribution
Editor generate / update WD
Editor send WD to Convener for general review
No
Review completed?
Yes
Editor circulate WD as input to next meeting
WG 24 Convener calls the next meeting

Figure 4-20 Operation and document flow of WG24

A 2-phase strategy to develop profiles was discussed. The objective of phase 1 was to create rules to select, from ISO 12207 (see Figure 4-21), core processes (or base practices and work products) based on business models and situational factors (Iberle 2002) for a 15504 Level 1 capability. This objective would be reached by following this proposed approach:

- Set business objectives
- Consider success factors related to business objectives
- Identify processes to include in the profiles

It was proposed that profiles be extended in phase 2 by identifying additional processes.

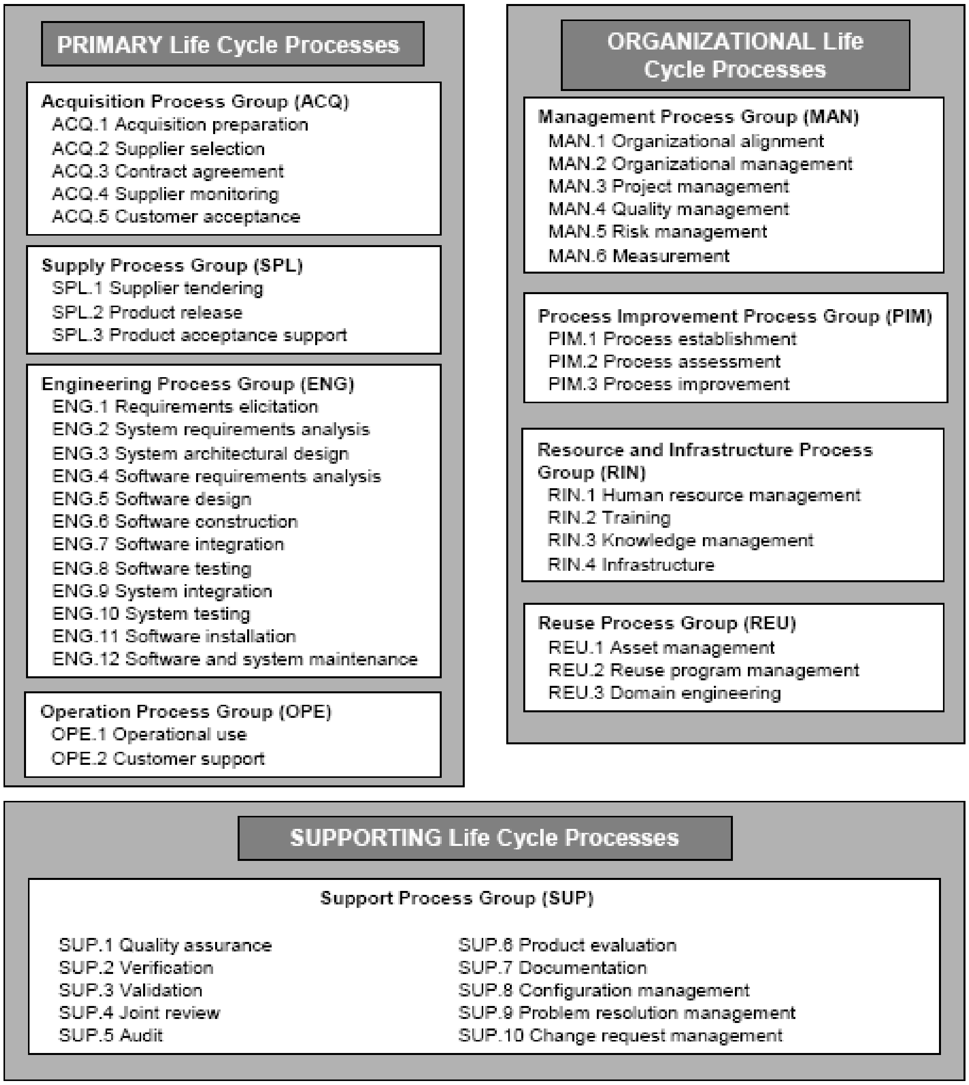

PRIMARY Life Cycle Processes
Acquisition Process Group (ACQ)
ACQ.1 Acquisition preparation
ACQ.2 Supplier selection
ACQ.3 Contract agreement
ACQ.4 Supplier monitoring
ACQ.5 Customer acceptance
Supply Process Group (SPL)
SPL.1 Supplier tendering
SPL.2 Product release
SPL.3 Product acceptance support
Engineering Process Group (ENG)
ENG.1 Requirements elicitation
ENG.2 System requirements analysis
ENG.3 System architectural design
ENG.4 Software requirements analysis
ENG.5 Software design
ENG.6 Software construction
ENG.7 Software integration
ENG.8 Software testing
ENG.9 System integration
ENG.10 System testing
ENG.11 Software installation
ENG.12 Software and system maintenance
Operation Process Group (OPE)
OPE.1 Operational use
OPE.2 Customer support
ORGANIZATIONAL Life Cycle Processes
Management Process Group (MAN)
MAN.1 Organizational alignment
MAN.2 Organizational management
MAN.3 Project management
MAN.4 Quality management
MAN.5 Risk management
MAN.6 Measurement
Process Improvement Process Group (PIM)
PIM.1 Process establishment
PIM.2 Process assessment
PIM.3 Process improvement
Resource and Infrastructure Process Group (RIN)
RIN.1 Human resource management
RIN.2 Training
RIN.3 Knowledge management
RIN.4 Infrastructure
Reuse Process Group (REU)
REU.1 Asset management
REU.2 Reuse program management
REU.3 Domain engineering
SUPPORTING Life Cycle Processes
Support Process Group (SUP)
SUP.1 Quality assurance
SUP.2 Verification
SUP.3 Validation
SUP.4 Joint review
SUP.5 Audit
SUP.6 Product evaluation
SUP.7 Documentation
SUP.8 Configuration management
SUP.9 Problem resolution management
SUP.10 Change request management

Figure 4-21 ISO 12207 Process Categories and Process Groups (ISO 1995)

- The group discussed the proposed architecture of documents. It was decided that the following documents be developed:
  - a TR (Technical Report) Framework and Overview
  - an ISP Profile Taxonomy
  - an ISP Profile 1 (as an interim name for the first profile)
  - a TR Assessment Guide
  - a TR Management Guide

Table 4-18 illustrates a subset of the profile matrix, for a subset of the engineering process group and reuse process group, for the following business models:

- Custom systems written on contract: We make money by selling our services to write other people's software.
- Custom systems written in-house: We make software to make our own business run better.
- Commercial products: We make money by writing and selling software to other businesses.
- Mass-market software: We make money by writing and selling software to consumers.
- Commercial and mass-market firmware: We make money by selling objects which happen to require embedded software.

An X in a column indicates that this process has been selected for a particular business model.

12207 Process Group	**12207 Process Name**	**Custom systems written on contract**	**Custom systems written in-house**	**Commercial products**	**Mass-market software**	**Commercial and mass-market firmware**
Engineering Process Group	System requirements analysis	X	X	X	X	X
Reuse Process group	Asset management	X	X			X
Reuse Process group	Reuse program management	X	X	X	X	X

Table 4-18 Example of a Profile Matrix

Canada offered to establish, using the facilities of the Standards Council of Canada, a password-protected electronic repository for all documents produced by the working group[45]. The documents stored on this site were to be accessible by delegates using an ID and password issued by the secretary of the working group. The main area for posting documents worked as follows:

- Folders are defined for various groups of documents (Administration, Meeting, Projects, etc.).
- Documents (or a URL, and/or attached files) are posted by submitters and members are notified and sent a URL.

[45] https://int.scc.ca/forums/int/dispatch.cgi/JTC1-SC7-WG24

- Comments are posted by their author and sequenced in threads, and members are notified.

This process subsumes any direct circulation to members by e-mail and, although a send-email facility exists on the forum, this is only for informal communication.

Finally, the editor and the secretary developed the ISP outline and invited the participants to provide text, from the 12207, 15504, and 15289 standards, by March 14. The participants were asked to download these texts after March 14, review them, and provide comments to be discussed at the May 2006 meeting.

4.4.6 Working Group meetings in 2006

In 2006, the group held two meetings, one during the SC7 Plenary meeting in Thailand and an interim meeting in Luxembourg.

The meeting in Thailand in May 2006[46]

Twenty-one delegates from 10 countries (Belgium, Canada, Finland, Korea, Luxembourg, Mexico, New Zealand, India, Ireland, South Africa, and Thailand), 2 liaison organizations (IEEE, INCOSE) and 18 observers from Thailand attended the second working group meeting. Three new countries had joined the working group since the last meeting: Korea, Mexico, and India.

Following a review of the action items from the minutes of the last meeting, the main topics discussed were as follows:

- The author introduced the results of the survey. Discussions were held on the survey in multiple sessions during the week. It was decided that a second round of the survey be conducted in order to obtain more data from some countries (the survey is presented in more detail in a later chapter).

- Mexico gave a presentation on the new Mexican standard MoProSoft (NMX 2005) targeted at VSEs. The members of the working group considered that this contribution could serve as the basis for the first working draft. However, it was felt that the Mexican standard was probably targeted toward a higher segment of the market (in terms of the size of the enterprise and the resources available), and might not be in line with our stated requirements. Whether we use this material as our first profile, or as input to the preparation of a scaled-down version, the first step is the same. The decision on further processing will be taken after the next interim meeting.

[46] Adapted from the minutes of the meeting (BK3-003 2006)

- Since the Mexican standard was written in Spanish, the Mexican delegation agreed to provide an English version for the next meeting.
- The group reviewed the texts received from the members of the following documents:
    - Framework and Overview
    - ISP Profile Taxonomy
    - ISP Profile 1
    - TR (Technical report) Assessment Guide
    - TR Management Guide
    - TR Engineering Guide

A new version of the documents was produced. In a later chapter, the set of documents developed by WG24 is described in more detail.

The meeting in Luxembourg[47]

Fifteen delegates from 7 countries (Australia, Canada, Finland, Ireland, Luxembourg, Mexico, South Africa, and Thailand) and 2 liaison organizations (IEEE, INCOSE) attended the third working group meeting.

The main objectives for the meeting were to identify, from the Mexican contribution, what elements to import into the future ISO standards and guides. The Mexican set of documents was found to be consistent, and there was a danger of breaking that consistency if the WG carved parts of it out to produce a lighter process.

Discussions have resulted in identifying the need for a profile for VSEs having 0 to 9 employees and a profile for VSEs with 10 to 25 employees. Whether these are two separate profiles or a single graduated profile was not decided at the meeting.

The author reported on the results of the survey as of September 22, 2006. Low participation in some major countries (e.g. United States, Germany) was discussed and deplored. Ways to correct this situation were investigated.

The author introduced what he had developed as a public Web site to publicize the work of the working group. There was general support for the idea. In a later chapter, the author will illustrate, with data, that the site has been visited by thousands of people.

[47] Adapted from the minutes of the meeting (LUX-003 2006)

The group processed texts received from the members since the last meeting, and participants volunteered to provide additional texts for the following documents:

- ISO/IEC 299110-1 Framework and Overview
- ISO/IEC 299110-2 ISP Profile Taxonomy
- ISO/IEC 299110-3.1 ISP Profile 1
- ISO/IEC 299110-4 TR Assessment Guide
- ISO/IEC 299110-5 TR Management and Engineering Guide

The author presented the Web site he had developed to make the work in progress and the papers published about the work of WG24 visible outside the working group.

4.4.7 Working Group meetings in 2007

In 2007, the group held two meetings, one during the SC7 Plenary meeting in Russia and an interim meeting in Canada.

The meeting in Russia[48]

The meeting was held in May 2007 in Moscow. Eighteen delegates from 10 countries (Australia, Belgium, Canada, Finland, India, Japan, Mexico, Luxembourg, South Africa, and Thailand) and 2 liaison organizations (IEEE, INCOSE) attended the fourth working group meeting.

Contributions from delegates were presented and reviewed. Since the documents were moving closer to an ISO balloting, it was decided that volunteer editors be sought for the set of documents:

- ISO/IEC TR29110-1 SELP4VSE Overview
- ISO/IEC ISP29110-2 SELP4VSE Framework and Profile Taxonomy
- ISO/IEC TR29110-3 SELP4VSE Assessment Guide 1
- ISO/IEC ISP29110-4.1 SELP4VSE Profile 1 Specification
- ISO/IEC TR29110-5.1 SELP4VSE Profile 1 Management and Engineering Guide

Since the working group will be receiving official comments from NBs for its next meeting, the secretary provided a set of rules to co-editors, as follows:

- Editors will use the ISO templates for the development of ISP and TR.

[48] Adapted from the minutes of the meeting (MOS-003 2007)

- Editors were authorized to include in documents material from other ISO documents, and material that had already been submitted to WG24, with an editor's note.
- Editors will implement the latest version of the document content, as discussed in the overview document.
- Editors will complete the introductory sections, but not necessarily the terms and definitions at this stage.
- Editors will provide Working Draft (WD) 1 documents to the WG24 Secretariat by July 5, 2007.

Participants were instructed to provide comments by September 15 to allow editors to pre-process them before the next meeting. The preprocessing of comments is a process whereby a document editor reviews, the comments received from the NBs and prepares a disposition (e.g. rejected, accepted, accepted in principle) for each comment prior to the next meeting. Preprocessing comments should accelerate the disposition of comments in front of all members of the working group. Then, during the official meeting, the editor presents each comment and the proposed disposition for discussion.

At the Moscow meeting, the author presented the concept of the deployment package (DP) to all participants. The assumption was that VSEs will still have difficulty reading and applying the proposed processes of the Engineering and Management Guide (these documents will be explained in detail in the next chapter). The author proposed the development of DPs so that any VSE can put practices in place to solve a problem in a short period of time. For example, many VSEs do not perform any basic configuration control activities and often suffer from the resulting situation (e.g. inability to find the latest version of the code, code that is "crushed" (i.e. overridden) by an older version). Now, a DP could be developed to cover basic version control activities, which would be made available to VSEs at no charge on Web sites. A table of contents was also proposed, as an Appendix to the Engineering and Management Guide, and discussed (see Table 4-19). DPs will be explained in detail in a later chapter.

- Title of Package
- Technical description
  - Link to the ISO framework (e.g. ISP)
  - Activity description
    - Text and graph
  - Reference to ISO standards
- Competencies required
  - Knowledge and skills
- Template(s)
  - Empty
  - Example(s)
- Checklist(s)
  - To facilitate implementation, assessment, and self-assessment
- Identification of software tool(s)
  - Application guide (i.e. User guide, installation guide)
- Training material
- Deployment package evaluation form

Table 4-19 Proposed Table of Contents of Future Deployment Packages

It was announced that a draft of the first deployment package, Requirements Analysis, would be developed jointly by Belgium and Canada, and would be presented to the delegates at the next meeting in Montreal.

The group decided to instruct the Secretariat of JTC1/SC7 to issue the following documents for a combined Working Draft (WD) Circulation and PDTR/PDISP Registration ballot, upon recommendation of the working group after the Montreal meeting, if the documents meet the requirements for a PDTR/PDISP Registration ballot and when they are received by the Secretariat.

- TR 29110-1WD1SELP4VSE Overview
- ISP 29110-2WD1SELP4VSE Framework and Profile Taxonomy
- TR 29110-3WD1SELP4VSE Assessment Guide
- ISP 29110-4WD1SELP4VSE Profile 1 Specification
- TR 29110-5WD1SELP4VSE Profile 1 Management and Engineering Guide

It was also decided that, in order to be able to finalize the set of documents, the PDTR/PDISP Registration time frame would be extended by 18 months (from September 2006 to May 2008), and the DTR/FDISP Ballot time frame by 12 months (from May 2009 to May 2010).

The meeting in Canada[49]

The fifth meeting of WG24 was held in Montreal, Canada, in October 2007. Fourteen delegates from 8 countries (Belgium, Canada, Columbia, Finland, Ireland, Luxembourg, Mexico, and Thailand) and 2 liaison organizations (IEEE, INCOSE) attended the fifth working group meeting.

Comments were processed and the disposition noted, and contributions from delegates were presented and reviewed. Delegates volunteered as co-editors of the ISP and TR as follows:

- TR 29110-1 VSE Overview (Spain)
- ISP 29110-2 VSE Framework and Profile Taxonomy (Belgium and Canada)
- TR 29110-3 VSE Assessment Guide (Finland)
- ISP 29110-4 VSE Profile 1 Specification (Canada)
- TR 29110-5 VSE Profile 1 Management and Engineering Guide (Mexico)

The draft of the first deployment package, Requirements Analysis, developed jointly by Belgium and Canada, was presented to the delegates. The delegates agreed that deployment packages would help VSEs in their countries in applying the processes mandated in the ISP. It was decided that Profile 1 would be renamed *Basic Profile*. Delegates volunteered as authors of deployment packages as follows:

- Testing Verification & Validation (Columbia)
- Configuration Management (Thailand)
- Project Management (Ireland and Thailand)

The author of this thesis announced that he would act as editor of the all deployment packages.

4.4.8 Working Group meetings in 2008

In 2008, the working group met twice: in May in Berlin, and in November in Mexico.

The meeting in Germany[50]

Fifteen delegates from 10 countries attended this meeting in Berlin. It was the first meeting where comments received from ballots were to be processed. It was decided that co-editors

[49] Adapted from the minutes of the meeting (MON-003 2007)

[50] Adapted from the minutes of the meeting (BER-003 2008)

would be formally assigned to lead the disposition of comments. The co-editors nominated were:

- International Council on Systems Engineering (INCOSE) for TR 29110-1 Overview
- Ireland and Belgium for ISP 29110-2 Framework and Profile
- Finland for TR 29110-3 Assessment Guide
- Mexico for ISP 29110-4 Basic Profile Specification
- Mexico and US for TR 29110-5 Basic Profile Management and Engineering Guide

Over 237 comments were processed by the delegates during this meeting (see Table 4-20), and two deployment packages were reviewed: configuration management and testing. The delegates decided to reduce the scope of both packages to bring them into line with the Basic Profile. The scope of the Configuration Management deployment package was reduced to version control, and the scope of the Testing deployment package was reduced to unit testing. It was also proposed that the deployment packages would be offered at no charge and a suitable copyright statement was agreed to.

Title of document	No. of comments
TR 29110-1 Overview	71
ISP 29110-2 Framework and Profile Taxonomy	33
TR 29110-3 Assessment Guide	18
ISP 29110-4 Basic Profile Specification	52
TR 29110-5 Basic Profile Management and Engineering Guide	63

Table 4-20 Comments Processed at the Berlin Meeting

The meeting in Mexico[51]

Thirty-two delegates from 11 countries attended the fall 2008 meeting in Mexico, and over 455 comments were processed (see Table 4-21). Canada suggested a modification to the name *Very Small Enterprises*, since it did not reflect the total mandate of the working group. The mandate in fact includes small software systems development departments and projects within larger organizations. It was decided to change the name to *Very Small Entities*, thus preserving the acronym VSE used since the beginning of the project.

[51] Adapted from the minutes of the meeting (MEX-003 2008)

Title of document	No. of comments
TR 29110-1 Overview	61
ISP 29110-2 Framework and Profile Taxonomy	94
TR 29110-3 Assessment Guide	38
ISP 29110-4 Basic Profile Specification	54
TR 29110-5 Basic Profile Management and Engineering Guide	208

Table 4-21 Comments Processed at the Mexico Meeting

The plan for the development of international standards for VSEs is as follows: the working group has decided to focus its activities on entities that develop generic software, i.e. entities that do not develop commercial products or mission-critical software. The size of VSEs, i.e. fewer than 25 people, fits well with the sizes of enterprises described in the model developed by Churchill.

The main question for the working group was this: how many sub profiles do we need to develop to bring VSEs to stage V? Initially, the group developed what is now called the *Basic* sub profile. At the Berlin meeting, the Belgian delegate and the author had proposed the development of a "lighter" sub profile for VSEs that have just started up, or for projects with about 6 person-months of effort expended. This proposed sub profile presented at the meeting was called the *Entry* sub profile. The skeleton of this sub profile is described in Figure 4-22.

Project Planning and Monitoring Process:
Develop an agreement with the customer
Develop a plan
Determine phases, tasks, milestones, deliverables
Assess available resources, estimate effort
Monitor project status and perform reviews (e.g. retrospective)
Collect data (e.g. effort expended on tasks)
Software Development Process
Requirement Analysis and Design
Identify the set of requirements to be implemented
Plan interactions with the customer
Design the software
Software Code and Test
Code and debug
Perform unit and system testing

Figure 4-22 High-level description of the Entry Sub profile

Using the MoProsoft framework, Mexico presented the characteristics of two sub profiles that would be 'above' the Basic sub profile. These sub profiles were temporarily titled *Intermediary* and *Advanced* sub profiles. It was decided that the proposed profiles would be presented and discussed at the next meeting in India.

The co-editors were asked to finalize the corrections to the document and upload them to the WG24 site by December 20, so that the secretary could submit the revised documents as well as the final disposition of comments to the secretariat of SC7 for the next round of document balloting.

4.4.9 Working Group meetings in 2009

The group held one meeting, in India in May 2009. A second meeting is scheduled in November in Peru.

The meeting in India[52]

Twenty-one delegates from 11 countries attended the 2009 meeting in India at Hyderabad, and over 259 comments were processed (see Table 4-22).

Title of document	No. of comments
TR 29110-1 Overview	60
ISP 29110-2 Framework and Profile Taxonomy	52
TR 29110-3 Assessment Guide	40
ISP 29110-4 Basic Profile Specification	54
TR 29110-5 Basic Profile Management and Engineering Guide	53

Table 4-22 Comments Processed at the India Meeting

A substantial reduction in the number of comments, i.e. from 455 to 259, indicates that the documents are moving to a higher level of maturity. Once they are updated, they will be sent for another round of comments. At the next meeting, in Peru in November 2009, the delegates will decide, based on the comments received, whether or not the documents will be ready for submission for a final vote by all participating ISO countries.

4.5 Conclusion

In this chapter, the International Organization for Standardization and the development process of engineering standards of that organization, as well as the accomplishments of WG24, have been described. Technology transfer was also discussed to illustrate the importance of this topic for the widespread adoption of the standards by VSEs. In the next chapter, its contributions to the development and conduct of an international survey of VSEs are present, along with the survey results.

[52] Adapted from the minutes of the meeting (HYD-003 2009)

This page is intentionally left blank

Chapter 5

Contributions to the International Survey of VSEs

5 Chapter 5 – Contributions to the International Survey of VSEs

5.1 Introduction[53]

This chapter covers phase 2 of the 6-phase innovation process illustrated in Figure 5-1, titled Basic and Applied Research. This phase began at the first Special Working Group meeting in Thailand in 2005, when the author proposed that an international survey of VSEs be conducted. One objective of the survey was to better understand the problems VSEs have with their use of standards. Another was to elicit the needs of VSEs before starting to develop future standards, instead of developing them from our own points of view and with our own biases about what we think would be good for VSEs. It was the first time that an ISO/IEC JT1 SC7 working group had surveyed the potential users of a future standard. Usually, standards are developed by experts, balloted, and then published by the ISO.

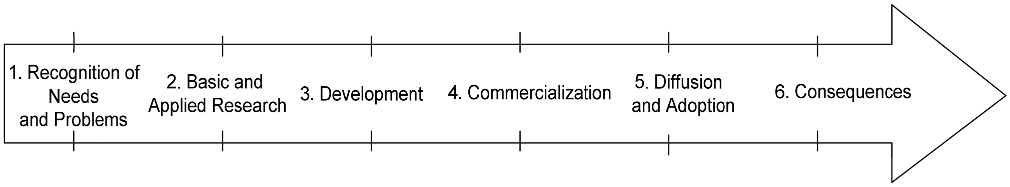

1. Recognition of Needs and Problems
2. Basic and Applied Research
3. Development
4. Commercialization
5. Diffusion and Adoption
6. Consequences

Figure 5-1 Phases of the Innovation Process (adapted from Rogers 2003)

In this chapter, the author's contribution to developing and conducting an international survey of VSEs is presented. In the first section, a survey conducted by the IEEE in 1997 to capture information about the utilization of standards is presented, followed by the survey conducted by ISO WG24 and the results of that survey.

5.2 Survey conducted by the IEEE

In 1997, the Technical Council on Software Engineering responsible for IEEE Software Engineering Standards (SES) conducted a survey to capture information from software engineering standards users in order to improve those standards (Land 1997). They gathered 148 answers, mainly from the United States (79%) and large companies (87% of them having more than 100 employees). The main application domains of the survey respondents were IT (22%), military (15%), and aerospace (11%).

Even though the IEEE survey objectives differ from those of the ISO/IEC survey, there are some interesting common findings. In response to the question concerning the reasons

[53] This chapter is adapted from published articles (Laporte et al. 2006d, Laporte et al. 2006e, Laporte 2007a, Laporte et al. 2008a, Laporte et al. 2008c)

why their organization does not use standards, 37% said that the standards were not available at their facilities, while 37% explained that they use other standards. In fact, the IEEE survey underscores the fact that ISO/IEC standards are often used in organizations, rather than IEEE standards.

The IEEE survey underlined the difficulties regarding IEEE standards use reported by the respondents. The two main difficulties were a lack of understanding of the benefits (28%) and a lack of useful examples (25%). The survey also revealed how IEEE standards are used in organizations. Most of the them (35 responses) claimed to use IEEE standards for internal plan elaboration.

The IEEE survey gathered several new requirements about the IEEE standards being requested by the respondents. These were principally examples and templates of deliverables (about 32 responses), support for metrics and measurement (about 30 responses), help on life cycle process definition (about 23 responses), a training course and support for small, rapid application development efforts.

5.3 Survey conducted by WG24

At the first meeting of a Special Working Group (SWG) in Bangkok in March 2005, the author suggested to the participants that a worldwide survey of VSEs be conducted (BK1-006 2005).

As mentioned above, one objective of the survey would be to elicit the requirements of VSEs before beginning development of future standards, instead of developing a standard that we think would be good for VSEs, from our own points of view and with our own biases. It was the first time that an SC7 working group had conducted a survey of potential users of a future standard. Usually, standards are developed by experts, balloted, and then published by the ISO.

The author proposed to the members of the working group that this survey be similar to the IEEE Software and Systems Engineering Standards Committee survey performed in 1997 (Land 1997) and seek answers to the following questions:

- Why are very small organizations not using ISO JTC1/SC7 standards?
- Which software standards are being used?
- What do users view as implementation difficulties?
- What is the level of satisfaction of standards users?
- What are the user's requirements?

- Who are the users of software standards?

WG24 decided that the survey would be developed to question VSEs about their use of standards and widely recognized documents or models, such as the CMMI, and to collect data to identify problems and potential solutions to help them apply the standards and become more competitive. At the very beginning, the working group drew up several working hypotheses regarding VSEs. The survey was intended to validate some of these, such as the following:

- The context of VSEs requires light and well focused life cycle profiles.
- Particular business contexts require particular profiles.
- There are differences, in terms of available resources and infrastructure, between a VSE employing 1 to 25 people and an IT department of the same size in a large company.
- VSEs are limited in both time and resources, which can lead to a lack of understanding of how to use the standards for their benefit.
- Benefits for VSEs may include recognition (e.g. certification) through assessment or audit by an accredited body.

The working group also wanted to know:

- the reasons for using standards, or for not using them,
- which standards were being used,
- the problems/barriers encountered when using them, and
- how to facilitate their adoption and utilization.

A schedule to conduct the survey and perform the analysis was proposed to the participants (BK1-006 2005). According to this schedule, the collection of data was to start in late 2005.

5.3.1 Development of the survey

A set of specifications for the survey was developed by the members of the working group. A project was given to a software engineering graduate student, the author of this thesis, (Bluteau 2007) with the following high-level objectives:

- Develop a preliminary set of questions to address the specifications of the WG. The working group would review and finalize the set of questions.

- Develop a Web site to host the survey which would be able to offer the questionnaire in something like 7 to 10 languages.
- Collect the data on monthly basis.
- Perform an analysis of the data.

Based on the requirements, the graduate student prepared a preliminary set of open-ended questions (Bluteau 2005), i.e. questions allowing respondents to answer, in their own words, closed questions, as well as hybrid questions where choices are provided and where respondents can also create a response.

At the second meeting of the Special Working Group in September 2005 in Bangkok, the author led a session to finalize the set of survey questions (BK1-014 2005). During that session, the following actions were taken, as suggested by (Kasunic 2005):

- identification and removal of ambiguities,
- verification that the order of the questions was logical,
- verification that it was easy to navigate through the questionnaire,
- verification that the choice of answers provided was adequate.

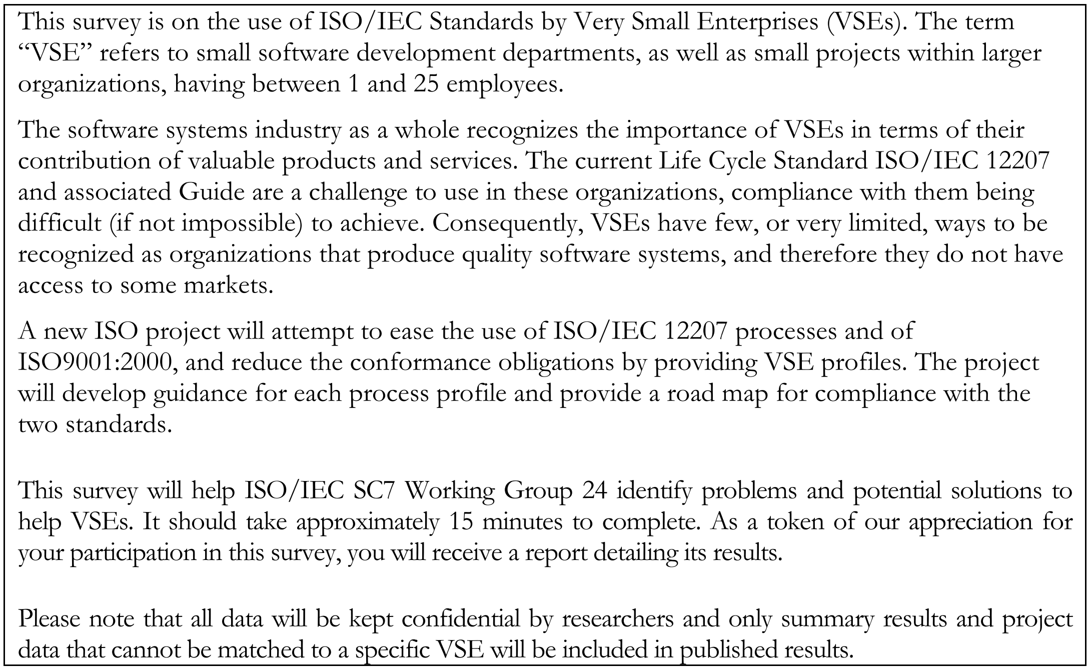
Finally, to make the survey more readable, the decision was made to add a glossary of acronyms used, including the complete names of the standards, instead of providing only their short identification; for example, ISO/IEC 15288 System Life Cycle Processes, and not just ISO/IEC 15288. Table 5-1 lists the acronyms defined in the glossary.

CMMI	Capability Maturity Model Integration
EIA/IS	Electronic Industry Alliance/Information System
FAA-iCMM	Federal Aviation Administration's Integrated Capability Maturity Model
INCOSE	International Council on Systems Engineering
IPD-CMM	Integrated Product Development Capability Maturity Model
ISMS	Information Security Management System
ISO	International Organization for Standardization
ISO/IEC 12207	Software Life Cycle Processes
ISO/IEC 15288	System Life Cycle Processes
ISO/IEC 15504	Software Process Assessment
ISO/IEC 15939	Software Measurement Processes
IT	Information Technology
ITIL	IT Infrastructure Library
MBNQA	Malcolm Baldrige National Quality Award
NGO	Non-Governmental Organization
P-CMM	People Capability Maturity Model
PSP	Personal Software Process
SA-CMM	Software acquisition Capability Maturity Model
SECAM	Systems Engineering Capability Assessment Model
SECM	Systems Engineering Capability Model
SE-CMM	Systems Engineering Capability Maturity Model
SEI	Software Engineering Institute
SPICE	Software Process Improvement Capability Determination
SPIN	Software Process Improvement Network
SW-CMM	Capability Maturity Model for Software
TSP	Team Software Process
VSE	Very Small Enterprise

Table 5-1 Glossary

An introductory text (see Figure 5-2) was developed and translated. The author was successful in finding software professionals who could translate the survey and its introduction from the original English version (BAR-026-2005a) into 8 languages:

- French (BAR-026-2005b),
- German (BAR-026-2005c),
- Thai (BAR-026-2005d),
- Turkish (BAR-026-2005e),
- Russian (BAR-026-2005f),
- Spanish (BAR-026-2005g),
- Portuguese (BAR-026-2005h).
- Korean (BAR-026-2005i),



This survey is on the use of ISO/IEC Standards by Very Small Enterprises (VSEs). The term "VSE" refers to small software development departments, as well as small projects within larger organizations, having between 1 and 25 employees.

The software systems industry as a whole recognizes the importance of VSEs in terms of their contribution of valuable products and services. The current Life Cycle Standard ISO/IEC 12207 and associated Guide are a challenge to use in these organizations, compliance with them being difficult (if not impossible) to achieve. Consequently, VSEs have few, or very limited, ways to be recognized as organizations that produce quality software systems, and therefore they do not have access to some markets.

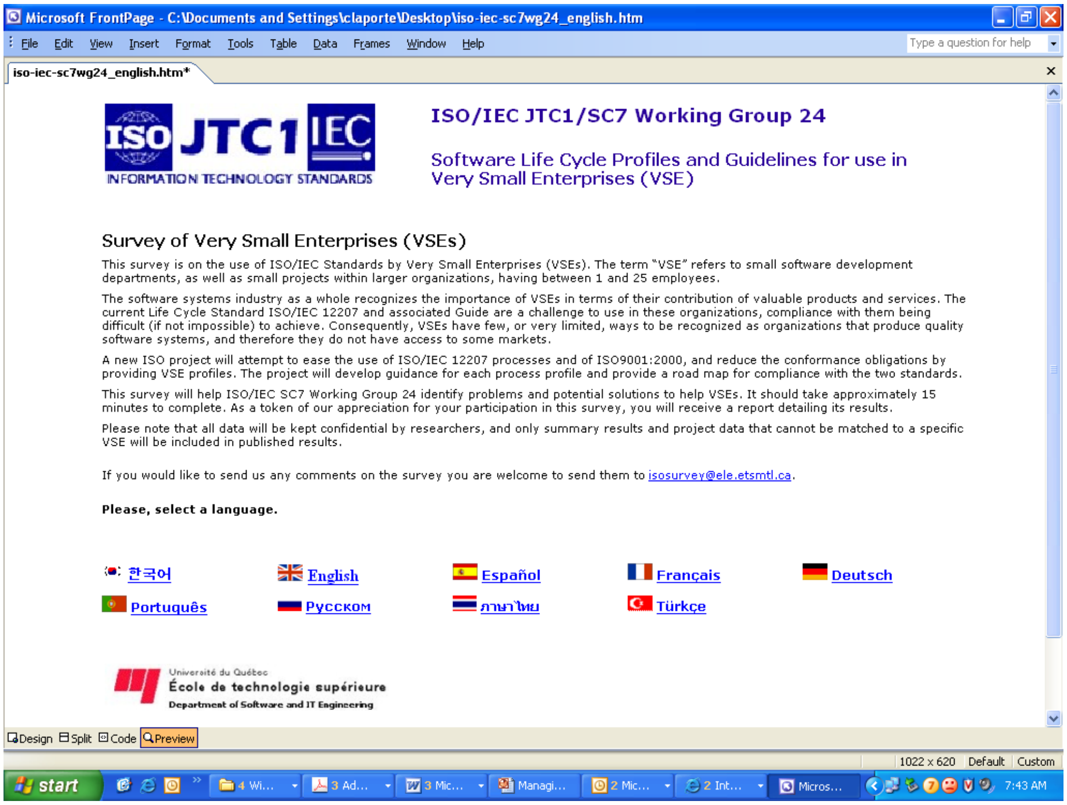
A new ISO project will attempt to ease the use of ISO/IEC 12207 processes and of ISO9001:2000, and reduce the conformance obligations by providing VSE profiles. The project will develop guidance for each process profile and provide a road map for compliance with the two standards.

This survey will help ISO/IEC SC7 Working Group 24 identify problems and potential solutions to help VSEs. It should take approximately 15 minutes to complete. As a token of our appreciation for your participation in this survey, you will receive a report detailing its results.

Please note that all data will be kept confidential by researchers and only summary results and project data that cannot be matched to a specific VSE will be included in published results.

Figure 5-2 Introduction to the survey (BAR-026 2005a)

Once the questions were finalized, the graduate student developed a Web site to automate the collection of the survey data. The Web site was hosted by the ÉTS and designed to maximize the number of responses, as well as facilitate data collection and analysis. Figure 5-3 illustrates the home page of the survey Web site.


Microsoft FrontPage - C:\Documents and Settings\claporte\Desktop\iso-iec-sc7wg24_english.htm
File Edit View Insert Format Tools Table Data Frames Window Help
Type a question for help
iso-iec-sc7wg24_english.htm*
ISO JTC1 IEC
INFORMATION TECHNOLOGY STANDARDS
Design Split Code Preview
1022 x 620 Default Custom
start
7:43 AM

ISO/IEC JTC1/SC7 Working Group 24

Software Life Cycle Profiles and Guidelines for use in Very Small Enterprises (VSE)

Survey of Very Small Enterprises (VSEs)

This survey is on the use of ISO/IEC Standards by Very Small Enterprises (VSEs). The term "VSE" refers to small software development departments, as well as small projects within larger organizations, having between 1 and 25 employees.

The software systems industry as a whole recognizes the importance of VSEs in terms of their contribution of valuable products and services. The current Life Cycle Standard ISO/IEC 12207 and associated Guide are a challenge to use in these organizations, compliance with them being difficult (if not impossible) to achieve. Consequently, VSEs have few, or very limited, ways to be recognized as organizations that produce quality software systems, and therefore they do not have access to some markets.

A new ISO project will attempt to ease the use of ISO/IEC 12207 processes and of ISO9001:2000, and reduce the conformance obligations by providing VSE profiles. The project will develop guidance for each process profile and provide a road map for compliance with the two standards.

This survey will help ISO/IEC SC7 Working Group 24 identify problems and potential solutions to help VSEs. It should take approximately 15 minutes to complete. As a token of our appreciation for your participation in this survey, you will receive a report detailing its results.

Please note that all data will be kept confidential by researchers, and only summary results and project data that cannot be matched to a specific VSE will be included in published results.

If you would like to send us any comments on the survey you are welcome to send them to isosurvey@ele.etsmtl.ca.

Please, select a language.

한국어 English Español Français Deutsch
Português Русском ภาษาไทย Türkçe

Université du Québec
École de technologie supérieure
Department of Software and IT Engineering

Figure 5-3 Home page of the survey Web site

Respondents were informed that it would take a maximum of 15 minutes to complete the survey. They were also informed that all data would be kept confidential, and that only summary results and project data that could not be matched to a specific VSE would be included in the published results. In order to increase participation in the survey, WG24 promised to send all respondents a report presenting the survey results on an anonymous basis.

Access to the Web-based survey was protected, as suggested by Kasunic (2005), to prevent unauthorized individuals from participating and to prevent duplicate submissions by a single respondent. The survey software, produced by Quask[54] was satisfactory, its main weakness being that it was not capable of supporting double characters. These characters are used in languages such as Thai, Korean, and Russian. To remedy this problem, we provided the survey questionnaire to the respondents from these countries as a Word document.

5.3.2 Survey questions

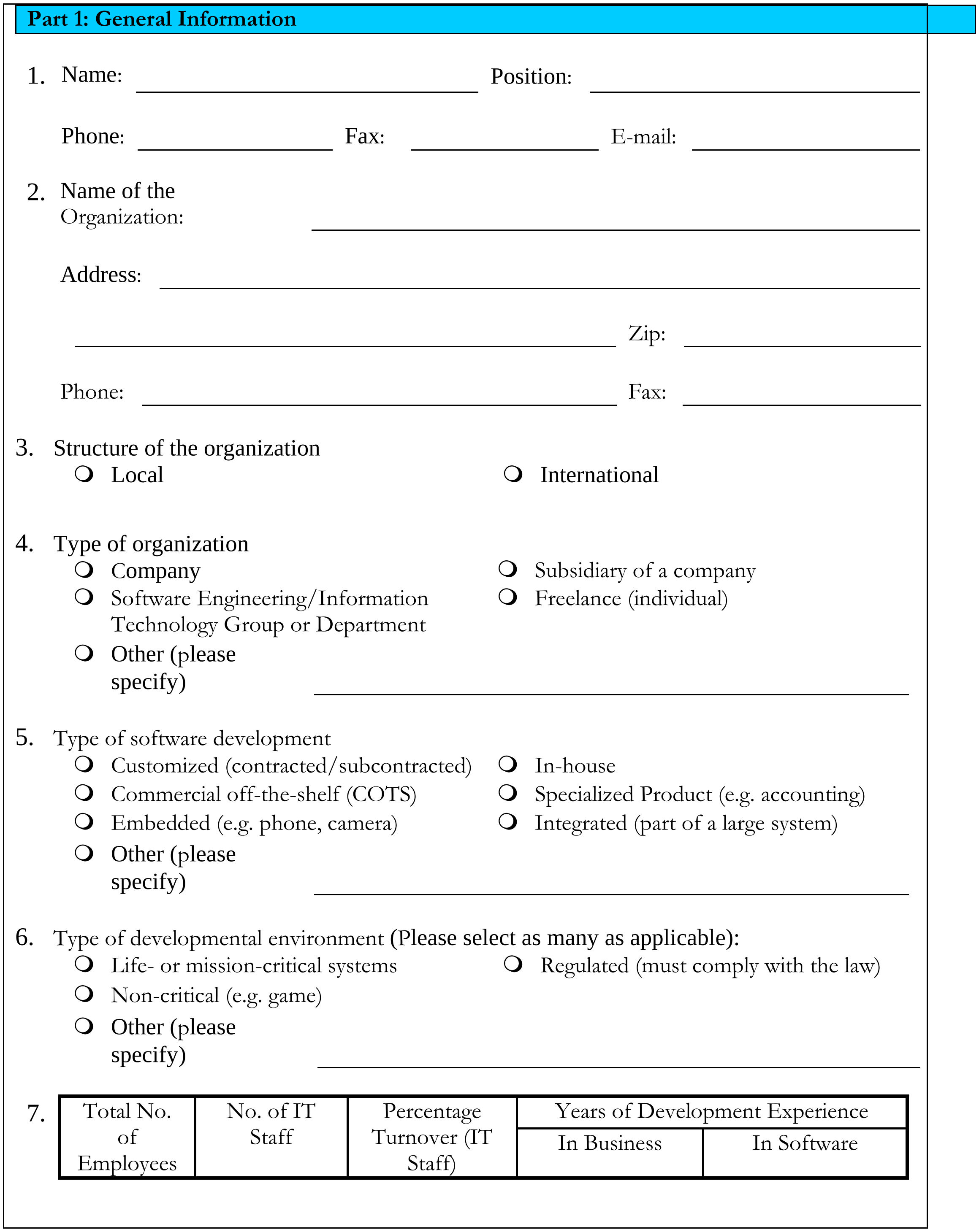
The survey is made up of 20 questions structured in 5 parts: general information about the organization, information about standards utilization in the organization, information about

[54] www.quask.com

implementation and assessment problems in the organization, information about the needs of the organization, and information about justification for compliance with standards. The information requested in each section of the survey is outlined below.

Part 1 – General information

This section of the survey is designed to collect personal and demographic characteristics of the respondents, such as the name and size of the organization and the type of software developed. Figure 5-4 lists these questions.



Part 1: General Information

1. Name: ______________________ Position: ______________________

   Phone: ______________ Fax: ______________ E-mail: ______________

2. Name of the Organization: ______________________________

   Address: ______________________________

   ______________________________ Zip: ______________

   Phone: ______________________________ Fax: ______________

3. Structure of the organization
   - ○ Local
   - ○ International

4. Type of organization
   - ○ Company
   - ○ Software Engineering/Information Technology Group or Department
   - ○ Other (please specify) ______________________________
   - ○ Subsidiary of a company
   - ○ Freelance (individual)

5. Type of software development
   - ○ Customized (contracted/subcontracted)
   - ○ Commercial off-the-shelf (COTS)
   - ○ Embedded (e.g. phone, camera)
   - ○ Other (please specify) ______________________________
   - ○ In-house
   - ○ Specialized Product (e.g. accounting)
   - ○ Integrated (part of a large system)

6. Type of developmental environment (Please select as many as applicable):
   - ○ Life- or mission-critical systems
   - ○ Non-critical (e.g. game)
   - ○ Other (please specify) ______________________________
   - ○ Regulated (must comply with the law)

7.

Total No. of Employees	No. of IT Staff	Percentage Turnover (IT Staff)	Years of Development Experience	
			In Business	In Software

Figure 5-4 General information about the respondent

Part 2 – Information about the standards used by VSEs

These questions were designed to collect information about the standards used by the respondents, as well as their reason(s) for not using standards. These questions are listed in Figure 5-5.

Part 2: Standards use by VSEs

8. Are you certified to any standard(s)?
   - ○ Yes (go to **10**)
   - ○ No. Are you using any standards? ○ Yes (go to **10**) ○ No (go to **9**)

9. Why are you *not* using any standards? (Please select as many as applicable and go to **15**)

Not required [ ] By customer [ ] By organization
Lack of support [ ] From top management [ ] From government [ ] From industry
Lack of resources [ ] Tools [ ] Financial [ ] Staff
Too time-consuming [ ] Implementation [ ] Assessment/Certification
Standard(s) [ ] Difficult/Bureaucratic [ ] Not enough guidance
[ ] Incompatible with existing management systems/approaches
Other (please specify) ____

10. What quality or process improvement models, approaches, or standards does your organization currently use in its improvement efforts? (Please select as many as applicable)

SECM EIA/IS 731 ○ INCOSE SECAM ○ MBNQA ○ ITIL ○ ISMS
ISO 9001:2000 [ ] 12207 [ ] 15288 [ ] 15504 [ ] 15939 [ ]
SEI CMMI [ ] SW-CMM [ ] SE-CMM [ ] SA-CMM [ ] IPD-CMM [ ]
National Standard (please specify) ______________________________________
Other (please specify)____

11. What are your reasons for implementation of the selected models, approaches, or standards? (Please select as many as applicable)

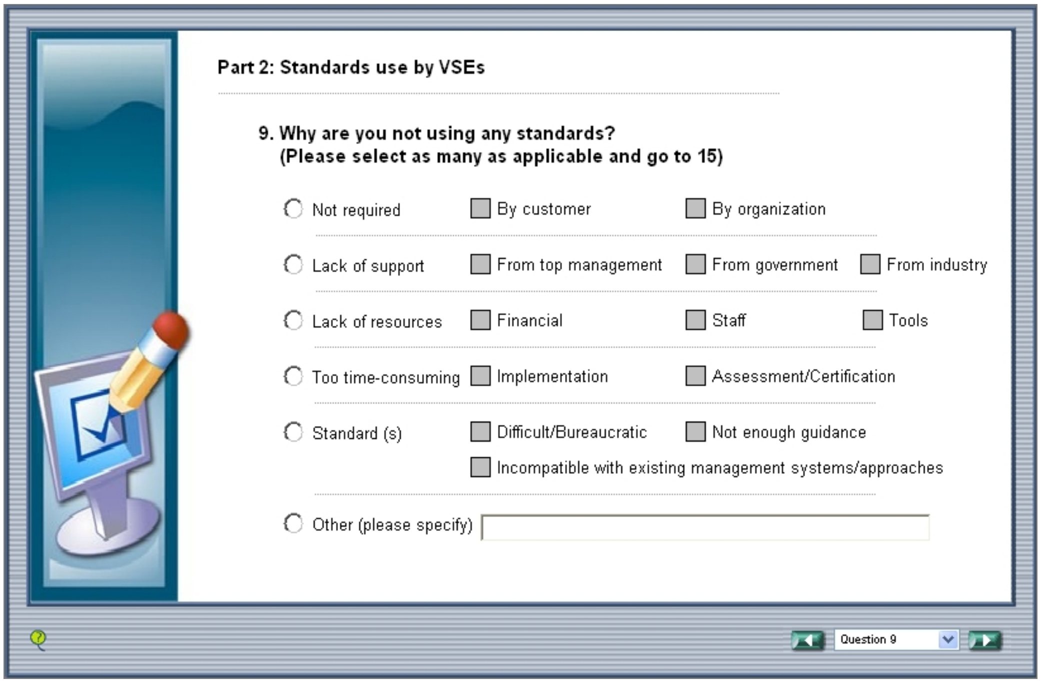
○ Required by/ [ ] Domestic market [ ] Export market
Needed by [ ] Customer [ ] Industry (e.g. Financial/Insurance)
○ To improve [ ] Product quality [ ] Productivity
[ ] Process [ ] Access to trade
○ To increase [ ] Customer satisfaction [ ] Software engineering capability [ ] Sales
○ To reduce [ ] Costs [ ] Customer complaints [ ] Defects
○ Other (please specify) ____

12. Which assessments or certifications have you performed or plan to perform?

○ CMMI ○ SPICE/15504 ○ ISO9001:2000
National Standard (please specify) ______________________________________
Other (please specify) ____

Figure 5-5 Section of the questionnaire requesting standards usage information

Figure 5-6 illustrates an example of a question as seen by a respondent on the Internet.



Part 2: Standards use by VSEs

9. Why are you not using any standards?
(Please select as many as applicable and go to 15)

○ Not required □ By customer □ By organization

○ Lack of support □ From top management □ From government □ From industry

○ Lack of resources □ Financial □ Staff □ Tools

○ Too time-consuming □ Implementation □ Assessment/Certification

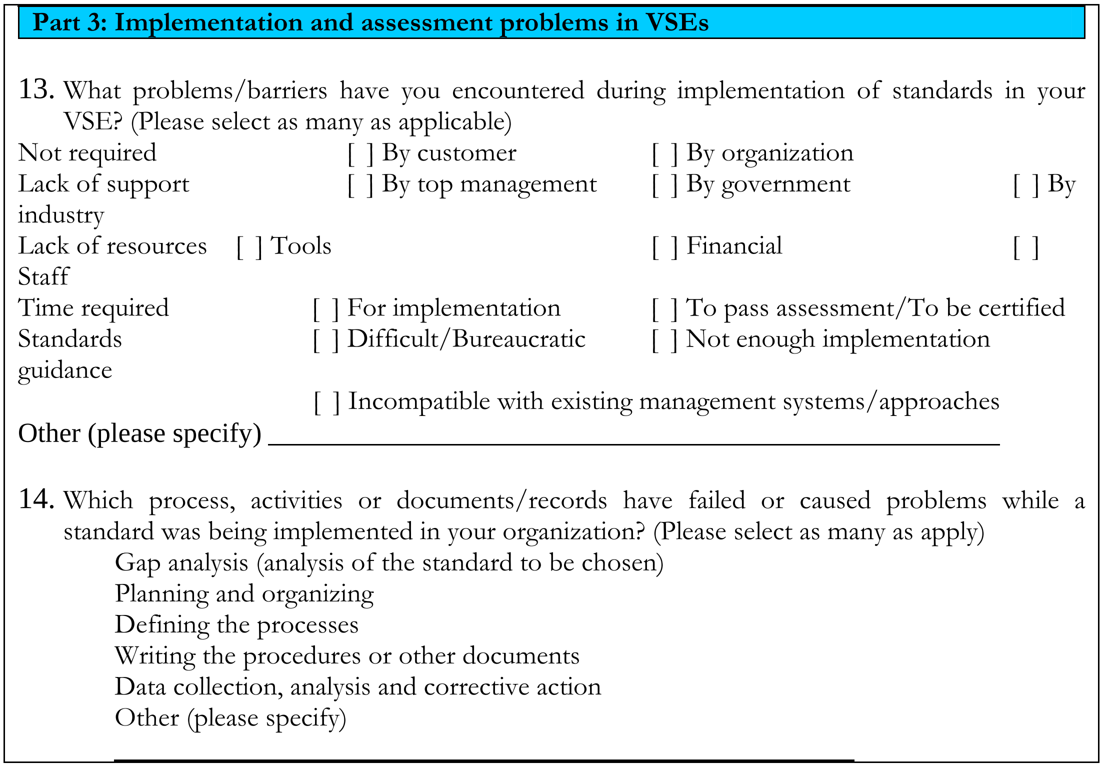
○ Standard (s) □ Difficult/Bureaucratic □ Not enough guidance
□ Incompatible with existing management systems/approaches

○ Other (please specify) ____________

Figure 5-6 Web page for the section requesting standards usage information

Part 3 – Information about implementation problems in VSEs

This section, illustrated in Figure 5-7, asked respondents to identify the barriers or problems encountered during the implementation of standards.



Part 3: Implementation and assessment problems in VSEs

13. What problems/barriers have you encountered during implementation of standards in your VSE? (Please select as many as applicable)

Not required [ ] By customer [ ] By organization
Lack of support [ ] By top management [ ] By government [ ] By industry
Lack of resources [ ] Tools [ ] Financial [ ] Staff
Time required [ ] For implementation [ ] To pass assessment/To be certified
Standards [ ] Difficult/Bureaucratic [ ] Not enough implementation guidance
[ ] Incompatible with existing management systems/approaches
Other (please specify) ____

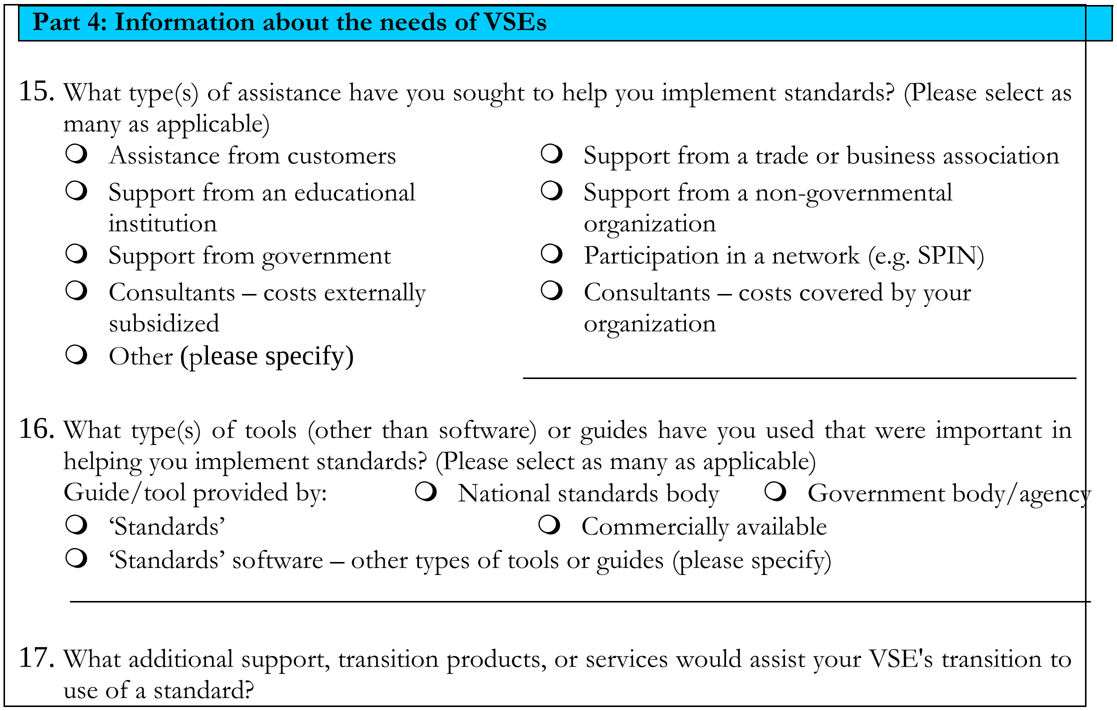
14. Which process, activities or documents/records have failed or caused problems while a standard was being implemented in your organization? (Please select as many as apply)

Gap analysis (analysis of the standard to be chosen)
Planning and organizing
Defining the processes
Writing the procedures or other documents
Data collection, analysis and corrective action
Other (please specify)
____

Figure 5-7 Section of the questionnaire requesting information about implementation problems

Part 4 – Information about the needs of VSEs in implementing standards

This section of the questionnaire, illustrated in Figure 5-8, collected information about the type of assistance VSEs need to help them implement standards.



Part 4: Information about the needs of VSEs

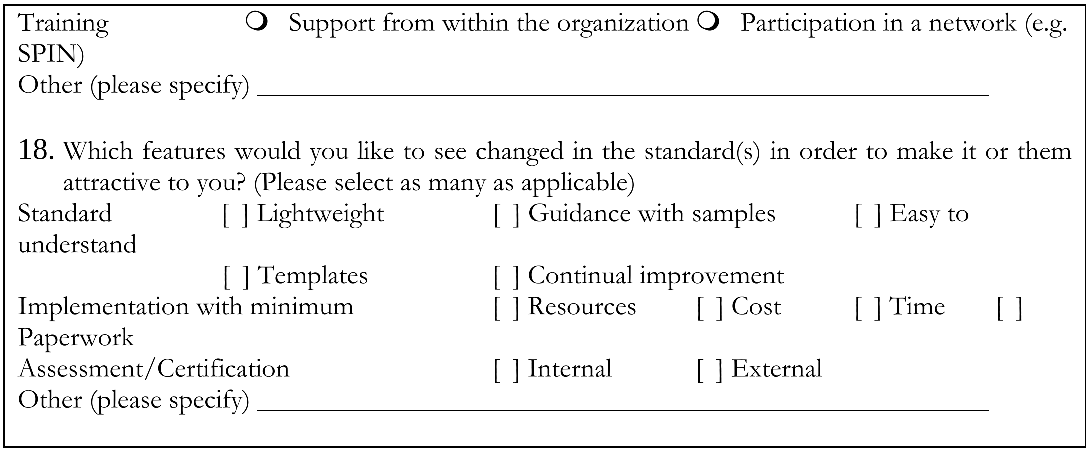
15. What type(s) of assistance have you sought to help you implement standards? (Please select as many as applicable)

○ Assistance from customers
○ Support from an educational institution
○ Support from government
○ Consultants – costs externally subsidized
○ Other (please specify)
○ Support from a trade or business association
○ Support from a non-governmental organization
○ Participation in a network (e.g. SPIN)
○ Consultants – costs covered by your organization
____

16. What type(s) of tools (other than software) or guides have you used that were important in helping you implement standards? (Please select as many as applicable)

Guide/tool provided by: ○ National standards body ○ Government body/agency
○ 'Standards' ○ Commercially available
○ 'Standards' software – other types of tools or guides (please specify)
____

17. What additional support, transition products, or services would assist your VSE's transition to use of a standard?



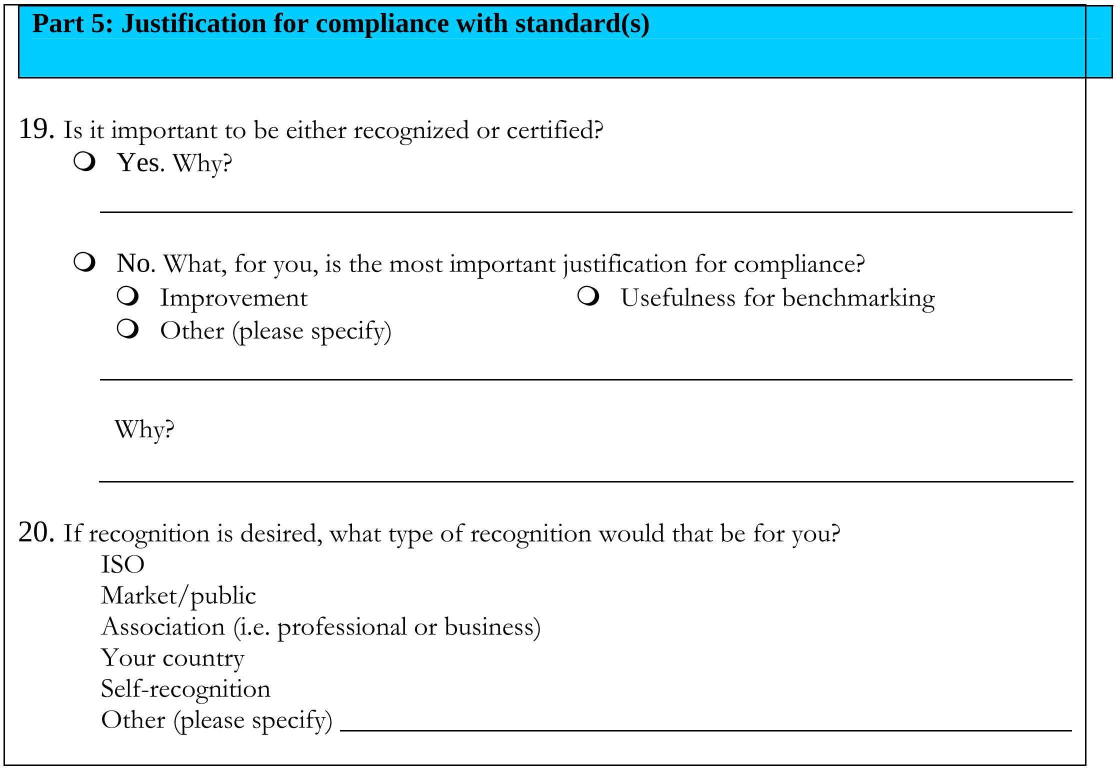
Training ○ Support from within the organization ○ Participation in a network (e.g. SPIN)
Other (please specify) ____

18. Which features would you like to see changed in the standard(s) in order to make it or them attractive to you? (Please select as many as applicable)
Standard [ ] Lightweight [ ] Guidance with samples [ ] Easy to understand
[ ] Templates [ ] Continual improvement
Implementation with minimum [ ] Resources [ ] Cost [ ] Time [ ] Paperwork
Assessment/Certification [ ] Internal [ ] External
Other (please specify) ____

Figure 5-8 Section of the questionnaire requesting information about the needs of VSEs in implementing standards

Part 5 – Justification for compliance with standards

This section, illustrated in Figure 5-9, asked respondents to state if it was important for them to be certified or recognized. The working group also wanted to know the motivations for compliance with a standard.



Part 5: Justification for compliance with standard(s)

19. Is it important to be either recognized or certified?
○ Yes. Why?
____

○ No. What, for you, is the most important justification for compliance?
○ Improvement ○ Usefulness for benchmarking
○ Other (please specify)
____

Why?
____

20. If recognition is desired, what type of recognition would that be for you?
ISO
Market/public
Association (i.e. professional or business)
Your country
Self-recognition
Other (please specify) ____

Figure 5-9 Section of the questionnaire requesting justification for compliance with standards

5.3.3 Description of the Communication Plan

A communication plan has been developed to specify the purpose, activities, and roles of those responsible for carrying out the survey (Laporte et al. 2005c). The goals of the plan were to:

- Inform the members of WG24, especially those unable to participate in a meeting, about the survey,
- Ensure that the survey of VSEs is conducted in as many countries as possible,
- Facilitate the rollout of the survey,
- Develop a wide network of VSEs, government agencies, and centers to help WG24 obtain reviews of draft ISPs and TRs, and support the conducting of future pilot projects.
- Facilitate the deployment and adoption of proposed ISPs, TRs, etc.

The Communication Plan includes the following sections:

- The Overview Section, which provides high-level information on the organization, goals, and time frames associated with the Plan.
- The Key Group Section, which identifies the customers (i.e. target audience) of the Plan.
- The Communication Schedule Section, which provides a schedule of the communication activities of the Plan.

The overall timeframe of the Plan was defined over a period of 4 months:

- By November 2005, the survey was available in 9 languages, in order to obtain a wide coverage of VSEs all around the world. The survey was ported to a server at the ÉTS, which was used to capture, in real-time, the responses of the survey conducted from VSEs around the world. A list of contacts in as many countries as possible was also developed. The contacts were asked to broadcast the announcement of the survey.
- In December of that year, the Web version of the survey was ready for testing. The members of the working group were asked to test the site by answering the questionnaire in all 9 languages. This allowed the author to verify that the survey

software was properly collecting and storing the survey data in the database of the survey tool.

- In January 2006, the list of contacts was used to deploy the survey internationally.
- In February 2006, a second round of solicitation was performed to encourage VSEs in countries with a low response rate to respond to the survey.
- In March 2006, the data from the survey were available for analysis by the WG.

5.3.4 Conducting of the survey

A mailing list was created using WG24 members' contact networks. We also contacted centers and software engineering professors focusing on the concerns of small software enterprises, such as the CETIC Center in Belgium, the Centre de Recherche Public Henri-Tudor (CRDP) in Luxembourg, the Thai Software Industry Promotion Agency (SIPA), the European Software Institute (ESI), the Parquesoft organization in Colombia, the Japan Information Technology Promotion Agency (JITEC), Enterprise Ireland, and the Software Process Improvement Networks (SPINs[55]) supported by the Software Engineering Institute worldwide.

The author, having been involved for many years in the establishment and management of the Montreal SPIN and having participated in many SPIN conferences, took great care to contact all the SPINs worldwide, and especially those in the United States. The SPIN distribution list included the chairperson of the following categories: active U.S. Software SPINs, emerging U.S. SPINs, active International SPINs, emerging U.S. SPINs, and emerging International SPINs. The author sent over 200 e-mails to those individuals, inviting them to broadcast the request to participate in the survey to the members of their SPINs.

The survey was launched in February 2006, and, as of June 2006, over 392 responses had been collected from 29 countries. Table 5-2 illustrates the rate at which the survey was completed around the world. Additional data was also collected between July 2006 and November 2007. Over 437 responses from 35 countries were collected from February 2006 to November 2007.

5.3.5 Data analysis

Of the 392 responders[56], 228 were enterprises with 0 to 25 employees (58%), as illustrated in Figure 5-10. These 228 VSEs constitute the sample for this study. Findings common to

[55] http://www.sei.cmu.edu/collaborating/spins

[56] Only the data captured up to June 2006 were analyzed.

those VSEs are presented below, and correlations inside the sample and findings that differ from those of the larger companies that contributed to the survey are identified.

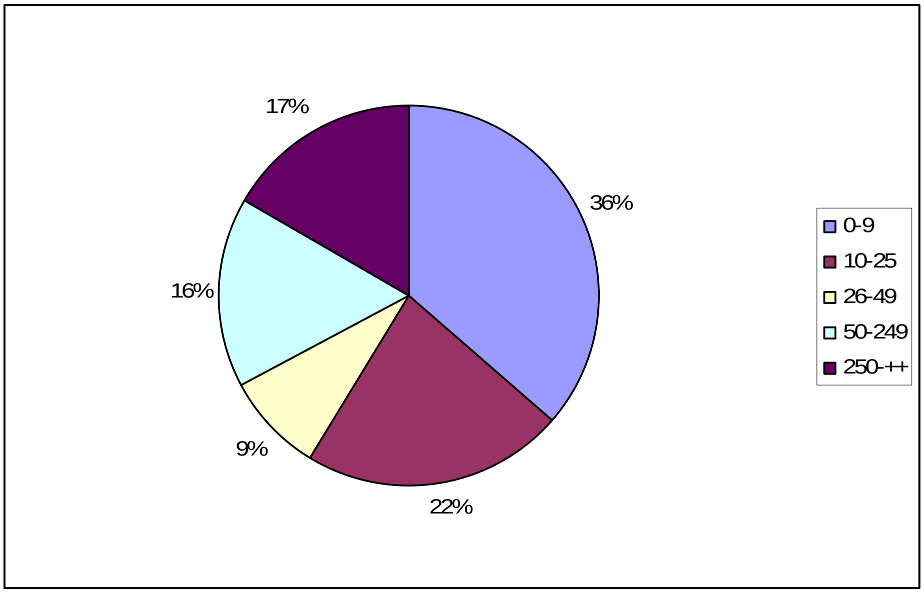


Figure 5-10 Number of employees in the enterprises surveyed

This categorization and several studies underscore the differences between micro, small, and medium-sized enterprises in terms of available resources. WG24 decided to focus on the first category (micro enterprises with 0-9 employees) and on a sub part of the small enterprise category (10-25 employees).

	March 10	March 17	March 24	March 31	April 7	April 14	April 21	April 28	May 5	May 12	May 19	May 26	June 9	June 23	July 7	July 21	July 28	August 4	Sept 1	Sept 22	Oct 20	Nov 17
Argentina	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Australia													8	10	10	10	10	10	10	10	10	10
Belgium				1	8	9	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
Brazil	45	50	53	54	54	65	65	65	66	67	67	68	68	68	69	69	69	70	70	70	71	72
Bulgaria			2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Canada			1	1	4	4	6	7	7	7	7	7	8	8	9	9	9	9	9	10	10	10
Chile		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Colombia	2	4	4	5	20	35	50	55	56	60	66	66	88	102	107	108	109	109	109	109	109	109
Czech Republic				1	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Dominican Republic																	1	1	1	1	1	1
Ecuador	3	6	8	8	8	8	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
Finland		9	9	12	12	12	12	12	12	12	12	12	13	13	13	13	13	13	13	13	13	13
France									1	2	3	3	3	3	3	3	3	3	4	4	4	4
Germany																	1	1	1	1	1	1
India		2	2	3	41	46	55	56	56	56	57	57	57	57	57	57	57	57	57	57	57	57
Ireland				1	4	6	9	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
Italy								1	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Japan	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Korea(South)			2	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Luxembourg																			2	3	3	3
Mexico			2	11	17	17	18	19	20	20	20	20	20	20	20	20	20	20	20	20	20	20
Morocco																						1
New Zealand	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Peru		1	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4	4	4	4
Russia			4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
South Africa		6	9	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
Spain			1	1	1	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	4	4
Taiwan													1	1	1	1	1	1	1	1	1	1
Thailand				1	1	1	7	41	46	48	50	51	52	54	56	57	58	58	58	58	59	59
Turkey		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
United Kingdom	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
United States		2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Total	**56**	**89**	**112**	**135**	**208**	**244**	**282**	**327**	**337**	**345**	**356**	**358**	**392**	**411**	**420**	**422**	**426**	**427**	**430**	**432**	**435**	**437**

Table 5-2 Survey Response Rate

5.3.6 General characteristics

Here, we draw attention to some weaknesses of the sample itself. Since the survey was initiated through WG24 contacts without a true random sample being built, the survey results may have been impacted. The first observation about the respondent sample, as illustrated in Table 5-3, is the geographical distribution of the responses. We collected a high number from Latin America (46%), mainly from Colombia (22%) and Brazil (17%).

Country	**Number of Responses**	**Country**	**Number of Responses**
Argentina	2	Italy	2
Australia	10	Japan	3
Belgium	10	South Korea	4
Brazil	70	Luxembourg	2
Bulgaria	3	Mexico	20
Canada	9	New Zealand	1
Chile	1	Peru	4
Colombia	109	Russia	4
Czech Republic	3	South Africa	10
Dominican Republic	1	Spain	3
Ecuador	9	Taiwan	1
Finland	13	Thailand	58
France	4	Turkey	1
Germany	1	United Kingdom	2
India	57	United States	3
Ireland	10		

Table 5-3 Number of Survey Responses per Country

One of these organizations, the Software Industry Promotion Agency (SIPA) of Thailand, acted as a host for the 2006 ISO/IEC JTC1/SC7 Plenary meeting. SIPA organized a series of free tutorials for their members the week before the SC7 meeting. One condition for the participation of an SIPA member in the tutorials was to respond to the survey. This resulted in over 58 responses from Thai VSEs. In Colombia, Parquesoft designated an individual to solicit VSEs and help them complete the survey. The fact that there were over 100 VSEs in the Parquesoft group at the time explains the high number of responses received from that country. At the same time, we received only a few responses from European countries (48), Japan (3), and the United States (3). The author was very surprised by the extremely low participation level in the United States, especially since over

100 e-mails had been sent to the chairperson of the U.S. SPIN group. Possible reasons for this are the following:

- The invitation to participate in the survey was not distributed in some countries.
- Many SPIN members were employed in larger companies not directly targeted by this survey.
- For some unknown reason, SPIN chairpersons did not broadcast the request to their members.
- Most SPIN members already use CMMI, and so they may not be interested in ISO standards.
- Most VSEs do not care about IT standardization, so only those aware of it took the time to contribute.

Our results might, therefore, only generalize to the broader populations of projects in each region to the extent that this sample represented them. Moreover, we had no evidence that participating companies were representative of the situation in their own countries. Conclusions drawn from these survey results should be confirmed with additional responses. To achieve this objective, WG24 plans to keep the survey Web site up and running and launch another survey blitz.

The strong representation of Latin American countries in the sample had no impact on the final results of the study. These VSEs differ from the rest of the respondents in the types of development in which they are involved, i.e. more specialized products, and the application domain, as they were more involved in critical applications at the time, with almost 50% of VSEs working in these fields.

Among the respondents, the majority (79%) were private companies and 78% operated at the national level only. Regarding the application domain, almost half the respondents were working either on life-/mission-critical systems or on regulated projects, as shown in Figure 5-11. Over 40% of the respondents were developing software for life-/mission-critical systems and 34% on regulated development projects.

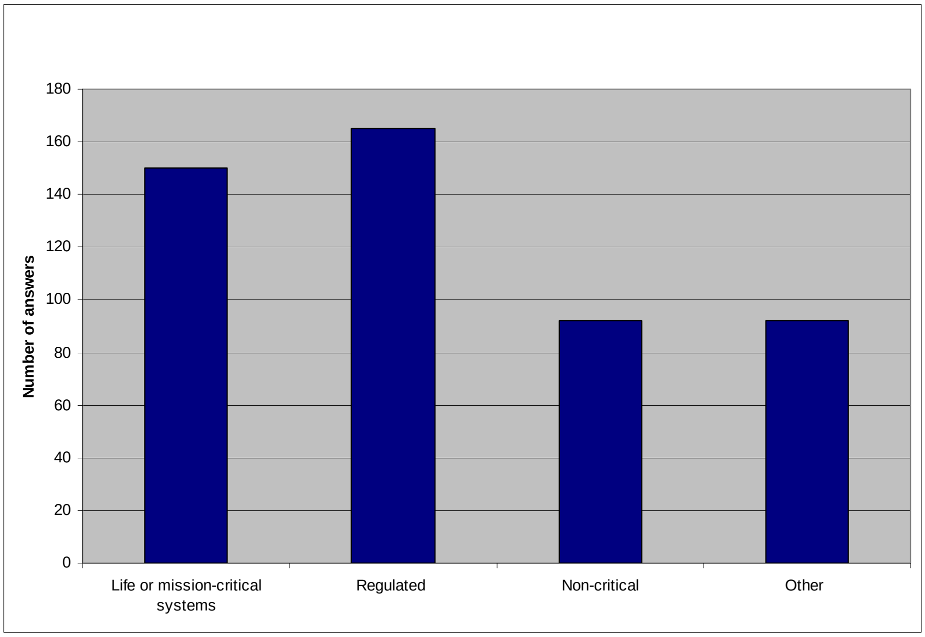


Figure 5-11 Application domain

With regard to the types of software development, a large majority were involved in the development of customized or tailor-made software and specialized products, as shown in Figure 5-12.

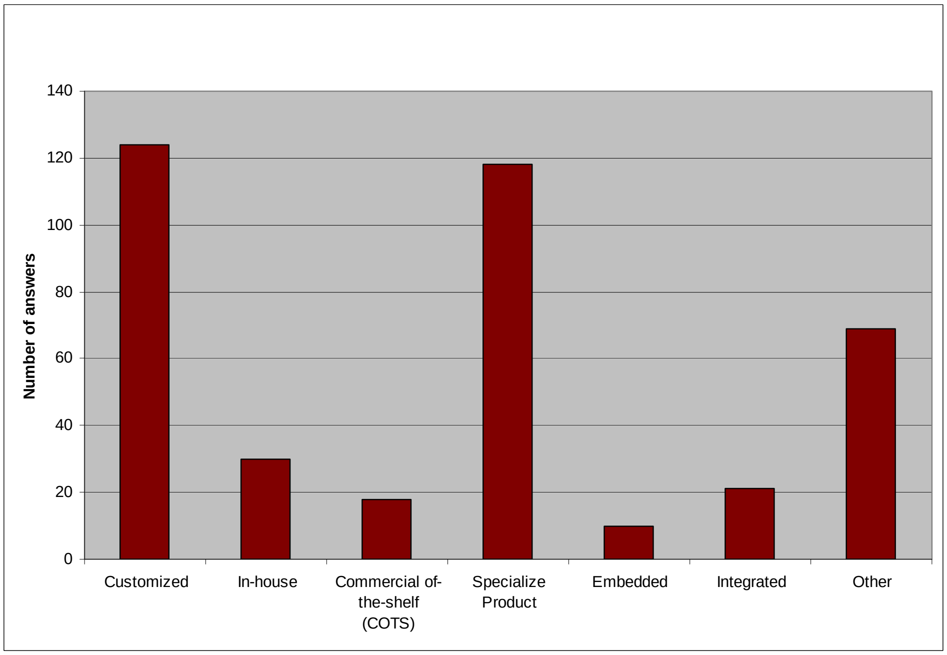


Figure 5-12 Types of software development

5.3.7 Features of VSE results

More than 70% of the VSEs were either working on life- or mission-critical systems, or in a regulated market. This underscored our hypothesis concerning the awareness of the participating companies, as it was assumed that companies working in these particular contexts were prone to using standards for contractual reasons.

An interesting finding of the survey was the difference in the percentage of certified companies with regard to company size: fewer than 18% of VSEs were certified, while 53% of larger companies (those with more than 25 employees) claim to be certified. Furthermore, among the 18% not certified, 75% did not use standards. In larger companies using standards, two families of standards and models emerged from the list: ISO standards (55%) and models from the Software Engineering Institute (SEI) (47%).

WG24 anticipated the limited use of standards by VSEs by asking questions designed to provide a better understanding of the reasons for this. There were three main ones, as shown in Figure 5-13. The first was a lack of resources (28%); the second was that standards were not required (24%); and the third derived from the nature of the standards themselves: 15% of the respondents considered that the standards were difficult to meet and bureaucratic, and that insufficient guidance was provided for use in a small business environment.

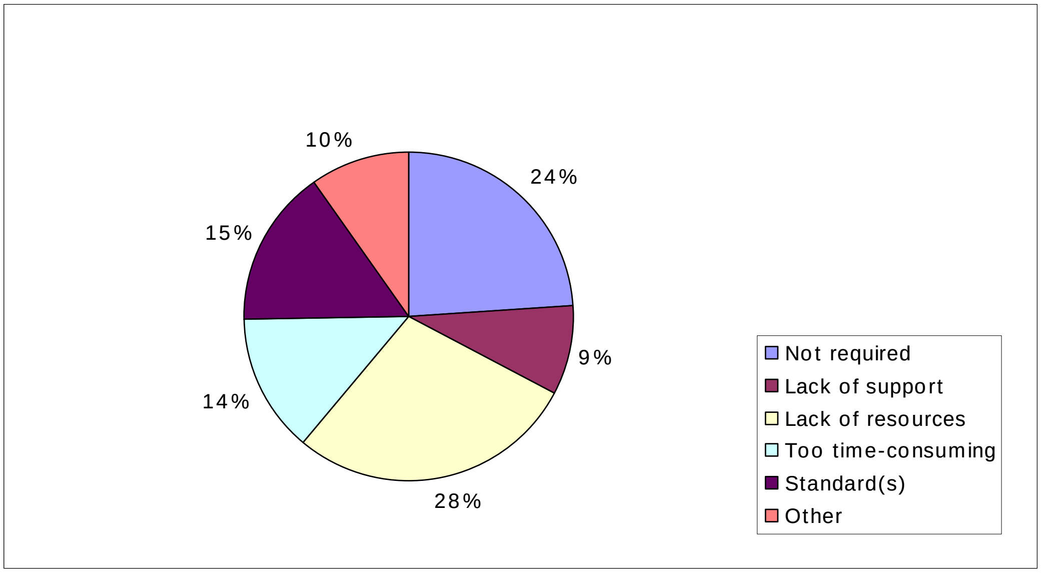


Figure 5-13 Why don't VSEs use standards?

However, for a large majority (74%) of VSEs, it was very important to be recognized or certified against a standard. ISO certification was requested by 40% of them. Of the 28% requesting official market recognition, only 4% were interested in a national certification. From the respondents of the survey, some benefits of certification were:

- Increased competitiveness
- Greater customer confidence and satisfaction

- Greater software product quality
- Increased sponsorship for process improvement
- Decreased development risk
- Facilitation of marketing (e.g. better image)
- Greater potential to export

However, VSEs expressed the need for assistance in order to adopt and implement standards. Over 62% indicated that they would like more guidance with examples, and 55% asked for lightweight and easy-to-understand standards, complete with templates. Finally, the respondents indicated that it had to be possible to implement standards with minimum cost, time, and resources.

5.4 Conclusion

The data collected from the survey on the problems and needs of VSEs and on standards utilization confirmed WG24's hypotheses about these issues. The results of the survey allowed the working group to refine some of the requirements, as illustrated in Figure 5-14, that will be used to develop profiles, guides, and deployment packages to meet the needs of VSEs.

- R07 - Implementation of the set of work products (i.e. Profiles, Guides) must be affordable (low cost of entry).
- R08 - Use of the set of work products must be affordable, and the set should be intuitive to use (i.e. consultant services should not be necessary).
- R09 - The use of the set of work products should yield an identifiable benefit to the VSE. The use of the set of work products should increase productivity and quality in some measurable fashion.
- R15 - The set of work products should provide the whole spectrum of documents (from standards to education material, i.e. tutorial information).
- R29 - The set of work products should propose examples of life cycles from which to choose.
- R33 - The set of work products should propose definitions of documents (as assemblies of work products); for example, templates (e.g. requirements templates – use cases).
- R37 - The set of work products should include compliance table checklists (Assessment Guide).
- R47 - The guide should include support for documentation, particularly templates.
- R52 - The guide should provide examples (e.g. plans, work products, and other deliverables).
- R56 - The guide should be compact (about 50 pages), not including templates.
- R57 - The guide should be available free of charge on the Web.

Figure 5-14 Partial list of requirements (adapted from BK1-032 2005)

The next chapter presents a detailed description of the development of ISs and TRs targeting VSEs. The documents used by WG24 and the approach that led to the development and balloting of ISs and TRs for VSEs are also presented.

This page is intentionally left blank

Chapter 6

The Development of International Standards for VSEs

6 Chapter 6 – The Development of International Standards and Technical Reports for VSEs

6.1 Introduction

In the previous chapters, the first 2 phases of the 6-phase innovation process have been presented in detail, as illustrated in Figure 6-1. Those first 2 phases were conducted during the early meetings of the working group, along with the international survey. The third phase, Development, was partially covered in a previous chapter, on the development process of international standards (ISs). This chapter presents a detailed description of the development of those standards, as well as that of Technical Reports (TRs) targeting VSEs. The documents used by WG24 and the approach that led to the development, balloting of the ISs and TRs for VSEs are also presented.

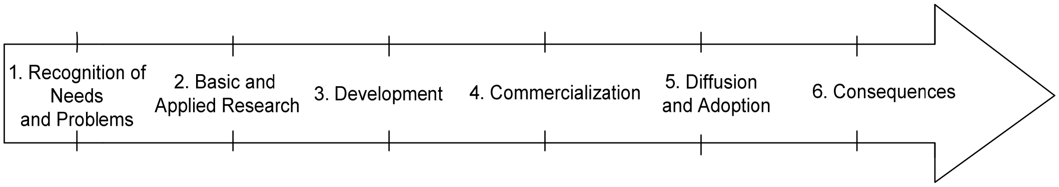

1. Recognition of Needs and Problems
2. Basic and Applied Research
3. Development
4. Commercialization
5. Diffusion and Adoption
6. Consequences

Figure 6-1 Phases of the Innovation Process (adapted from Rogers 2003)

Since an ISO standard dedicated to software life cycle processes was already available, i.e. ISO/IEC 12207 (ISO 2008d), WG24 decided to use the concept of the ISO International Standardized Profile (ISP) to develop the new standards for VSEs. A profile is defined as "a set of one or more base standards and/or ISPs, and, where applicable, the identification of chosen classes, conforming subsets, options and parameters of those base standards, or ISPs necessary to accomplish a particular function" (ISO 1998). From a practical point of view, a profile is a matrix that identifies the elements that are taken from existing standards from those that are not to produce an ISP.

The overall approach followed by WG24 to develop this new standard for VSEs consisted of three steps:

- Select, from existing standards, process subsets applicable to VSEs;
- Tailor the subset to fit VSE needs;
- Develop guidelines.

In the next section, the standards that have been used by WG24 to develop the ISs and TRs for VSEs are described.

6.2 Standards used to develop standard for VSEs

6.2.1 ISO/IEC 12207 – Software life cycle processes[57]

ISO/IEC 12207 establishes a framework for software life cycle processes and terminology: "It applies to the acquisition of systems and software products and services, to the supply, development, operation, maintenance, and disposal of software products and the software portion of a system, whether performed internally or externally to an organization" (ISO 2008d).

This standard defines two sets of processes (see Figure 6-2): in one of these, called Software Specific Processes, the final product is a standalone software product or service, and in another, called System Context Processes, the software is part of a larger system. Since most modern systems are controlled by software, this standard has been recently updated to 'interface' with the equivalent standard at the systems engineering level: ISO/IEC 15288:2008 Systems engineering – Systems life cycle processes (ISO 2008c).

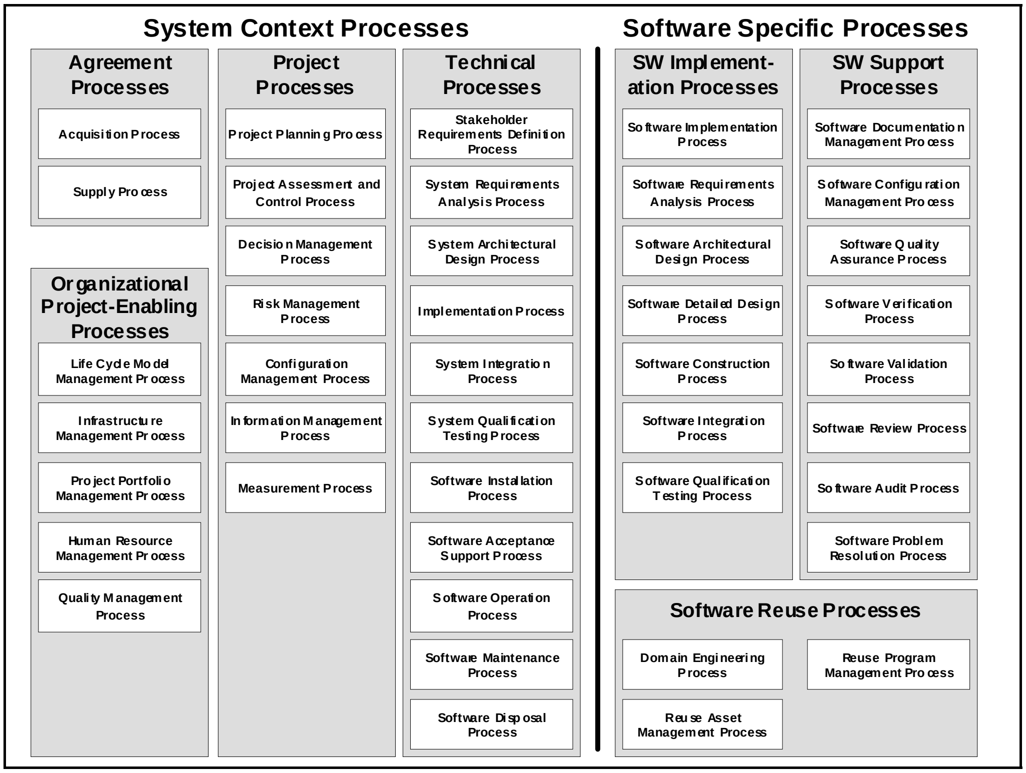

System Context Processes
Software Specific Processes
Agreement Processes
Acquisition Process
Supply Process
Organizational Project-Enabling Processes
Life Cycle Model Management Process
Infrastructure Management Process
Project Portfolio Management Process
Human Resource Management Process
Quality Management Process
Project Processes
Project Planning Process
Project Assessment and Control Process
Decision Management Process
Risk Management Process
Configuration Management Process
Information Management Process
Measurement Process
Technical Processes
Stakeholder Requirements Definition Process
System Requirements Analysis Process
System Architectural Design Process
Implementation Process
System Integration Process
System Qualification Testing Process
Software Installation Process
Software Acceptance Support Process
Software Operation Process
Software Maintenance Process
Software Disposal Process
SW Implementation Processes
Software Implementation Process
Software Requirements Analysis Process
Software Architectural Design Process
Software Detailed Design Process
Software Construction Process
Software Integration Process
Software Qualification Testing Process
SW Support Processes
Software Documentation Management Process
Software Configuration Management Process
Software Quality Assurance Process
Software Verification Process
Software Validation Process
Software Review Process
Software Audit Process
Software Problem Resolution Process
Software Reuse Processes
Domain Engineering Process
Reuse Program Management Process
Reuse Asset Management Process

Figure 6-2 Life cycle process groups (adapted from ISO 2008c)

Each 12207 process is described in terms of the following attributes, as defined in ISO TR 24774 (ISO 2007b):

[57] Adapted from: ISO/IEC 12207:2008 Systems and software engineering — Software life cycle processes (ISO 2008d).

- A title, which conveys the scope of the process as a whole. The title of a process is a short noun phrase intended to summarize the scope of the process.
- A purpose, which describes the goals of the process.
- A set of outcomes, which expresses what is expected from the execution of the process. An outcome is an observable result of the successful achievement of the process purpose.
- A set of activities, which is a list of actions that may be used to achieve the outcomes. Each activity may be further elaborated as a grouping of related lower-level actions (e.g. a task)
- A set of tasks, which is a list of specific actions that may be performed to achieve an activity. Multiple related tasks are often grouped within an activity.

To illustrate the structure of the 12207 standard, the software configuration management process is used as an example (see Figure 6-3).

Purpose
The purpose of the Software Configuration Management Process is to establish and maintain the integrity of the software items of a process or project, and make them available to concerned parties.

Outcomes
As a result of the successful implementation of the Software Configuration Management Process:

- a software configuration management strategy is developed;
- items generated by the process or project are identified, defined, and baselined;
- modifications and releases of the items are controlled;
- modifications and releases are made available to affected parties;
- the status of the items and modifications is recorded and reported;
- the completeness and consistency of the items is ensured; and
- the storage, handling, and delivery of the items are controlled.

Activities and Tasks
The project shall implement the following activities in accordance with applicable organization policies and procedures with respect to the Software Configuration Management Process:

Process Implementation. This activity consists of the following task:
A software configuration management plan shall be developed. The plan shall describe: the configuration management activities; procedures and schedule for performing these activities; the organization(s) responsible for performing these activities; and their relationship with other organizations, such as software development or maintenance. The plan shall be documented and implemented.

NOTE: The plan may be a part of the System Configuration Management Plan.

Configuration Identification. This activity consists of the following task:
A scheme shall be established for identification of software items and their versions to be controlled for the project. For each software item and its versions, the following shall be identified: the documentation that establishes the baseline; the version references; and other identification

details.

Configuration Control. This activity consists of the following tasks:
Change requests shall be identified and recorded; the changes shall be analyzed and evaluated; the request shall be approved or not; and the modified software item shall be implemented, verified, and released. An audit trail shall exist, whereby each modification, the reason for the modification, and authorization of the modification can be traced. Control and audit of all accesses to the controlled software items that handle safety- or security-critical functions shall be performed.

NOTE: The Software Problem Resolution Management Process could provide support for this activity.

Configuration Status Accounting. This activity consists of the following task:
Management records and status reports that show the status and history of controlled software items including baseline shall be prepared. Status reports should include the number of changes for a project, the latest software item versions, release identifiers, the number of releases, and comparisons of releases.

Configuration Evaluation. This activity consists of the following task:
The functional completeness of the software items against their requirements and the physical completeness of the software items (whether or not their design and code reflect an up-to-date technical description) shall be determined and ensured.

Release Management and Delivery. This activity consists of the following tasks:
The release and delivery of software products and documentation shall be formally controlled. Master copies of code and documentation shall be maintained for the life of the software product. The code and documentation that contain safety- or security-critical functions shall be handled, stored, packaged, and delivered in accordance with the policies of the organizations involved.

Figure 6-3 Software configuration management process (adapted from ISO 2008c)

As has been explained previously, the 12207 standard is almost useless to most VSEs, since these entities do not have the expertise to transform the requirements of 12207 into a set of usable processes. In the next chapter, the author describes the documents developed, i.e. the deployment packages, to further help VSEs in applying the 12207 standard.

6.2.2 ISO/IEC 15289 – Content of systems and software life cycle process information products (Documentation)[58]

This standard is a companion standard to ISO 12207. It is used to identify and plan the information items to be produced during a project. It describes the information content of different types of documents, such as a record or a plan. Table 6-1 lists the different types of documents described in the 15289 standard (ISO 2006a).

Type	**Purpose**	**Sample of recommended output information types**
Record	Characterizes the data retained by an organizational entity.	Configuration record Problem record
Description	Represents a planned or actual function, design, or item	High-level software design description

[58] Adapted from ISO/IEC 15289:2006, Systems and Software Engineering — Content of systems and software life cycle process information products (Documentation)

Type	Purpose	Sample of recommended output information types
Plan	Define when, how, and by whom specific activities are to be performed.	Project management plan
Procedure	Define in detail when and how to perform certain activities or tasks, including tools needed.	Problem resolution procedure
Report	Describe the results of activities such as investigations, assessments, and tests.	Problem report Validation report
Request	Record information needed to solicit a response.	Change request
Specification	Specify a required function, performance or process (such as, requirements specification, standard, policy).	Software requirements specification

Table 6-1 Life cycle Product Types (adapted from ISO 2006a)

Since most VSEs are not prone to documenting their project activities, and since most practitioners do not like to produce documentation, we can hardly expect them to develop a set of templates which describes the content and format of documents produced during a project. The deployment packages described in the next chapter describe the strategy used to help VSEs produce useful project documentation.

6.2.3 The Mexican national standard, MoProSoft

The Mexican standard has been developed with three levels of decisions in mind, as illustrated in Figure 6-4: the decisions made by top-level management, those made by middle management, and those made by the people who develop projects.

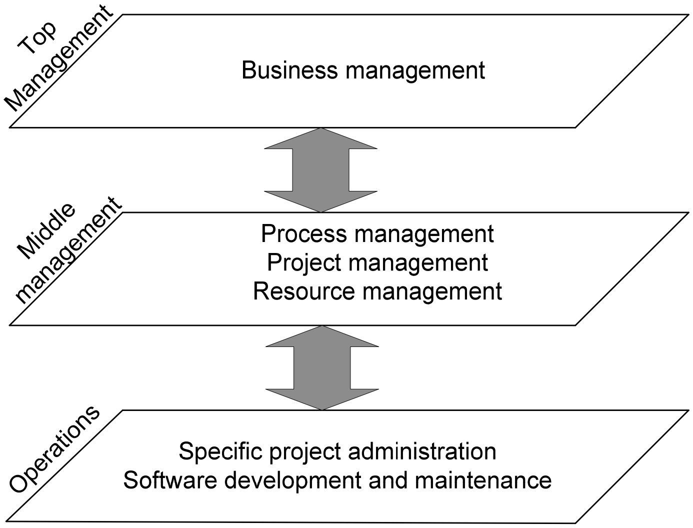


Figure 6-4 MoProSoft's process categories (Oktaba et al. 2007)

Processes are grouped into three categories: Top Management, Middle management, and Operations (adapted from NMX 2005):

- *Top management*. Members in this category are concerned with business-management practices, and receive and direct reports from middle management.

- *Middle management*. Members in this category deal with process-, project-, and resource management practices in line with top management's business goals. They provide elements for the performance of operations processes, receive and evaluate the information those processes generate, and inform top management of the results. The resource management process includes three sub processes: human resources and work environment; goods, services, and infrastructure; and knowledge of the organization.

- *Operations*. Members in this category address the practices of software-development and -maintenance projects. They perform activities using elements provided by management, and deliver reports and the software products generated.

WG24 decided to use the Mexican standard as a framework to develop the standards for VSEs. The approach used by WG24 to develop a set of standards and technical reports is described in the next section.

6.3 The Approach Used by WG24

Since WG24 wanted to prepare an initial set of standards as quickly as possible, WG24 analyzed international reference standards and models that could help subset ISO 12207 for low-maturity VSEs. To create these initial products quickly, WG24 began a search for existing standards or models that could be tailored or adapted to the needs of VSEs. MoProSoft, a Mexican standard (NMX 2005) developed to assist small and medium-sized Mexican enterprises (SMEs) was selected to achieve this objective.

WG24 decided to use the notion of the profile to develop standards to meet the needs of VSEs. A profile is a grouping of one or more base standards to accomplish a particular function. The notion of the profile was selected for the following reasons (adapted from ISO 2009b):

- Standards generally target large enterprises, making initial compliance difficult for VSEs;

- Preparing profiles with progressive capability levels enables a stepwise approach to full compliance;

- SE Standards are generally large, and specify many elements that are not necessarily applicable to VSEs;

- Preparing profiles that subset the base standards facilitates the match between the standards and the targeted VSEs;
- Since the ISO standards do not necessarily cover all the topics needed, profiles can be used to integrate required elements not yet in the ISO standards catalog.

To assemble profiles, WG24 used two types of standards:

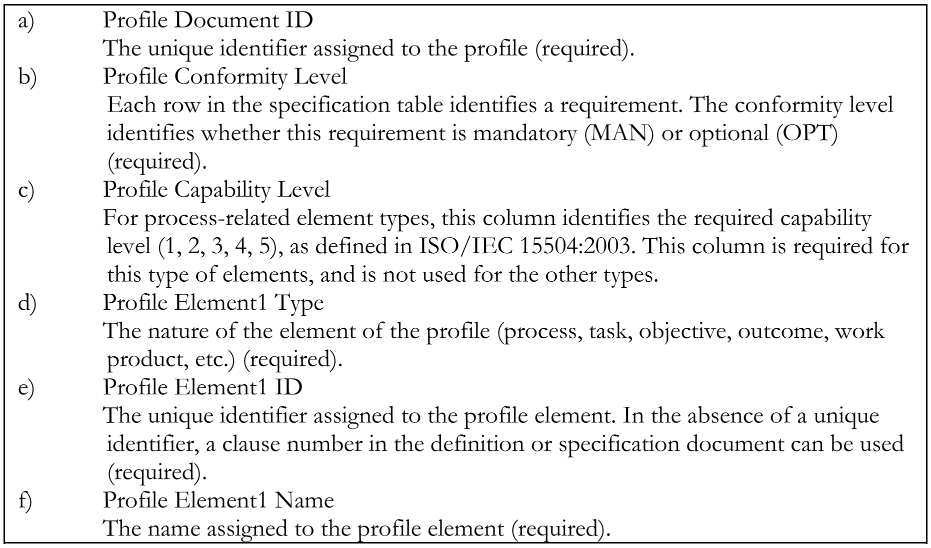
- Process standards, such as ISO 12207, which define the activities required to achieve identified objectives or outcomes;
- Product standards, such as ISO 15289, which define the structure and content of artifacts produced by the processes.

WG24 defined the scope of the life cycle processes described in the set of ISPs and TRs as follows (adapted from ISO 2009b): they were not intended to preclude or discourage their use by organizations larger than VSEs. Certain issues faced by large organizations may not be covered by this set of ISPs. The life cycle processes can be used by VSEs when acquiring and using, as well as when creating and supplying, a software system. They can be applied at any level in a software system's structure and at any stage in the life cycle. The processes described were not intended to preclude or discourage the use of additional processes that VSEs find useful.

6.3.1 Specification of a profile

A profile includes the following (adapted from ISO 2009b):

- The profile element identification and composition part, which identifies elements in the profile, as listed in Figure 6-5.



a) Profile Document ID
The unique identifier assigned to the profile (required).

b) Profile Conformity Level
Each row in the specification table identifies a requirement. The conformity level identifies whether this requirement is mandatory (MAN) or optional (OPT) (required).

c) Profile Capability Level
For process-related element types, this column identifies the required capability level (1, 2, 3, 4, 5), as defined in ISO/IEC 15504:2003. This column is required for this type of elements, and is not used for the other types.

d) Profile Element1 Type
The nature of the element of the profile (process, task, objective, outcome, work product, etc.) (required).

e) Profile Element1 ID
The unique identifier assigned to the profile element. In the absence of a unique identifier, a clause number in the definition or specification document can be used (required).

f) Profile Element1 Name
The name assigned to the profile element (required).

Figure 6-5 Elements of the composition columns (ISO 2009b)

- The profile element relationship specification tables: when a profile contains the specification for a relationship between two elements; for instance, if activity A produces work product W, this relationship is specified by the identification of each element, as described above, separated by a relationship type.

- Profile Relationship Type: the relationship is expressed by a meaningful abbreviation; for instance, INP for *uses as input*.

- The source document reference table identifies which elements in the source documents have been selected to be part of the profile. These tables are created by adding the columns to the tables specified in the two previous clauses, as described in Figure 6-6.

a) Source Document ID
The unique identifier assigned to the source document (required).

b) Source Conformity Level
The conformity level of the source requirement. The conformity level identifies whether this requirement is mandatory (MAN) or optional (OPT) (required).

c) Source Element1 Type
The nature of the element of the source document (process, task, objective, outcome, work product, etc.) (required).

d) Source Element1 ID
The unique identifier assigned to the source element. In the absence of a unique identifier, a clause number in the source document can be used (required).

e) Source Element1 Name
The name assigned to the source element (optional, for readability purposes).

f) Source Element1 Property(ies)
The selected value of the selected property applicable to the selected element (required if applicable).

g) Source Element1 Mapping Notes
Explanatory text on the selection and the correspondence (optional).

Figure 6-6 Elements of the part columns (ISO 2009b)

When a profile has selected both informative and normative elements from the source documents, then these must be clearly identified to facilitate conformance evaluation and assessment. An example of the application of these specifications is provided in a later section.

6.3.2 Conformance to an ISP

The purpose of an ISP is to specify the use of sets of specifications to provide clearly defined functionality. Hence, conformance to a 29110 ISP specification always implies conformance to the referenced base standards specifications, if it is referenced in its entirety in the profile. Conformance is specified within each ISP specification document published as a separate document, called Part 4. ISPs are pre-tailored packages of related software engineering standards, therefore:

- ISPs cannot be tailored;
- Partial compliance is not allowed (except in one case described below);
- There are no levels of conformance.

It is acceptable for an implementation to incorporate functionality beyond what is defined in the specification of the profile. This is called an extension. If a profile allows extensions, each implementation shall fully support all the required functionality of the profile specification exactly as specified, and the extensions shall neither contradict the functionality defined in the profile specification, nor cause its non-conformance.

6.3.3 Generic Profile Group

WG24 decided to develop an initial profile group, titled Generic Profile Group (GPG). A profile group (PG) is defined as a collection of profiles related either by composition of processes (i.e. activities, tasks), or by capability level, or both. A GPG contains profiles which are applicable to a vast majority of VSEs that do not develop mission-critical software or commercial off-the-shelf software products, and which are characterized by typical situational factors. Mission-critical software is defined as software whose failure could have an impact on safety, or could cause major financial or social losses (IEEE 1990). Membership in this PG does not imply any specific application domain; however, it is envisaged that new domain-specific profiles may be developed in the future (ISP 2009b). In the final chapter, the author proposes a few domain-specific profiles.

WG24 decided to develop a set of four profiles for the GPG to provide VSEs with a roadmap for growing in capability. The following profiles have been identified: Entry, Basic, Intermediate, and Advanced. In the next section, the Basic Profile, which was developed first, is presented. The other profiles are presented in subsequent sections.

6.4 Overview of the Basic Profile and its development[59]

The purpose of the Basic VSE Profile is to define software implementation and project management elements for a subset of processes and outcomes of ISO12207 and ISO15289 products, appropriate for a set of common VSE characteristics. The main reason to include project management is that the core business of VSEs is software development, and their financial success depends on successful project completion within schedule and on budget, as well as on making a profit. The Basic Profile describes the development of a single software application by a single project team with no special risk or situational factors. The project may be to fulfill an external or internal contract. This profile will draw on sections from the following standards:

ISO/IEC 12207:2008 – Information Technology – Software life cycle processes.

ISO/IEC 15289:2006 – Software Engineering – Software life cycle processes – Guidelines for the content of software life cycle process information products (Documentation).

The preparation of the Basic VSE Profile follows these five steps, as illustrated in Figure 6-7:

[59] Adapted from ISO 2009a, ISO 2009d, and ISO 2009e.

- The recognition of VSE characteristics related to: finance, resources, customer interface, internal business processes, learning, and growth.
- The identification of VSE needs and suggested competencies that derive from those characteristics.
- The specification of the Basic VSE Profile elements appropriate to responding to the VSE needs and suggested competencies, according to the ISP 29110-2 VSE Framework and Taxonomy (described below).
- The selection and linking of the subset of ISO/IEC 12207 process and outcome elements and ISO/IEC 15289 product elements related to the Basic VSE Profile elements.
- The definition of the Basic VSE Profile Guides: TR 29110-5.1 Management and Engineering Guide for the Implementation of the Basic VSE Profile (described below).

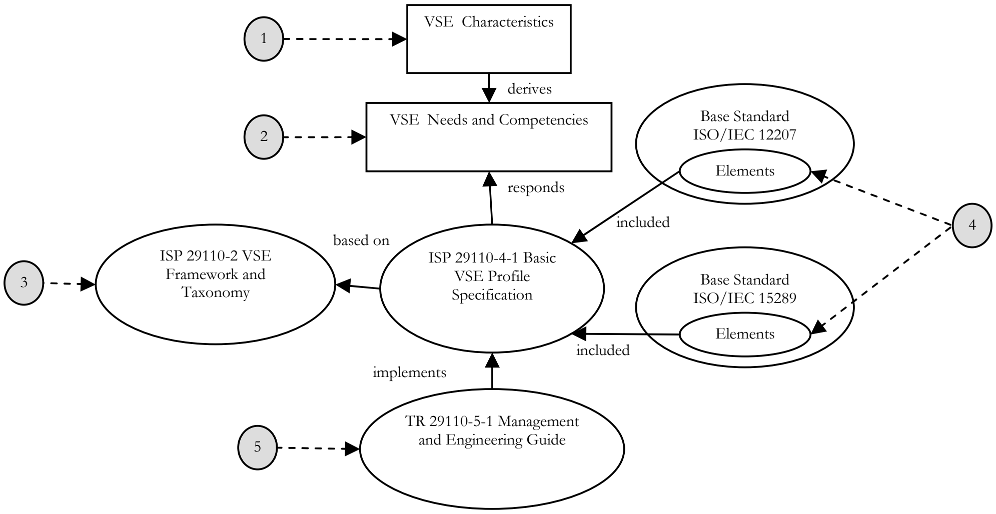

1
VSE Characteristics
derives
2
VSE Needs and Competencies
responds
Base Standard
ISO/IEC 12207
Elements
4
included
based on
3
ISP 29110-2 VSE Framework and Taxonomy
ISP 29110-4-1 Basic VSE Profile Specification
Base Standard
ISO/IEC 15289
Elements
included
implements
5
TR 29110-5-1 Management and Engineering Guide

Figure 6-7 Basic VSE Profile Preparation Steps (ISO 2009e)

The diagram is to be interpreted as follows: rectangles represent the VSE elements; ellipses represent standards or subsets of their elements; solid arrows show a labeled relation; and circles with a dashed arrow indicate the number of the preparation step.

In the next section, the author describes the set of 29110 documents developed for the GPG and the Basic Profile. Then, a detailed description of the Basic Profile is presented.

6.5 Description of the Set of ISO/IEC 29110 documents targeted by audience[60]

A set of documents, targeted by audience, has been developed to improve product, service quality, and process performance (see Figure 6-8). These documents include Part 1: Overview; Part 2: Framework and Taxonomy; Part 3: Assessment Guide; Part 4: Profile Specifications; and Part 5: Management and Engineering Guides. Parts 1 and 5 target VSEs, Part 3 targets assessors and VSEs; and Parts 2 and 4 target standards producers, tool vendors, and methodology vendors. If a new profile is needed, Parts 4 and 5 can be developed or tailored from the existing Parts 4 and 5 without impacting the other documents, becoming Part 4-x and Part 5-x respectively through the ISO/IEC development process, as explained in a previous chapter.

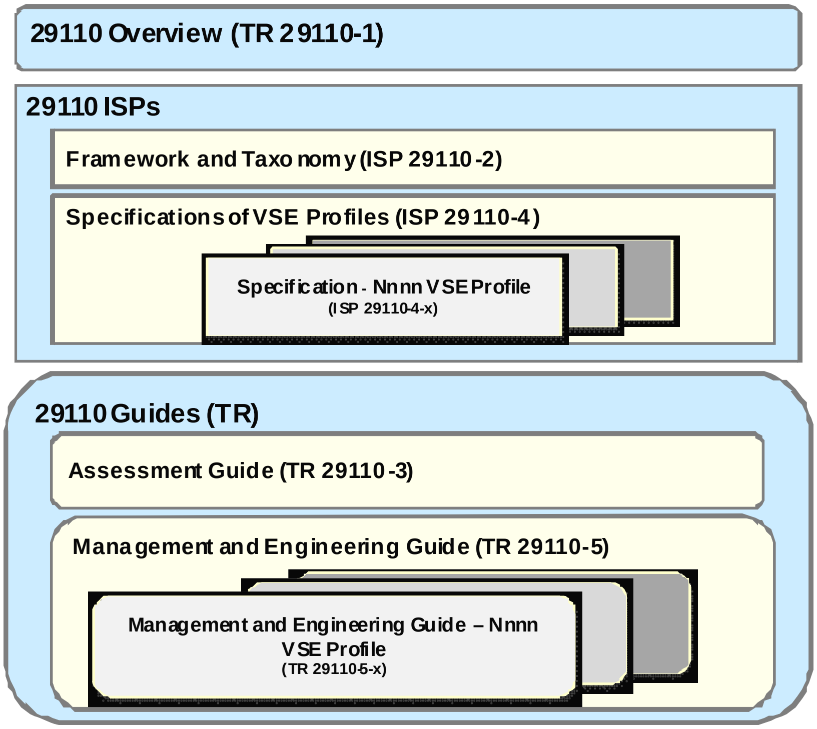

29110 Overview (TR 29110-1)
29110 ISPs
Framework and Taxonomy (ISP 29110-2)
Specifications of VSE Profiles (ISP 29110-4)
Specification - Nnnn VSE Profile
(ISP 29110-4-x)
29110 Guides (TR)
Assessment Guide (TR 29110-3)
Management and Engineering Guide (TR 29110-5)
Management and Engineering Guide – Nnnn VSE Profile
(TR 29110-5-x)

Figure 6-8 Set of ISO/IEC 29110 documents targeted by audience (ISO 2009a)

6.5.1 The overview document

The first document, titled Overview, is an ISO technical report. It introduces the major concepts required to understand and use the suite of documents. It introduces the business aspects, characteristics, and requirements of VSEs, and clarifies the rationale for VSE-specific profiles, documents, standards, and guides. It also introduces life cycle concepts, improvement, capability and assessment concepts, standardization concepts, and the 29110 set of documents. It targets both a general audience interested in the set of documents and, more specifically, users of the set of documents. The Overview is identified as a technical

[60] Adapted from ISO 2009d

report (TR) as follows: ISO/IEC 29110 – Life cycle profiles for Very Small Entities (VSEs) – Part 1: Overview. This document can be found in Volume III.

6.5.2 The International Standardized Profile documents

The second set of documents, titled International Standardized Profile (ISP), consists of ISO standards. It was defined to formally package references to other documents or parts of documents in order to adapt them to VSE needs and characteristics. Preparing profiles is an ISO/IEC JTC1 process involving the production of two types of documents, a framework and taxonomy, and a profile specification:

- **Framework and Taxonomy** – The Framework and Taxonomy document establishes the logic behind the definition and application of profiles. It specifies the elements common to all profiles (structure, conformance, assessment) and introduces the taxonomy (catalog) of 29110 profiles. It targets authors and reviewers of ISPs, authors of other document parts, as well as the authors of other VSE-targeted profiles. The Framework and Taxonomy is applicable to all profiles and is identified as follows: ISO/IEC 29110 – Life cycle profiles for Very Small Entities (VSEs) – Part 2: Framework and Taxonomy. This document can be found in Volume III.

- **Profile Specifications** – There is a profile specification document for each profile. Its purpose is to provide the definitive composition of a profile, provide normative links to the normative subset of standards (e.g. ISO/IEC 12207) used in the profile, and provide informative links (references) to input documents (e.g. ISO/IEC 90003, SWEBOK, PMI). It targets the authors/providers of guides and the authors/providers of tools and other support material. There is one profile specification document for each profile, identified as 29110-4.X, where x is the number assigned to the profile. The profile specification for the Basic Profile is identified as follows: ISO/IEC 29110 – Life cycle profiles for Very Small Entities (VSEs) – Part 4-1: Specification – Basic VSE Profile. This document can be found in Volume III.

6.5.3 The Assessment, and Engineering and Management Guides

The third set of documents, titled Guides, are ISO TRs. They contain implementation guidelines (domain-specific) on how to perform the processes to achieve the maturity levels (e.g. recommended activities, measures, techniques, templates, models, methods). Guides are developed for process implementation and for assessment based on the domain's issues, business practices, and risks. Guides target VSEs, and should be VSE-accessible, both in terms of style and cost. There are two guides, an assessment guide and a management and engineering guide:

- **Assessment Guide** – This guide describes the process to follow to perform an assessment to determine process capabilities and organizational process maturity, that is, when an organization wants an assessment carried out in order to obtain a process capability profile of the processes implemented, as well as an organizational process maturity level assigned to it. It is also applicable to the situation where a customer asks for a third-party assessment in order to obtain a capability level profile of the process implemented by the software development and maintenance provider. It is also suitable for self-assessment. The Assessment Guide is applicable to all profiles and is identified as follows: ISO/IEC 29110 – Life cycle profiles for Very Small Entities (VSEs) – Part 3: Assessment Guide. This document can be found in Volume III.

- **Management and Engineering Guide** – The management and engineering guides provide guidance on process implementation and the use of a profile. It targets VSEs (management and technical staff), VSE-related organizations (technology transfer centers, government industry ministries, national standards, consortia and associations, academic use for training, authors of derived products (software, courseware, and acquirers and suppliers. There is one management and engineering guide document for each profile, identified as 29110-5.X, where x is the number assigned to the profile. This number matches the number assigned to the profile specification. The management and engineering guide for the Basic Profile is identified as follows: ISO/IEC 29110 – Life cycle profiles for Very Small Entities (VSEs) – Part 5-1: Management and Engineering Guide – Basic VSE Profile. This document can be found in Volume III.

The final step of the approach consists of defining guidelines explaining the processes defined in the profile in more detail. At the Moscow meeting, the author proposed the development of a series of deployment packages as guidelines. A deployment package (DP) is a set of artifacts developed to facilitate the implementation of a set of practices of the selected framework in a VSE. The table of contents of a DP is illustrated in Figure 6-9. DPs are explained in more detail in the next chapter.

1. Introduction
   Purpose of this document
   Key definitions
2. Why this topic is important
3. Overview of the main tasks
   3.1 Tasks
   3.2 Roles and artefacts
   3.3 Activity life cycle and examples of lifecycles

Appendix A Templates
Appendix B Checklists
Appendix C Coverage Matrices (ISO 12207, ISO 9001, CMMI)
Appendix D Tools
Appendix E Training Material
Appendix F Deployment Package Evaluation Form

Figure 6-9 Table of contents of a deployment package

6.6 Detailed description of the Basic Profile[61]

VSEs are subject to a number of characteristics, needs, and desirable competencies that affect the content, nature, and extent of their activities. The Basic VSE Profile addresses the VSEs that are described through the following characteristics, needs, and desirable competencies, classified into four categories: Finance and Resources, Customer Interface, Internal Business Processes, and Learning and Growth. The four categories and their needs and desirable competencies are as follows:

Finance and Resources characteristics

- Small number of engineers (e.g. the cost of a payroll up to 25 people)
- Potential for short-term cash flow problems
- Low-budget projects, which last a few months and involve only a few people developing small products
- Dependent on successful project completion within schedule and budget
- Preference for separate projects to perform corrective post delivery maintenance
- Limited internal resources to perform management support and organizational processes like: risk management, training, quality management, process improvement, and reuse.

Needs and desirable competencies of the Finance and Resources characteristics:

- Projects carried out within budget and the product delivered on schedule

[61] This section is adapted from (ISO 2009e)

- Close communication maintained with the customer to manage risks

Customer Interface characteristics:

- Usually one customer per project at a time
- Customer satisfaction dependent on:
  - fulfillment of specific requirements that may change during the project;
  - information received in a timely fashion during product development;
  - delivery on schedule;
  - low level of defects found post-delivery; and
  - close communication and prompt response to any changes.
- Quantitative quality requirements not usually defined by customers
- A VSE usually not in charge of the management of the system, or of software integration, installation, or operation

Needs and desirable competencies for the customer interface characteristics:

- Fulfillment of customer requirements
- Management of changes to customer requirements during the project
- Provision of close communication and timely update information to the customer during product development
- Delivery of the product with a low level of defects

Internal Business Process characteristics

- The main process is designed to develop custom software systems written in-house on contract.
- The software product is elaborated progressively and has to be consistent with customer requirements.
- Products are developed or maintained through projects with a single line of communication between implementation group and customer.
- There is a small number of engineers (e.g. up to 25 people) in the organization, and therefore most of the communication, decision making, and problem resolution can be performed promptly, face-to-face.

- VSEs have lean project management budgets and conduct focused software implementation activities.
- The Infrastructure Management, Project Portfolio Management, and Human Resources Management Processes are performed through informal, face-to-face mechanisms.
- Products generated in projects are software items which may have more than one version and have to be saved and controlled.

Needs and desirable competencies for the Internal Business characteristics:

- Version control and storage of the products generated during a project
- Progressive elaboration of the software product, achieving consistency with customer requirements

Learning and Growth characteristics:

- Awareness of the importance of standards
- Lack of human resources to engage in standardization
- Lack of information of ISO/IEC standards
- Lack of knowledge of software process improvement and process evaluation

Needs and desirable competencies for the Learning and Growth characteristics:

- Guidelines which are flexible and easy to use for beginners on the adoption of practices of international standards focused on processes to support their software development projects needs

To use the Basic VSE Profile, it is assumed that the VSE fulfills the following entry conditions:

- Project contract or agreement with a statement of work
- Cost, technical, and schedule feasibility assessments performed before the start of the project
- Project work team, including project manager, assigned and trained
- Goods and infrastructure services available

Now that the characteristics of the Basic Profile have been defined, the next step consists of identifying the process elements, from the base standards, that will be used to define the

Basic Profile specifications. To illustrate this step, Figure 6-10 lists the needs and suggested competencies derived from the finance and resources characteristics. In this figure, the acronym PM stands for project management and the acronym SI stands for software implementation.

Perform the projects within budget and deliver the product on schedule. To respond to this need and suggested competencies, Basic VSE Profile processes, objectives, and work products are the following:

Project Management Process
PM.O1. The Project Plan for the execution of the project is developed according to the Statement of Work and validated with the Customer. The tasks and resources necessary to complete the work are sized and estimated.

PM.O2. Progress of the project is monitored against the Project Plan and recorded in the Progress Status Record. Corrections to remedy problems and deviations from the plan are undertaken when project targets are not achieved. Appropriate treatment is applied to correct or avoid the impact of risk. Closure of the project is performed to obtain Customer acceptance, which is documented in the Acceptance Record.

Software Implementation Process
SI.O1. Tasks of the activities are performed through the accomplishment of the current Project Plan.
Work Products: Statement of Work, Progress Status Record, Project Plan, Correction Register, and Acceptance Record.

Maintain close communication with the customer to manage risks. To respond to this need and suggested Competencies, the Basic VSE Profile processes, objectives, and work products are the following:

Project Management Process
PM.O5. Risks are identified as they develop and during the conduct of the project.
Work Product: Project Plan.

Figure 6-10 Needs and suggested competencies derived from finance and resources characteristics (ISO 2009d)

Similar lists have been developed for the other characteristics, e.g. customer interface characteristics. The final list of the Basic Profile VSE elements is composed of PM and SI processes with their corresponding objectives (identified as PM.O1, PM.O2, etc.; and SI.O1, SI.O2, etc.), and the work products are listed below. The Project Management Objectives are listed in Figure 6-11.

PM.O1. The Project Plan for the execution of the project is developed according to the Statement of Work and validated with the Customer. The tasks and resources necessary to complete the work are sized and estimated.
PM.O2. Progress of the project is monitored against the Project Plan and recorded in the Progress Status Record. Corrections to remedy problems and deviations from the plan are undertaken when project targets are not achieved. Appropriate treatment is applied to correct or avoid the impact of risk. Closure of the project is performed to obtain Customer acceptance documented in the Acceptance Record.
PM.O3. Change Requests are addressed through their reception and analysis. Changes to the software requirements are evaluated for cost, schedule, and technical impact.
PM.O4. Review meetings with the Work Team and the Customer are held. Agreements are registered and tracked.
PM.O5. Risks are identified as they develop and during the project.
PM.O6. A software Version Control Strategy is developed. Items of Software Configuration are identified, defined, and baselined. Modifications and releases of the items are controlled and made available to the Customer and Work Team, including the storage, handling, and delivery of the items.

PM.O7. Software Quality Assurance is performed to provide assurance that work products and processes comply with the Project Plan and Requirements Specification.

Figure 6-11 Project management objectives

The Project management Work Products are listed in Figure 6-12.

- Statement of Work,
- Progress Status Record,
- Project Plan,
- Change Requests,
- Meeting Record,
- Correction Register,
- Verification Results,
- Validation Results Project Repository,
- Project Repository Backup
- Acceptance Record.

Figure 6-12 Project Management Work Products

The Software Implementation Objectives are listed in Figure 6-13.

- SI.O1. Tasks of the activities are performed through the accomplishment of the current Project Plan.
- SI.O2. Software requirements are defined, analyzed for correctness and testability, approved by the Customer, baselined, and communicated.
- SI.O3. Software architectural and detailed design are developed and baselined. Their software items and internal and external interfaces are described. Consistency and traceability to software requirements are established.
- SI.O4. Software components defined by the design are produced. Unit tests are defined and performed to verify their consistency with requirements and the design. Traceability to the requirements and design are established.
- SI.O5. Software is produced integrating software components and verified using Test Cases and Test Procedures. Results are recorded on the Test Report. Defects are corrected and consistency and traceability to Software Design are established.
- SI.O6. A Software Configuration that meets the Requirements Specifications as agreed with the Customer, which includes user, operation, and maintenance documentation, is integrated, baselined, and stored in the Project Repository. The need for change to the Software Configuration is detected and related Change Requests are initiated.
- SI.O7. Verification and Validation tasks of all required work products are performed using the defined criteria to achieve consistency among output and input products in each activity. Defects are identified and corrected; records are stored in Verification/Validation Results.

Figure 6-13 Software Implementation Objectives

The Software Implementation Work Products are listed in Figure 6-14.

- Requirements Specification,
- Components,
- Test Report,
- Maintenance Documentation,
- Product Operation Guide, Software,
- Software Configuration,
- Software Design,
- Software User Documentation,
- Test Cases and Test Procedures,
- Traceability Record,
- Verification Results and Validation Results.

Figure 6-14 Software Implementation Work Products

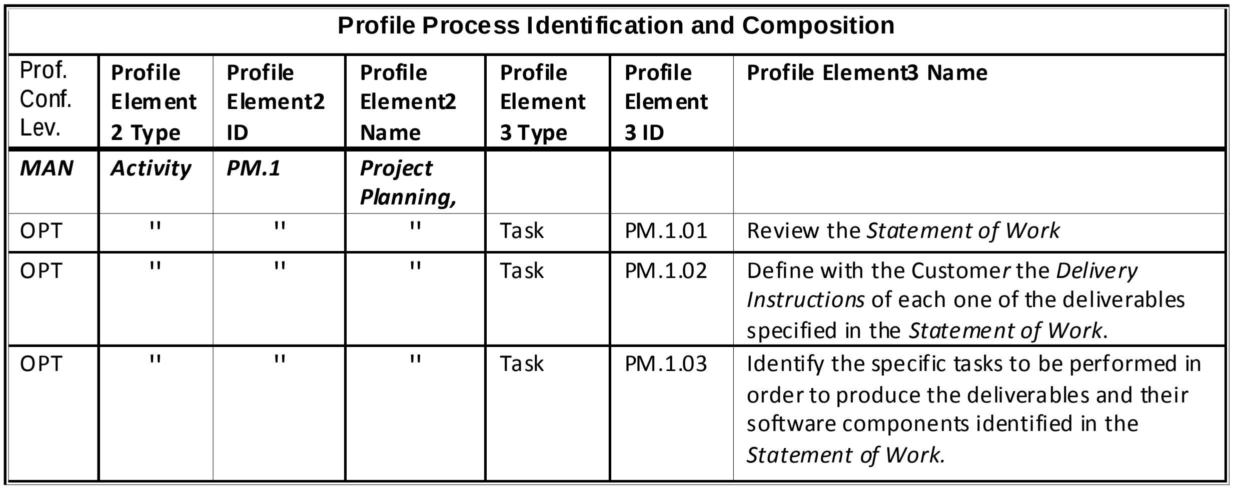
6.6.1 Basic VSE Profile specifications[62]

The Basic Profile specifications are a set of tables containing the information described in section 1.3.1. The following tables are defined in clause 7 of 29110 Part 4:

- A table for Process Specifications containing the following information:
    - Process definition and composition specification:
        - All processes are mandatory.
        - All activities are mandatory.
        - All tasks are optional.

For illustration purposes, a subset of the table listing the Project Management Tasks is reproduced in Table 6-2.

[62] Adapted from ISO 2009d



Profile Process Identification and Composition

Prof. Conf. Lev.	**Profile Element 2 Type**	**Profile Element2 ID**	**Profile Element2 Name**	**Profile Element 3 Type**	**Profile Element 3 ID**	**Profile Element3 Name**
MAN	***Activity***	***PM.1***	***Project Planning,***			
OPT	"	"	"	Task	PM.1.01	Review the *Statement of Work*
OPT	"	"	"	Task	PM.1.02	Define with the Customer the *Delivery Instructions* of each one of the deliverables specified in the *Statement of Work*.
OPT	"	"	"	Task	PM.1.03	Identify the specific tasks to be performed in order to produce the deliverables and their software components identified in the *Statement of Work.*

Table 6-2 Subset of the Table Listing the Project Management Tasks

- A table for Objective Specifications containing the following information:
  - Process objective specifications:
    - All objectives are required.
- A table for Work Product Specifications containing the following information:
  - Work product specifications:
    - All work products are required.
- A table for Input/Output Specifications containing the following information:
  - Activity input and output specifications:
    - All outputs are required.
    - All inputs are optional.

6.6.2 Basic VSE Profile base document references

The last step in the specification of a profile is the development of a set of tables referencing the base standard selected for the Basic Profile, the ISO/IEC 12207 and ISO/IEC 15289 standards. Table 6-3 lists the source elements from ISO 12207 which have been used as project management activities for this profile.

Source Element from ISO 12207	**Project Management Activities**
6.3.1	Project Planning
6.3.2	Project Assessment and Control
6.3.4	Risk Management
6.3.7	Measurement
6.4.8	Software Acceptance Support

Source Element from ISO 12207	Project Management Activities
7.1.2	Software Requirements Analysis
7.2.2	Software Configuration Management
7.2.3	Software Quality Assurance
7.2.6	Software Review
7.2.8	Software Problem Resolution

Table 6-3 Source Elements from ISO/IEC 12207 for the Project Management Activities (ISO 2009e)

For illustration purposes, one table of 29110 Part 4, for Project Management activities, is presented in Table 6-4.

Profile Process Identification and Composition			Process Source Document References							
Profile Element2 Type	Profile Elemen t2 ID	Profile Element2 Name	Source Conf. Level	Source Doc. ID	Source Element1 Type	Source Element1 ID	Source Element1 Name	Source Element2 Type	Source Element2 ID	Source Element2 Name
Activity	PM.1	Project Planning,	OPT	ISO/IEC 12207:2008	Process	6.3.1	Project Planning	Activity	6.3.1.3.2	Project Planning

Table 6-4 References for the Project Management Activities

6.6.3 Description of processes documented in 29110 Part 5

ISO/IEC 29110 Part 5-1, the Engineering and Management Guide for the Basic Profile (ISO 2009e), provides Project Management (PM) and Software Implementation (SI) processes at a level of detail that should allow a VSE to be able to translate this information into usable processes. The high-level relationship between the SI process and the PM process is illustrated in Figure 6-15.

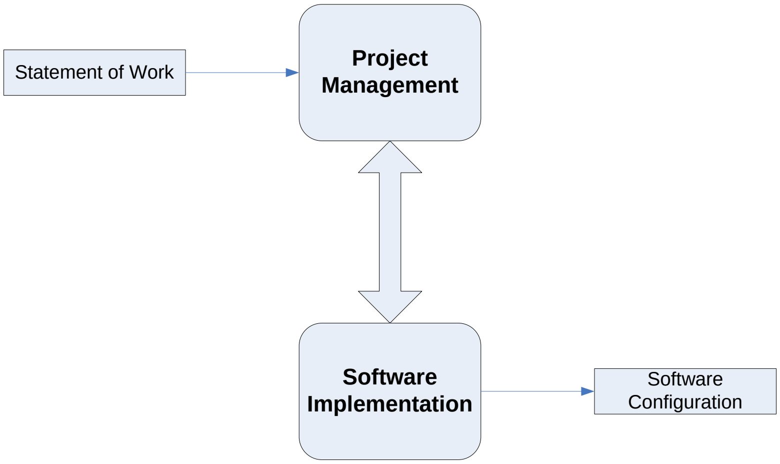


Figure 6-15 Basic Profile process relationship (ISO 2009e)

The PM process uses the customer's statement of work to elaborate the project plan. The PM project assessment and control tasks compare the project progress against the project plan, and action is taken to eliminate deviations from the project plan, or incorporate changes into it. The PM project closure activity delivers the software configuration, produced by SI, and obtains the customer's acceptance to formalize the end of the project. A project repository is established to save the work products and to control its versions during the project.

The execution of the SI process is driven by the project plan. The SI process starts with an initiation activity of the project plan review. The project plan will guide the execution of the software requirements analysis, software architectural and detailed design, software construction, and software integration and test, and product delivery activities. To remove a product's defects, verification, validation, and test tasks are included in the activities workflow.

The customer provides a statement of work as an input to PM process and receives a software configuration as a result of SI process execution.

6.6.4 Description of the Project Management process

The purpose of the Project Management process is to establish and carry out the tasks of the software implementation project in a systematic way, which allows compliance with the project's objectives in terms of expected quality, time, and costs (see Figure 6-16).

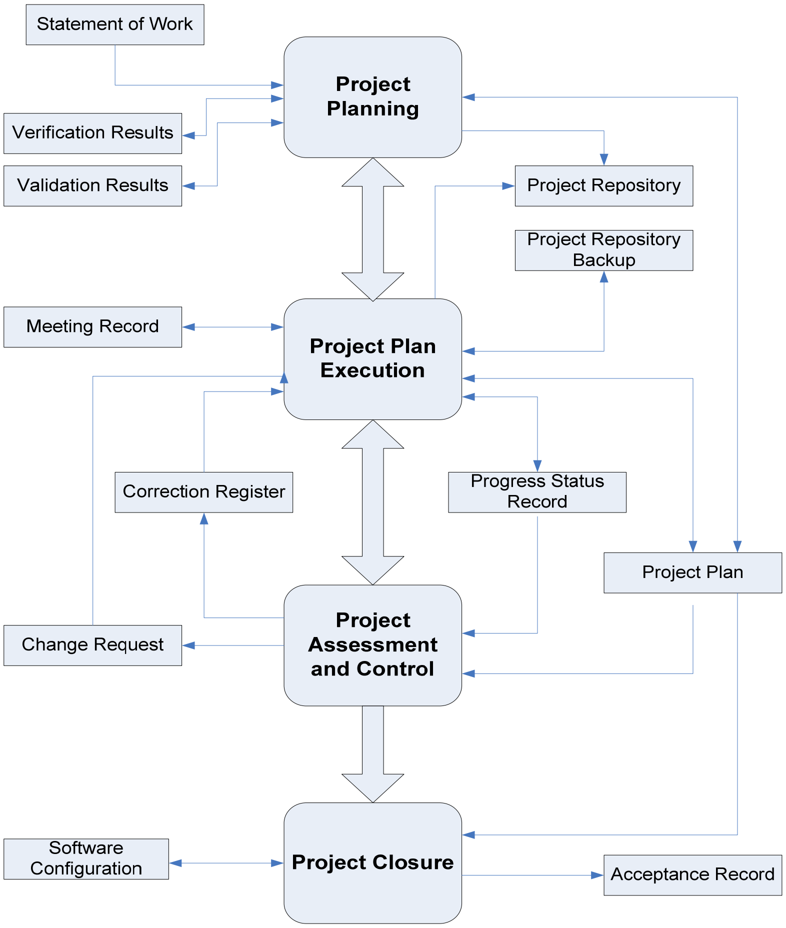


Figure 6-16 Project Management process diagram (ISO 2009e)

6.6.5 Description of the Software Implementation process

The purpose of the Software Implementation process is to achieve systematic performance of the analysis, design, construction, integration, and test activities for new or modified software products according to the specified requirements (see Figure 6-17).

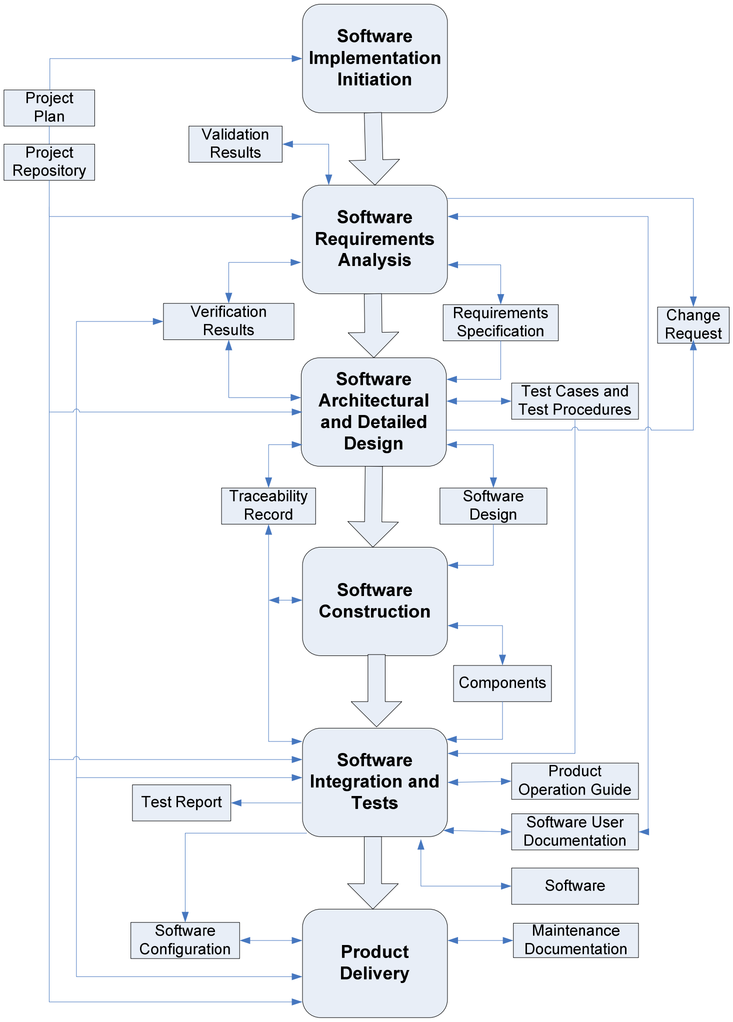

Software Implementation Initiation
Project Plan
Project Repository
Validation Results
Software Requirements Analysis
Verification Results
Requirements Specification
Change Request
Software Architectural and Detailed Design
Test Cases and Test Procedures
Traceability Record
Software Design
Software Construction
Components
Software Integration and Tests
Product Operation Guide
Test Report
Software User Documentation
Software
Software Configuration
Product Delivery
Maintenance Documentation

Figure 6-17 Software Implementation process diagram (ISO 2009e)

For illustration purposes, the tasks of the software requirements analysis activity are listed in Table 6-5.

Identification of Task	Description of Tasks
SI.2.1	Assign tasks to the work team members in accordance with their role, based on the current Project Plan.
SI.2.2	Document or update the Requirements Specifications.
SI.2.3	Verify the Requirements Specifications.
SI.2.4	Validate the Requirements Specifications.
SI.2.5	Document the preliminary version of the Software User Documentation or update the present manual.
SI.2.6	Verify the Software User Documentation.
SI.2.7	Incorporate the Requirements Specifications and Software User Documentation into the Software Configuration in the baseline.

Table 6-5 Software Requirements Analysis Tasks (ISO 2009e)

The description of the Analyst role is presented in Table 6-6.

	Role	Abbreviation	Competency
1.	Analyst	AN	Knowledge and experience in eliciting, specifying, and analyzing the requirements. Knowledge in designing user interfaces and ergonomic criteria. Knowledge of revision techniques and experience in software development and maintenance. Knowledge of editing techniques and experience in software development and maintenance.

Table 6-6 Description of the Analyst's Role

Table 6-7 illustrates the definition of a work product: the Change Request.

	Name	Description	Source
1.	Change Request	It may have the following characteristics: – Identifies the purpose of the change – Identifies the request status (new, accepted, rejected) – Identifies the requester's contact information – Identified the impacted system(s) – Impact on operations of existing system(s) defined – Impact of associated documentation defined – Criticality of the request, deadline The applicable statuses are: initiated, evaluated, and accepted.	Software Implementation Customer Project Management

Table 6-7 Description of the Change Request work product

6.7 Description of the Entry Profile[63]

At the Berlin meeting, the delegates from Belgium and Canada proposed the elaboration of a profile targeting start-up VSEs and short-intensity projects of about 6 person-months of effort. At the Mexico meeting, the delegate from Canada presented a set of practices that could be embedded in this future profile. After discussion, the members of the working group agreed to the practices listed in Figure 6-18.

[63] This profile will be discussed at the meetings in Peru in November 2009 and in Japan in May 2010.

Project Planning and Monitoring

Develop an agreement with customer

Develop a plan

1. Determine phases, tasks, milestones, deliverables
2. Assess available resources, estimate effort
3. Monitor project status and perform reviews (e.g. retrospective)
   Collect data (e.g. effort spent on tasks)

Software Development

Requirement Analysis and Design

1. Identify the set of requirements to implement,
2. Plan interactions with customer
3. Design the software

Software Code and Test

1. Code and debug
2. Perform unit and system testing

Figure 6-18 Proposed Practices for the Entry Profile

Although a consensus on the practices targeting VSEs was reached during the meeting, the working group has to formally define characteristics, needs, and desirable competencies, as was done for the Basic Profile,. The following section presents a list of characteristics that will be used to prepare the ISP.

6.7.1 Characteristics of start-up VSEs

Essentially, start-up software companies are organizations without an established product, customer base, or revenue stream. They have been neglected in process studies, and, because they lack the foundation required for success with the CMM, it can be problematic for them to strictly adhere to it (adapted from Sutton 2000).

To better define this profile, the author developed a set of characteristics of start-up VSEs. Nambisan describes start-up VSEs as "entrepreneur-driven [entities that] design, implement, or maintain complex, high technology products or services. The capabilities of start-up technology companies and their products are largely untested in the markets." They have the following characteristics (adapted from Nambisan 2002):

- Primarily involved in the design and/or coding of minor software packages, as minor software products involve incremental innovation;
- Carry out minor tasks or complement larger established software products;
- Tend to be small units with limited financial resources or product vision to develop and market their own products;
- Do not have significant experience with large software development projects, and so do not attract contract jobs from larger software firms;

- Focus on managing their human resources judiciously and gaining valuable experience in product coding and testing.

Additional attributes of start-up enterprises are listed in Figure 6-19 (adapted from (Ruokolainen et al. 2007), (Nambisan 2002), and (Sutton 2000)).

- Small
- Personnel relatively young and inexperienced
- Focus on outward-looking activities: getting the product out
- Entrepreneur-driven
- Technology-oriented
    - Since they are established to develop technologically innovative products, they require cutting edge tools and techniques.
    - Entrepreneurs tend to concentrate on solving technological problems at the expense of commercialization:
        - more than 40% of start-up companies have problems with marketing;
        - limited product vision to develop and market their own products.
- Lack of tangible assets
- Limited access to investments and loans
- Limited cash flow
- Focus on managing human resources judiciously and gaining valuable experience in product coding and testing (Nambisan 2002)
- No significant experience in large software development projects (Nambisan 2002)
    - They do not attract contract jobs from larger software firms.
- Lack discipline in product development tasks (Nambisan 2002)
- Most projects relatively less complex and of limited scope
- Lack of process rigor, although with no serious consequences, as the projects tend to be complex and highly resource-intensive (Nambisan 2002)
- Considerable process discipline imperative, to evolve from a start-up into an expert coder, as projects tend to be complex and highly resource-intensive. (Nambisan 2002)
- Process maturity models like the CMM have been largely responsible for the success of several Indian software start-ups in evolving into expert coders (Sukhathankar 1997).
- The attitude of top management towards process rigor and discipline is important, otherwise proper investments may not be made in implementing appropriate methodologies (Nambisan 2002).
- Successful evolution to a star status requires that other development tasks (e.g. version control, domain knowledge reuse) be performed, possibly along with the adoption of additional process control measures, which calls for investment in process discipline by both utility developers and expert coders (Nambisan 2002).
- Lack of credibility:
    - may engender scepticism in high-technology markets. concerning new players
    - represents a risk for a corporate customer
    - means that securing its first customer is a primary problem
    - results in the need to use its entrepreneur's social capital to find its first referral or to obtain venture capital
    - means that the main investment criteria used by venture capitalists related to the business experience and characteristics of the entrepreneur
    - results in the need for a reference (e.g. from a customer)
- Customer-driven
    - customer involvement criticized as limiting innovativeness in research and development:
        - The customers' proposals are often improvements to existing products rather than heralding radical change.
        - The strong involvement of the customer might put the start-up in the role of

subcontractor, rather than developing the product for a large market.

        - The customer may impose a software development process on the start-up.
    - existing customers are potential customers for new versions of a software product
- Problems with intellectual property rights
    - in some cases, the intellectual property rights of the start-up to the product not recognized by customers, who may claim to own those rights themselves
    - an interest in the intellectual property rights of the product on the part of its first commercial user, typically a firm, if, for example, the cost of implementation and development is not shared between them
- Difficulty intervening in existing buyer-seller relationships

Figure 6-19 Characteristics of start-up technology companies

The author proposed at the Mexico meeting that only one deployment package be developed to cover all the activities, tasks, and work products associated with this profile.

6.8 Description of the Intermediate and Advanced Profiles

At the upcoming meetings of the working group in Peru and in Japan, new profiles will be presented and discussed. Briefly, while the Basic Profile targeted VSEs having only one development project, the Intermediate Profile addresses VSEs carrying out more than one software project at a time. The Advanced Profile should target mature VSEs wishing to improve their business capabilities by adding business management and portfolio management practices.

6.9 Conclusion

This chapter presented the third phase of the innovation process, Development. The documents used by WG24 and the approach that led to the development of the International Standards (ISs) and Technical Reports (TRs) for VSEs were presented. The next chapter, titled Development of Means to Accelerate the Utilization and Adoption of International Standards by VSEs, presents the strategies developed to address the fourth and fifth phases of the innovation process: Commercialization, and Diffusion and Adoption.

Chapter 7

Development of a Means to Accelerate the Adoption and Utilization of International Standards by VSEs

7 Chapter 7 – Development of a Means to Accelerate the Adoption and Utilization of International Standards by VSEs

7.1 Introduction

As stated by Magee, 'when a standard is released, all of its "parents, relatives and friends" seem to walk away and hope the new standard will survive in the harsh environment of the real world' (Magee et al. 2004). The official mandate of ISO SC7 working groups does not include activities related to facilitating the deployment of an ISO standard or technical report. That mandate ends after the final version of a standard or a technical report has been submitted to the ISO for publication. Since, as stated in previous chapters, VSEs are not interested in software standards, the author proposed to the working group, at the Bangkok meeting in 2005, that a series of activities be planned to accelerate the deployment and utilization of international standards for VSEs.

At one meeting, the working group members decided to develop a series of requirements which would be used to develop a set of standards documents. At the Moscow meeting, the working group finalized the list of requirements for that set of documents (MOS-023 2007). Table 7-1 lists the requirements with respect to access to future international standards and technical reports, and to their deployment and utilization.

- R07 – Implementation of the set of workproducts (i.e. profiles, guides) must be affordable (very low initial cost of entry in terms of training).
- R08 – Use of the set of workproducts must be affordable and should be intuitive (i.e. consultant services should not be necessary).
- R15 – The set of workproducts should provide the whole spectrum of documents (from standards to educational material, i.e. tutorial information).
- R29 – The set of workproducts should propose examples of life cycles to facilitate choice.
- R33 – The set of workproducts should propose a definition of documents (as assemblies of workproducts). For example, templates (e.g. requirements templates – use cases)
- R37 – The set of workproducts should include compliance table checklists (assessment guide).
- R47 – The guide should include support for documentation, particularly templates.
- R52 – The guide should provide examples (e.g. plans, workproducts, and other deliverables).
- R56 – The guide should be compact (about 50 pages), excluding templates.
- R57 – The guide should be available free of charge on the Web

Table 7-7-1 Requirements for Facilitating the Adoption of ISO Standards for VSEs (adapted from MOS-023 2007)

As illustrated in Figure 7-1, the adoption rate for innovations varies considerably. It should be possible for the standards and technical reports developed by WG24 to achieve an adoption rate similar to that of wireless phones, i.e. to reach about 60% of the targeted VSEs within 15 years of their publication by the ISO in mid-2010.

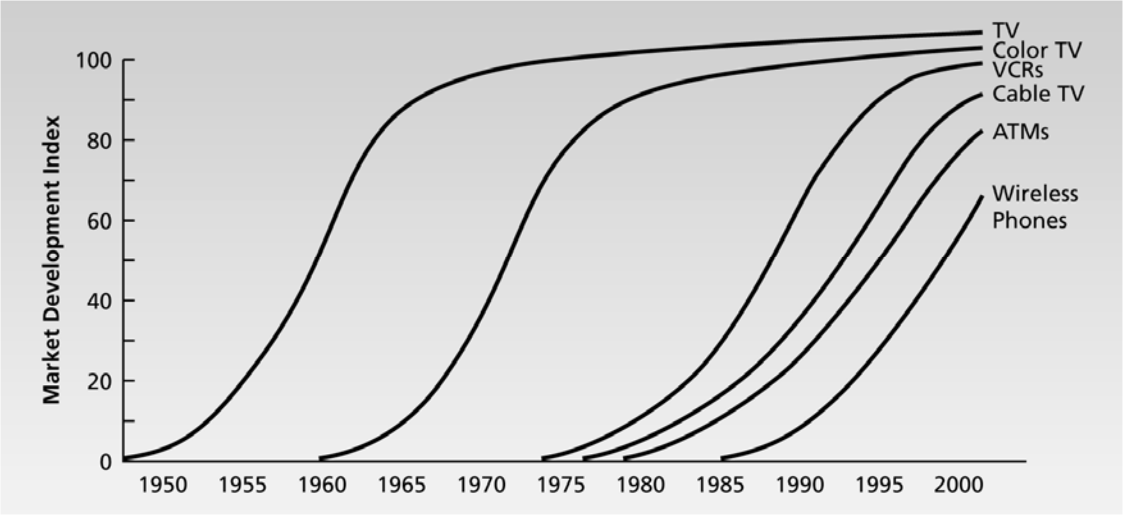


Figure 7-1 Various rates of technology diffusion (Rogers 2003)

Some research on diffusion has been conducted by the author since 2005, with the publication of papers and presentations at workshops and conferences in North America, Central America, South America, Asia, and Europe.

In previous chapters, the first three phases of the 6-phase innovation process illustrated in Figure 7-2 were discussed. In this chapter, the mechanisms developed to address the fourth and fifth phases of the innovation process are presented: the commercialization phase, and the diffusion and adoption phase. The last phase, consequences, will be presented in the next chapter.

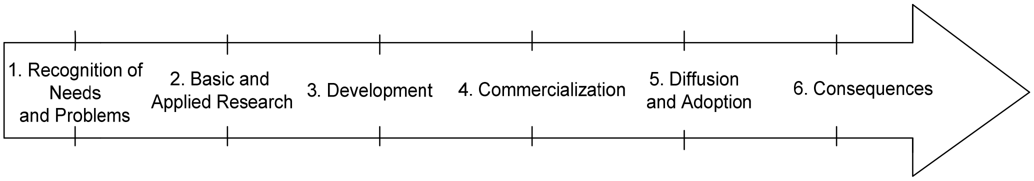

1. Recognition of Needs and Problems
2. Basic and Applied Research
3. Development
4. Commercialization
5. Diffusion and Adoption
6. Consequences

Figure 7-2 Phases of the innovation process (adapted from Rogers 2003)

This chapter presents the author's published works and contributions to the development of means to accelerate the adoption of ISO standards by VSEs, systems integrators, and governmental organizations: the development of a set of deployment packages, the conducting of pilot projects, the development of a process asset library and a Web site, the establishment of a network of support centers, and the creation of an education and training interest group.

Rogers defines commercialization as 'the production, manufacturing, packaging, marketing, and distribution of a product that embodies an innovation.' He defined diffusion as 'the process by which an innovation is communicated through certain channels over time among the members of a social system,' and adoption as 'a decision to make full use of an innovation as the best course of action available' (Rogers 2003).

7.2 The variables determining the rate of adoption of innovations

Rogers defines the rate of adoption as 'the relative speed with which an innovation is adopted by members of a social system. It is generally measured as the number of individuals who adopt a new idea in a specified period' (Rogers 2003). Figure 7-3 lists the variables, identified by Rogers, which affect the rate of adoption of an innovation.

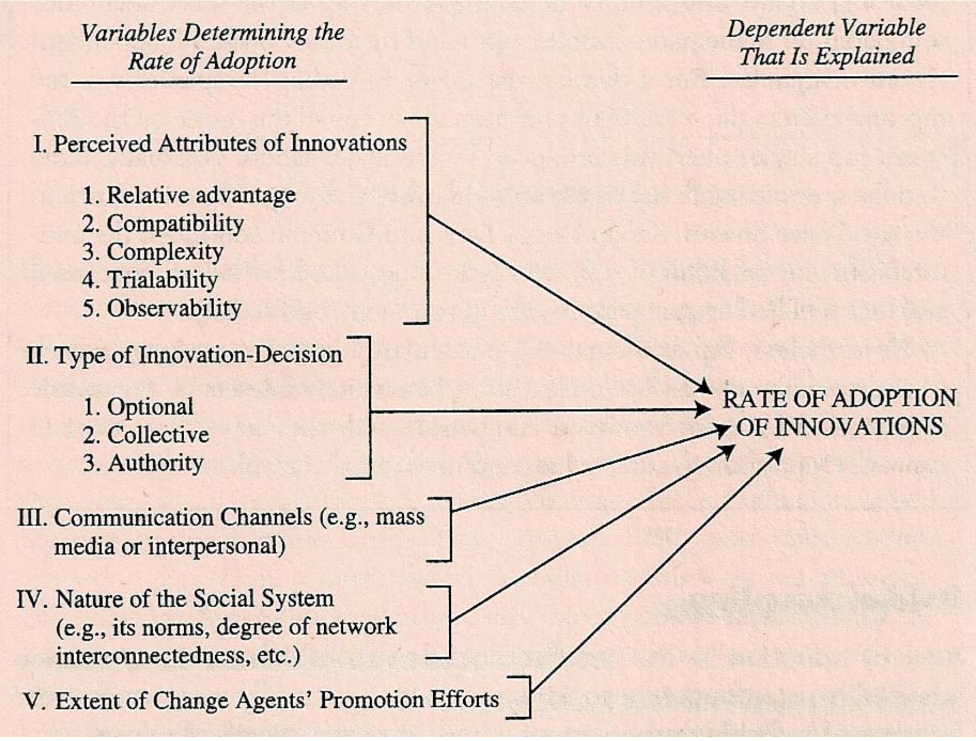

Variables Determining the Rate of Adoption
Dependent Variable That Is Explained
I. Perceived Attributes of Innovations
1. Relative advantage
2. Compatibility
3. Complexity
4. Trialability
5. Observability
II. Type of Innovation-Decision
1. Optional
2. Collective
3. Authority
III. Communication Channels (e.g., mass media or interpersonal)
IV. Nature of the Social System (e.g., its norms, degree of network interconnectedness, etc.)
V. Extent of Change Agents' Promotion Efforts
RATE OF ADOPTION OF INNOVATIONS

Figure 7-3 Variables determining the rate of adoption of innovations (Rogers 2003)

These variables affect the shape of the S curve illustrated in Figure 7-4. The mechanisms explained in this chapter are intended to accelerate the adoption of innovations, so that the adoption curve will look like the one at the far left (i.e. diffusion strategy X).

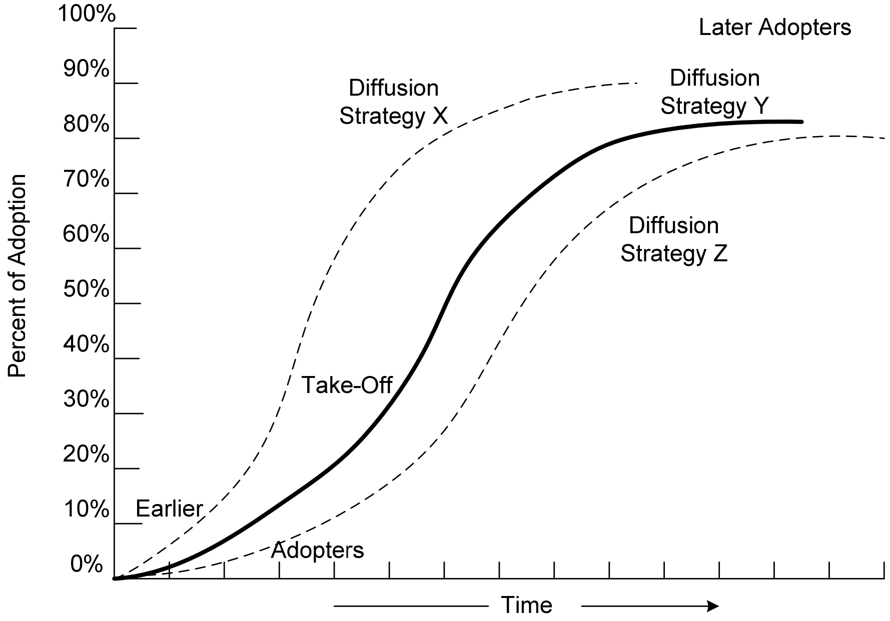


Figure 7-4 Diffusion rate of innovation (adapted from Rogers 2003)

Summarized in Table 7-2 are the various adoption mechanisms explained in more detail in this chapter to accelerate the adoption of future ISO standards and technical reports by VSEs, systems integrators, and governmental organizations. Briefly, the mechanisms presented in this chapter are: presentations (e.g. at conferences and workshops), publications, education (e.g. courses for undergraduate students), a Web site for VSEs, deployment packages, pilot projects, case studies, and a network of support centers for VSEs.

Category of Variable	**Name of Variable**	**Definition of Variable**	**Mechanisms Proposed by the Author**
Perceived Attributes of Innovations	Relative advantage	Degree to which an innovation is perceived as better than the idea it supersedes	Presentations Case studies Web site
	Compatibility	Degree to which an innovation is perceived as being consistent with the existing values, past experiences, and needs of potential adopters	Presentations
	Complexity	Degree to which an innovation is perceived as difficult to understand and use	Presentations Deployment packages Education
	Trialability	Degree to which an innovation may be experimented with on a limited basis	Pilot projects Education
	Observability	Degree to which the results of an innovation are visible to others	Pilot projects Presentations Case studies

Category of Variable	Name of Variable	Definition of Variable	Mechanisms Proposed by the Author
Type of Innovation Decision	Optional	Choice to adopt or reject an innovation made by an individual independent of the decisions of the other members of the system	Presentations
	Collective	Choice to adopt or reject an innovation made by consensus among the members of a system	Presentations
	Authoritative	Choice to adopt or reject an innovation made by relatively few individuals in a system who possess power, status, or technical expertise	Presentations
Communication Channels	Mass media, interpersonal channels, for example	The means by which messages are sent from one individual to another	Web site for VSEs Communications Network of Support Centers Education group Case Studies Survey
Nature of the Social System	Norms, for example	A set of interrelated units engaged in joint problem solving to accomplish a common goal	Presentations Education Pilot projects
Extent of Change Agent's Promotional Efforts		Individual who influences a client's innovation decisions in a direction deemed desirable by a change agency	Network of Support Centers Delegates to WG24 Pilot projects

Table 7-2 Variables and Mechanisms to Facilitate the Adoption of Innovations (adapted Rogers 2003)

7.3 Development of deployment packages[64]

The concept of a deployment package was introduced at the Spring 2007 Moscow meeting (MOS-027 2007) by the author, who defines it as a set of artifacts developed to facilitate the implementation of a set of practices, of the selected framework, in a VSE. However, a deployment package is not a complete process reference model, nor is it normative. The issues discussed at the meeting about the needs associated with developing deployment packages are listed in Table 7-3.

- To address the mandate of WG24
- To target micro-enterprises, i.e. fewer than 10 employees
- To provide micro-enterprises with ready-to-use ISO 12207-based technologies
- To use existing ISO/IEC standards
    - e.g. 15504
- To allow the development of deployment packages in parallel
    - i.e. CETIC, Tudor, ETS, and others

[64] This section is adapted from an article originally published in (Laporte et al. 2008d)

- To address the most pressing needs first
    - as identified by our contacts in the field
        - e.g. requirements, configuration, tests, etc..
- To pilot technologies (in the short term)
- To provide feedback to the developers of deployment packages
- To make deployment packages available on the Web
- To provide feedback to WG24

Table 7-3 Issues Discussed concerning Deployment Packages

At that meeting, proposed content was presented for discussion (see Table 7-4), with which the delegates agreed. The elements of a typical deployment package are: process description (e.g. activities, inputs, outputs, and roles), guide, template, checklist, example, presentation material, reference and mapping to standards and models, and list of tools. The mapping is only given as information to show that a deployment package has explicit links to standards, such as 12207 and 9000, or to models, such as the CMMI. Hence, by deploying and implementing the package, a VSE can see what concrete step it needs to take to achieve or demonstrate compliance with a standard or model. Packages are designed such that a VSE can implement its content without having to implement the complete framework at the same time.

1. Introduction
   Purpose of this document
   Key Definitions
2. Why this topic is important
3. Overview of the main tasks
   3.1 Tasks
   3.2 Roles and artefacts
   3.3 Activity life cycle and examples of life cycles

Appendix A Templates
Appendix B Checklists
Appendix C Coverage Matrices (ISO 29110 Part 5, ISO 12207, ISO 9001, CMMI)
Appendix D Tools
Appendix E Deployment Package Evaluation Form

Table 7-4 Table of Contents of a Deployment Package

In the paragraphs below, each component of a deployment package is described using the Requirements Analysis deployment package as an example.

At that meeting, Belgium and Canada proposed that an initial version of a deployment package on requirements analysis be developed for presentation at the next meeting. At the Fall 2007 Montreal meeting, the deployment package was presented for discussion. The delegates agreed that such a document would help VSEs implement the proposed standard. They also agreed on the list of deployment packages to be developed for the next meeting (see Figure 7-5): configuration management (Thailand), testing (Columbia), and project management (Ireland and Thailand). It was agreed that the deployment packages had to be in line with Part V, i.e. use the same terminology, and contain the same processes, tasks, roles, and work products.

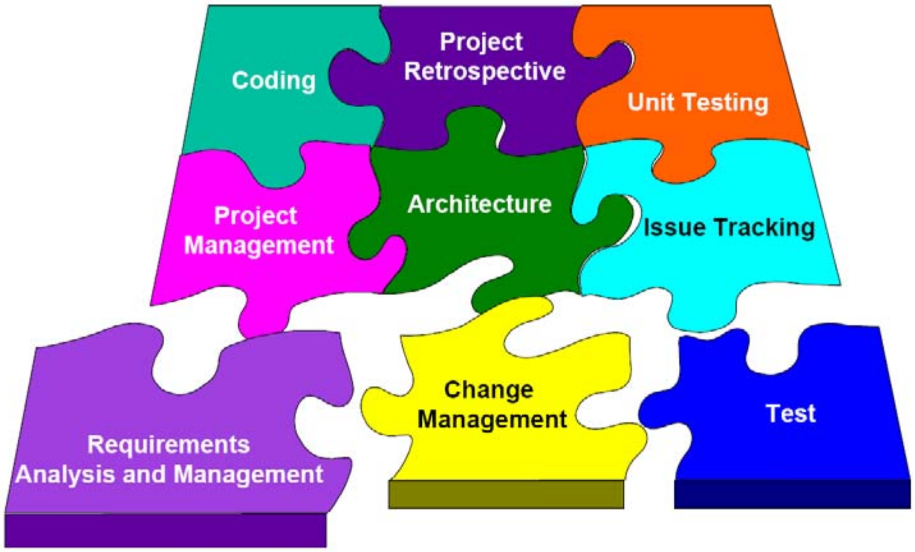


Figure 7-5 Initial list of deployment packages

7.4 Description of the Requirements Analysis deployment package

First component: Cover page

The cover page contains 3 sections: the title of the package, the copyright clause, and the names of the contributors. The first section provides the title of the deployment package and the profile and sub-profile addressed by the package.

Deployment Package – Software Requirements Analysis
For the Generic Profile – Basic Profile

Section 2 contains the copyright clause. It was decided that the deployment package would be offered at no cost to VSEs. The copyright clause is similar to those used by the developers of Open Source software.

Notes:

This document is the intellectual propriety of its author's organization. However, information contained in this document can be used free of charge. The distribution of all or parts of this document is authorized for non commercial use as long as the following legal notice is mentioned:

© Centre d'Excellence en Technologies de l'Information et de la Communication and École de technologie supérieure

Commercial use of this document is strictly forbidden. This document is distributed in order to enhance exchange of technical and scientific information.

This material is furnished on an "as is" basis. The author(s) make(s) no warranties of any kind, either expressed or implied, as to any matter including, but not limited to, warranty of fitness for purpose or merchantability, exclusivity, or results obtained from use of the material.

The processes described in this Deployment Package are not intended to preclude or discourage the use of additional processes that Very Small Entities may find useful.

Section 3 lists the author(s) of the deployment package and his (their) affiliation, the name of the individual who performed the quality assurance for the package, the dates of creation and last update, the status of the deployment package, and its version number.

Authors, organization, country	S. ALEXANDRE – Centre d'excellence en technologies de l'information et de la communication (CETIC), Belgium C. Y. LAPORTE – École de technologie supérieure (ÉTS), Canada
Quality assurance	B. Gill - SIE Center del Tecnológico de Monterrey, Mexico
Creation date	DD MM YYYY
Last update	DD MM-YYYY
Status	Not Inspected/Inspected
Version	1.2

Second component: Version and modification identification and list of acronyms and references

The second page contains 3 sections: a table describing the modifications made to the document, a list of acronyms, and a list of references used in the document.

The first section lists the modifications, the name of their author, the version number, and the date of the modifications.

Versions

Date	Version	Author	Modification
07/08/2007	0.1	S. ALEXANDRE	Document creation
20/08/2007	0.2	C.Y. LAPORTE	Comments on document structure
1/10/2007	0.3	S. ALEXANDRE	Implementation of comments and finalization of V1.0
1/10/2007	0.4	S. ALEXANDRE	Comments after review
8/10/2007	0.5	S. ALEXANDRE	Update and completion of section 3.1
14/10/2007	0.6	S. ALEXANDRE – C.Y. LAPORTE	Update and revision of section 3.1

Section 2 lists the abbreviations and acronyms used in the document.

Abbreviations/Acronyms

Abre./Acro.	Definition
VSE	Very Small Entities – enterprises, organizations, departments, or projects having up to 25 people

Section 3 lists the references made in the document:

References

Key	Reference
[OWPL-EN]	Renault A., Habra N., Alexandre S., Deprez J.-C., OWPL. Software Process Improvement for VSE, SME and low maturity enterprises. Version 1.2.2, FUNDP-CETIC, 2000. (http://www.cetic.be/internal393.html)
[IEEE830-98]	IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications, IEEE, 1998.
[ISO/IEC12119]	ISO/IEC 12119:1994 Information technology – Software packages – Quality requirements and testing.
[ISO/IEC12207]	ISO/IEC 12207:2008 Systems and software engineering – Software life cycle processes.
[ISO/IEC24765]	ISO/IEC 24765, Systems and Software Engineering Vocabulary
[ConstSoft02]	Construx Software – Checklist for Software Requirements Specifications, 2002. www.construx.com
[SELB07]	Selby, P., Selby, R.W., Measurement-Driven Systems Engineering Using Six Sigma Techniques to Improve Software Defect Detection, Proceedings of 17th International Symposium, INCOSE, June 2007, San Diego.
[STAN02]	Standish Group – Chaos report 2002. www.standishgroup.com/
[SPEM05]	Software Process Engineering Metamodel Specification, OMG, 2005.
[VOLE07]	Volere, Requirements Resources - http://www.volere.co.uk

Third component: Table of contents

The third page contains the table of contents of the deployment package, as illustrated below.

1 Introduction
1.1 Purpose of this document
1.2 Key definitions
2 Why this topic is important
3 Overview of tasks
4 Tasks
5 Roles and artifacts
6 Activity life cycle
Example of a practice life cycle
Appendix A – Templates
Appendix B – Checklist
Appendix C – Coverage Matrix to ISO Standards and CMMI Model
ISO/IEC 29110-5, Software Engineering – Life Cycle Profiles for Very Small Entities (VSEs) – Part 5-1: Management and Engineering Guide – Basic Profile
ISO 9001 Coverage Matrix
ISO/IEC 12207 Coverage Matrix
CMMI Coverage Matrix
Appendix D – Tool
Appendix E – Feedback Form

Table of Contents of a Deployment Package

Fourth component: Introduction and key definitions

The introduction section presents the purpose of the deployment package and explains the structure of the remaining sections of the document.

Purpose of this document
The purpose of this document, entitled "*Deployment Package – Software Requirements Analysis*", is to provide Very Small Entities (VSEs) with tailorable and easily usable guidelines and materials in order to implement a good customer requirements management and analysis process in their company.

In IT projects, it is critical to define the customer requirements as unambiguously as possible to ensure a common understanding of requirements among stakeholders, and to guarantee that requirements evolution is handled as part of the project.

The Requirements Analysis process includes the production and maintenance of Software Requirements Specifications on the basis of customer demands and changes in these demands. Software Requirements Specifications will then constitute the basis for cost estimates, planning, implementation, and tracking of activities throughout the project.

Requirement management is one of the principal parameters for process stabilization and successful repeatability.

A deployment package is a set of artefacts developed to facilitate the implementation of a set of practices, of a framework, in a VSE. A deployment package is not a process reference model (i.e. it is not prescriptive). Deployment package task descriptions aim to facilitate the implementation of process reference models like ISO/IEC12207. The elements of a typical deployment package are: process description (i.e. activities, inputs, outputs, roles, etc.), references to recognized standards and models, guide, template, checklist, example, presentation material, and list of tools. Examples of deployment packages are: requirement management and analysis, version control, change management, and testing.

This document supports Profile 1 as defined in ISO/IEC ISP 29110, and consists of the following parts, under the general title Software Engineering — Life Cycle Profiles for Very Small Enterprises (VSEs), Part 5.1: Management and Engineering Guide.

The content of this document is entirely *informative*.

Key definitions are provided to make this document as complete as possible for the reader. They have been extracted from current standards. The source of the reference is indicated in brackets.

Key Definitions

Requirements Analysis: The process of studying user needs to arrive at a definition of system, hardware, or software requirements. [ISO/IEC 24765]

Requirements document: a document containing any combination of recommendations, requirements or regulations to be met by a software package. [ISO/IEC 24765]

Requirements phase: the period of time in the software life cycle during which the requirements for a software product are defined and documented. [ISO/IEC 24765]

Software Requirements Specifications (SRS): The SRS constitutes a specification for a particular software product, program, or set of programs that performs certain functions in a specific environment. The SRS may be written by one or more representatives of the supplier, one or more representatives of the customer, or by both. [IEEE 830]

The SRS document contains both functional and non functional requirements.

The SRS can be presented in a Word document, but it can also be managed in a database or in an Excel file.

Requirement: *1.* a statement that identifies what a product or process must accomplish to produce required behavior and/or results. *IEEE 1220-2005 IEEE Standard for the Application and Management of the Systems Engineering Process.* 3.1.16. *2.* a system or software requirement that specifies a function that a system/software system or system/software component must be capable of performing. *ISO/IEC 24765, Systems and Software Engineering Vocabulary. 3.* a requirement that specifies a function that a system or system component must be able to perform. [ISO/IEC24765]

Task: a requirement, recommendation, or permissible action, intended to contribute to the achievement of one or more outcomes of a process. [ISO/IEC 12207]

Non Functional Requirement: a software requirement that describes not what the software will do, but how the software will do it. ISO/IEC 24765, Systems and Software Engineering Vocabulary. Syn. design constraints, non-functional requirement. See also: functional requirement. For example, software performance requirements, software external interface requirements, software design constraints, and software quality attributes. Non functional requirements are sometimes difficult to test, so they are usually evaluated subjectively. [ISO/IEC24765]

Prototype: *1.* an experimental model, either functional or non functional, of the system or part of the system. *IEEE 1233, 1998 Edition (R2002) IEEE Guide for Developing System Requirements Specifications.* 3.12. *2.* a preliminary type, form, or instance of a system that serves as a model for later stages or for the final, complete version of the system. *ISO/IEC 24765, Systems and Software Engineering Vocabulary. 3.* model or preliminary implementation of a piece of software suitable for the evaluation of system design, performance or production potential, or for the better understanding of the software requirements. *ISO/IEC 15910:1999 Information Technology -- Software user documentation process.* [ISO/IEC24765]

Fifth component: Why is this topic important?

This section explains why the reader should implement the practices presented in the next section. For this package, information is provided to the reader about the reasons for project failure.

Why this topic is important

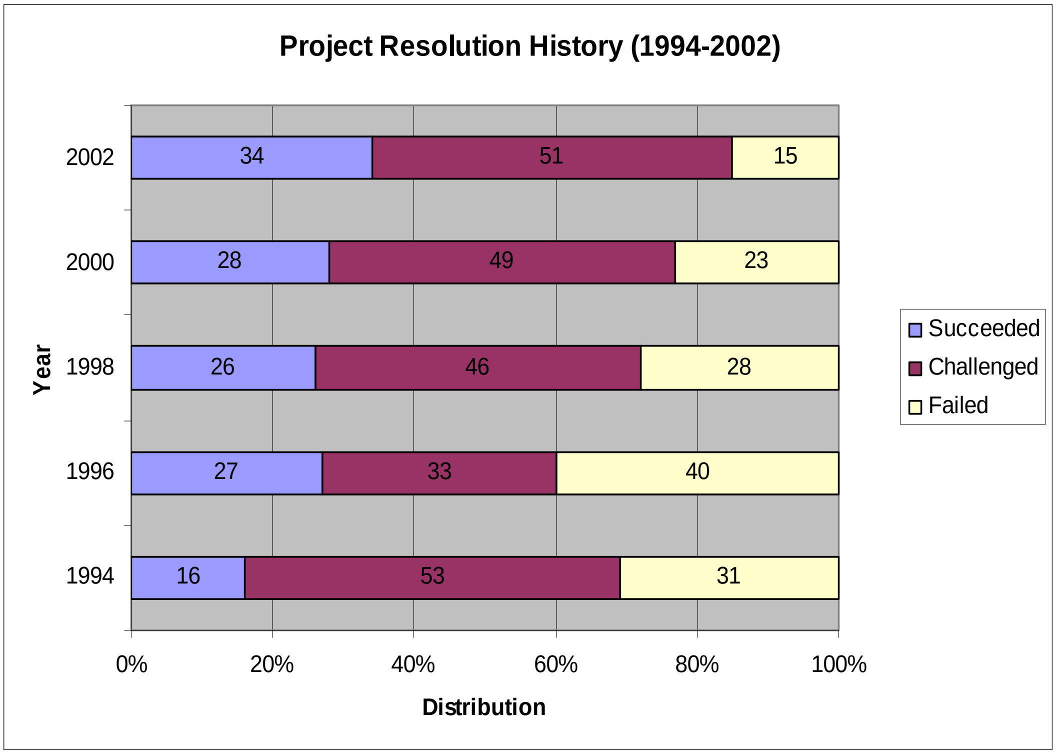
The Chaos Report

Several studies clearly underline the importance of requirement management in software engineering. Among these studies, the Chaos Report published by the Standish Group (www.standishgroup.com/) from 1994 the Standish Group analyzed thousands of IT projects all over the world, categorizing them into three resolution categories:

– *Successful*: The project is completed on time and on budget, with all features and functions as originally specified.

– *Challenged*: The project is completed and operational, but over-budget, over the time estimate, and with fewer features and functions than initially specified.

– *Failed*: The project is cancelled before completion or is never implemented.



Project resolution history

Main Causes of Success

According to the Standish Group, the main causes of *SUCCESS* are:

– User Involvement

– Executive Support

– Clear Business Objectives

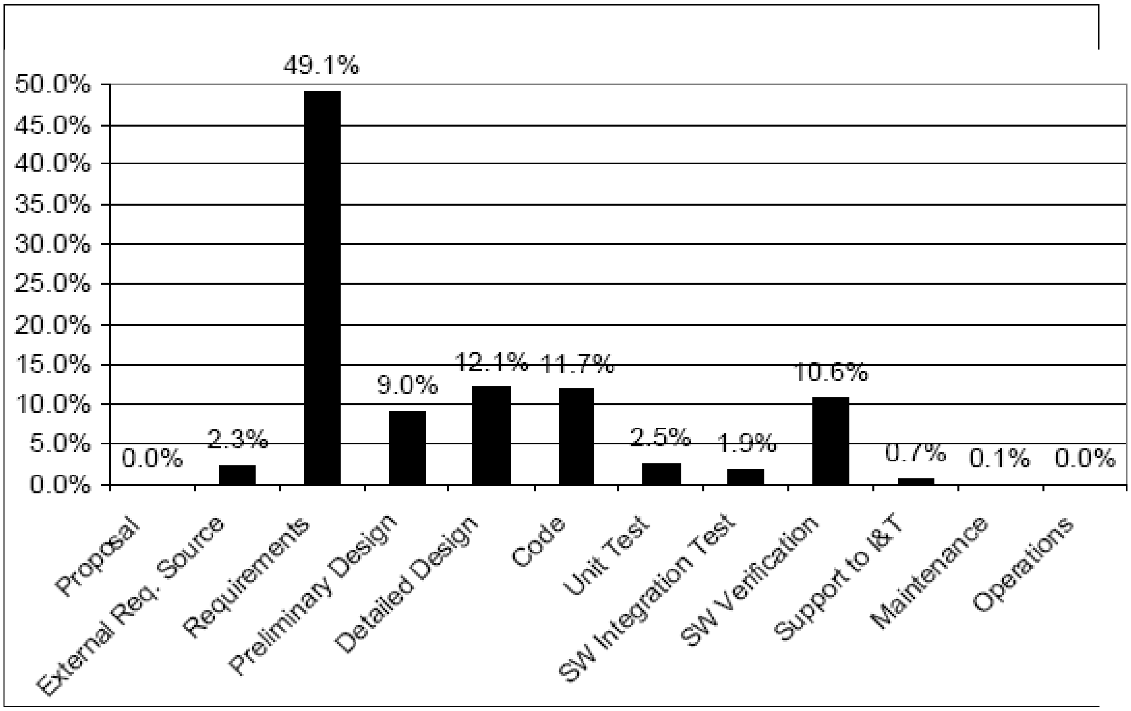
– Experienced Project Manager

– Small Milestones

– Firm's Basic Requirements

Standish Group experts stress the importance of *user involvement* and the *good management and analysis of their requirements*.

Close to 50% of the defects are produced during the requirements phase.



Finally, the reader is presented with data from a real company showing that close to 50% of the defects are produced during the requirements phase.

Sixth component: Overview of tasks

In this section, an overview of the tasks is presented, as well as the component describing each one, i.e. Objectives, Rationale, Roles, Steps, and Artifacts. A list of tasks is also presented.

Overview of tasks
The set of tasks set out in this section is directly related to the Requirements Analysis process. The tasks can be organized according to the chosen life cycle (e.g. sequential or iterative).

Note: The tasks are listed sequentially in this section, but that does not mean to imply their order within any life cycle (i.e. detailed tasks can be organized either sequentially or iteratively). Also in this section are some examples of activity ordering within some life cycles.

For each of the engineering tasks listed, the following elements are briefly described:

- ***Objectives***: key goals of the task *(e.g. understand customer business processes)*
- ***Rationale:*** explanation of why this task is important
- ***Roles***: key roles[65] involved in the task *(e.g. analyst, developer, tester)*
- ***Steps:*** Steps in the task *(e.g. use case elicitation, coding)*
- ***Artefact:*** piece of information or deliverable that can be produced (not mandatory) by one or more tasks *(e.g. design document, source code)*

Note: There are no rules regarding the precise format of an artefact (e.g. requirement specification can be managed in a Word document or on an Excel spreadsheet, or in another (Web-based) tool).

Note: Each of the steps described below must be adapted to the organizational and project

[65] Note that several roles can be played by a single person, especially in VSEs.

context. The rationale behind them is to reduce the risks related to a lack of requirement management and analysis for the VSE.

The effort required for each step will vary according to the size of the project (small or large system) from a few person/hours to several person/days or person/weeks.

The tasks are:

- Requirement identification
- Requirement refinement and analysis
- Requirement verification and validation
- Requirement change management

Seventh component: Detailed description of tasks

This section describes in detail the elements of each task and the steps of each task. Since a task, such as requirement identification, may not be well understood by a VSE, each task has been broken down into a certain number of steps. As an example, the requirement identification task, illustrated below, is decomposed of 4 steps, as follows:

- Collect information about the application domain (e.g. finance, medical),
- Identify project's scope,
- Identify and capture requirements,
- Structure and prioritize requirements.

Requirement identification

Objectives:	The goal of this activity is to clearly define the scope of the project and identify the key requirements of the system.
Rationale:	It is important to clearly define the project scope (boundaries) and to identify key functionalities of the future system with the customer to avoid problems like forgotten key functionalities or requirement "creep".
Roles:	Project Manager
	Analyst
Artifacts:	Use Cases – scenarios
	Requirements Document
Steps:	1. Collect information about the application domain (e.g. finance, medical)
	2. Identify the project's scope
	3. Identify and capture requirements
	4. Organize and prioritize requirements:
Step Description:	***Step 1. Collect information about the domain***: During this step, the analyst captures the key concepts of the customer's business domain. The customer assists the analyst by giving him all the information (existing documentation or explanation) needed to facilitate this understanding. Key concepts are listed in a glossary in the *Software Requirements Specification*

	Document Outline. ***Step 2. Identify the project's scope*** The software analyst, helped by the person in charge of the contractual aspects of the project (sales manager), clearly identifies the main functionalities that are included in the project's scope. *Tips*: Identifying functionalities that are *outside* the project's scope is also very valuable, to clarify any differences of understanding with your customers. ***Step 3. Identify and capture requirements***: Having in mind key concepts related to the customer's business domain, the analyst can start the requirement identification process. No two situations in IT projects are identical. In some cases, most of the requirements have already been identified in a document (a call for tender, in the case of fixed-priced projects). However, in most cases, key requirements are only (orally) mentioned by the customer. The analyst must identify and list the key requirements of the system to be built. During this step, the analyst should not be detailing those requirements. The main goal at this time is to gain a comprehensive view of the system requirements. ***Step 4. Organize and prioritize requirements***: Using the requirements identified in the previous step, the analyst has to organize and structure the identified requirements accordingly (e.g. by business process or by system function). A priority must be identified by the customer for key system functionalities. Priorities can be labelled: • *High* – a functionality that *shall* be implemented • *Medium* – a functionality that *should* be implemented • *Low* – a functionality that *could* be implemented The output of this step is a list of requirements that are organized in the *Requirements Document.*

Eighth component: Example of a life cycle

In order to illustrate the implementation of a set of tasks, the decision was made to add one or two examples of a life cycle. Figure 7-6 illustrates an example of the requirement practice life cycle.

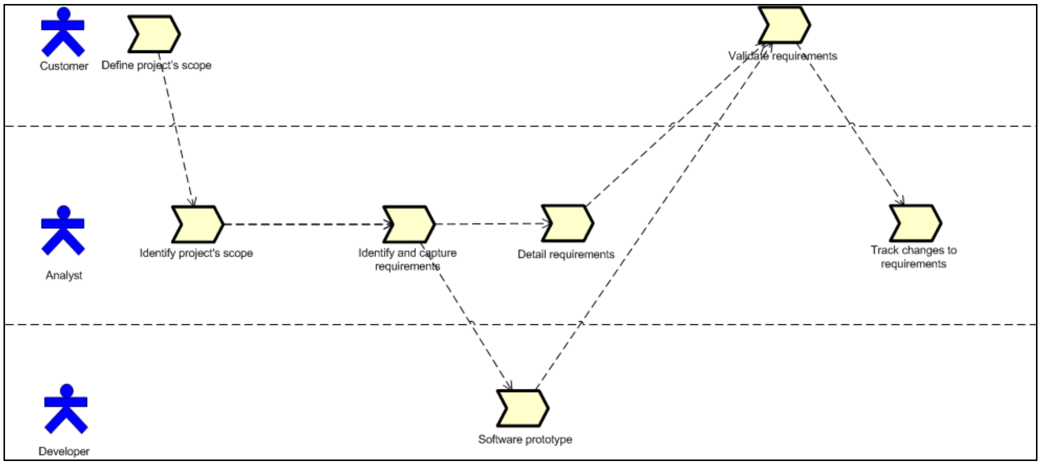


Figure 7-6 Example of a requirement practice life cycle

Ninth component: Appendix A – Templates

In order to facilitate implementation of the tasks, the decision was made to propose examples of a table of contents of a requirements document, as illustrated below. A VSE can then easily adapt the template to suit its needs.

Appendix A – Templates
The templates provided with this deployment package should be customized for the project.

SRS Template Table of Contents –Basic List of Requirements

To be used in an Excel spreadsheet structured as, for example:

ID	*Requirement*	*Description*	*Priority*

SRS Template Table of Contents –Adapted from IEEE 830

1. Introduction
1.1 Purpose
1.2 Document conventions
1.3 Intended audience
1.4 Additional information
1.5 Contact information/SRS team members
1.6 References
2. Overall Description
2.1 Product perspective
2.2 Product functions
2.3 User classes and characteristics
2.4 Operating environment
2.5 User environment
2.6 Design/implementation constraints
2.7 Assumptions and dependencies
3. External Interface Requirements
3.1 User interfaces
3.2 Hardware interfaces
3.3 Software interfaces
3.4 Communication protocols and interfaces
4. System Features

4.1 System feature A
4.1.1 Description and priority
4.1.2 Action/result
4.1.3 Functional requirements
4.2 System feature B
5. Other Non functional Requirements
5.1 Performance requirements
5.2 Safety requirements
5.3 Security requirements
5.4 Software quality attributes
5.5 Project documentation
5.6 User documentation
6. Other Requirements
Appendix A: Terminology/Glossary/Definitions list
Appendix B: To be determined

Tenth component: Appendix B – Checklist

It is well known that a checklist can help in the development of a good work product. In this case, the checklist should help in reviewing the requirements document.

Appendix B – Checklist
This requirements checklist is adapted from [Constr07]

RS 1 Testable	Are all the requirements verifiable (objectively)?
RS 2 Complete	Are the requirements complete?
RS 3 Clear	Are all the requirements clear to all reviewers (e.g. testers)?
RS 4 Correct	Are all the requirements correct (i.e. reflect the customer's requirements exactly)?
RS 5 Unique	Are all the requirements stated only once?
RS 6 Elementary	Are all the requirements broken down into their most elementary form?
RS 7 Scope	Are all the requirements within the scope of the project?
RS 8 High Level	Are all the requirements stated in terms of final need, not perceived means (solutions)?
RS 9 Quality	Are all the quality attributes defined?
RS 10 Traceable	Are all the requirements traceable to a customer's need, systems specification, contractual/proposal clause?
RS 11 Hardware	Is the hardware environment completely defined?
RS 12 Solid	Are the requirements a solid base for design?

Eleventh component: Appendix C – Coverage Matrix for Standards and Models

One of the needs of VSEs is to be able to show to a potential customer or an evaluator that its process complies with a standard or a model. Most VSEs do not have the expertise to develop such a compliance matrix. Also, the authors of the deployment package want to know which component of the ISO standard has been covered in each deployment package.

Although our mandate was to develop a standard aligned with ISO standards, we knew that many VSEs have customers who are not interested in ISO standards, but in the CMMI.

The decision was therefore made to provide a coverage matrix to illustrate which practices or subpractices of a process area were covered by a particular deployment package.

Appendix C – Coverage Matrix for Standards and Models

This appendix shows the traceability of this deployment package to ISO/IEC standards and to the Capability Maturity Model Integration[SM] version 1.2 (CMMI®[66]). For each element of the table, coverage is indicated using the following convention:

- Full Coverage = F
- Partial Coverage = P
- No Coverage = N

Only tasks covered by this package are listed in the compliance matrices.
Comments may be added if necessary.

***Note:* Coverage matrices are not complete, but are provided as an example to illustrate how to fill them in.**

ISO 9001 Coverage Matrix

Title of the Task and Step	**Coverage** F/P/N	**Clause of ISO 9001**	**Comments**

ISO/IEC 12207 Coverage Matrix

Title of the Task and Step	**Coverage** F/P/N	**Clause of ISO/IEC 12207**	**Comments**

CMMI Coverage Matrix

Title of the Task and Step	**Coverage** F/P/N	**Objective/Practice of CMMI**	**Comments**

Twelfth component: Appendix D – Tool

To facilitate the implementation of a deployment package, the decision was made to identify free tools or low-cost tools. For the Requirement Analysis deployment package, we provided a simple and cheap tool: a traceability matrix that could be implemented in Word or Excel. In this case, the Excel spreadsheet tool is available on the author's Web site (details below).

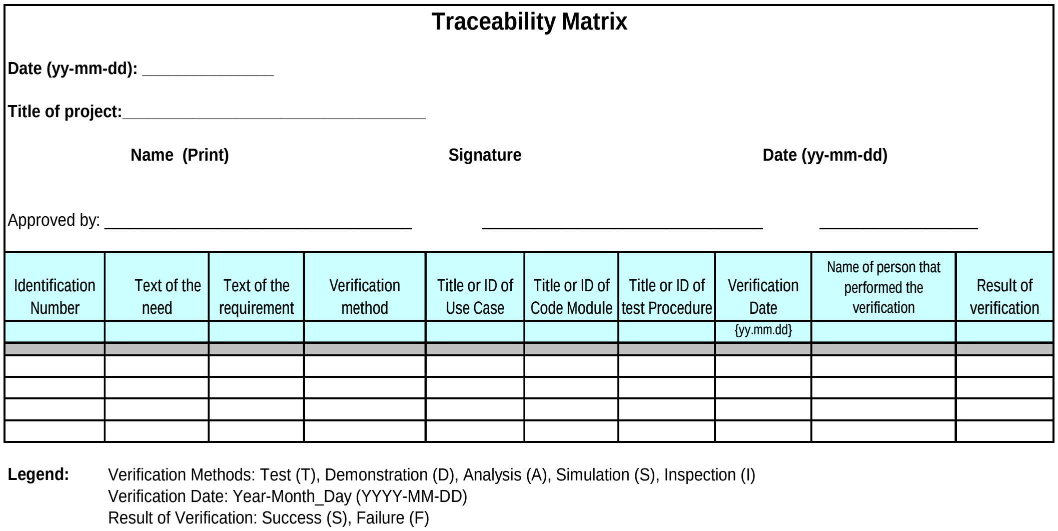
[SM] CMM Integration is a service mark of Carnegie Mellon University.

[®] Capability Maturity Model, CMMI have been registered with the U.S. Patent and Trademark Office by Carnegie Mellon University.

Traceability Tool

Requirement traceability should:

- Ensure traceability for each level of decomposition performed on the project. In particular:
  - Ensure that every lower level requirement can be traced to a higher-level requirement or to the original source
  - Ensure that every design, implementation, and test element can be traced to a requirement
  - Ensure that every requirement is represented in the design and implementation
  - Ensure that every requirement is represented in testing/verification
- Ensure that there is traceability in conducting the impact analysis of requirement changes on project plans, and in activities and work products
- Maintain and update as changes occur
- Be consulted during the preparation of Impact Analysis for every proposed change to the project
- Be planned for, since maintaining the links/references is a labour intensive process that should be tracked/monitored and should be assigned to a project team member
- Be maintained as an electronic document.



Traceability Matrix

Date (yy-mm-dd): ______________

Title of project:________________________________

Name (Print) **Signature** **Date (yy-mm-dd)**

Approved by: ________________________________ _____________________________ ________________

Identification Number	Text of the need	Text of the requirement	Verification method	Title or ID of Use Case	Title or ID of Code Module	Title or ID of test Procedure	Verification Date	Name of person that performed the verification	Result of verification
							{yy.mm.dd}		

Legend: Verification Methods: Test (T), Demonstration (D), Analysis (A), Simulation (S), Inspection (I)
Verification Date: Year-Month_Day (YYYY-MM-DD)
Result of Verification: Success (S), Failure (F)

<table>
<tr><th colspan="2">Instructions</th></tr>
<tr><td colspan="2">The above table should be created in a spreadsheet or database, so that it can be easily sorted by column to achieve bi-directional traceability between columns. The unique identifiers (IDs) should be assigned in a hierarchical outline form, so that the lower level (i.e. more detailed) items can be traced to higher items.</td></tr>
<tr><td>Identification Number</td><td>The Unique Requirement Identification (ID) where the requirement is referenced, and/or the unique identification for decomposed requirements.</td></tr>
<tr><td>Text of the need</td><td>The original text of the need from the customer</td></tr>
<tr><td>Text of the requirement</td><td>The text of the requirement</td></tr>
<tr><td>Verification Method</td><td>The verification method is identified (e.g. Test (T), Demonstration (D), Analysis (A), Simulation (S), Inspection (I)).</td></tr>
<tr><td>Title or ID of Use Case</td><td>The unique identifier of the Use Case or design component where a requirement is designed.</td></tr>
</table>

Title or ID of Code Module	The unique identifier of the software module where the design is realized or coded.
Verification Date	The date the requirement is verified (e.g. tested)
Name of person who performed the verification	The name of the person who performed the verification
Result of the verification	Result of verification (i.e. Success (S) or Failure (F))

Thirteenth component: Evaluation form

Finally, the deployment package includes an evaluation form in order to improve a future version of the package. This form is completed by a VSE and sent to the editor of the package.

Deployment Package – Software Requirements Analysis – Version 1.1

Your feedback will allow us to improve this package, and we welcome your comments and suggestions.

1. How satisfied are you with the CONTENT of this deployment package?

□ *Very satisfied* □ *Satisfied* □ *Neither satisfied nor dissatisfied* □ *Dissatisfied* □ *Very dissatisfied*

2. Are you satisfied that the sequence in which the topics are discussed is logical and easy to follow?

□ *Very satisfied* □ *Satisfied* □ *Neither satisfied nor dissatisfied* □ *Dissatisfied* □ *Very dissatisfied*

3. How satisfied are you with the APPEARANCE/FORMAT of this deployment package?

□ *Very satisfied* □ *Satisfied* □ *Neither satisfied nor dissatisfied* □ *Dissatisfied* □ *Very dissatisfied*

4. Have any unnecessary topics been included? (please describe)

5. What missing topic would you like to see in this package? (please describe)

- Proposed topic:
- Rationale for new topic:

6. **Are there any errors in this deployment package? (please indicate)**

- Description of error:
- Location of error (section #, figure #, table #):

7. Other feedback or comments:

8. Would you recommend this deployment package to a colleague in another VSE?

□ *Definitely* □ *Probably* □ *Not sure* □ *Probably not* □ *Definitely not*

Optional

- Name: ____
- e-mail address : ______________________________

Email this form to : simon.alexandre@cetic.be or claude.y.laporte@etsmtl.ca

7.5 Training material

Once a deployment package has been finalized, the author is asked to prepare some training material. We have proposed to the author of a deployment package that they prepare presentation material (i.e. Power Point slides) to facilitate the training of practitioners in a VSE. A template for the preparation of this material is available on this author's Web site. Once completed, the training material will be available from the same Web site.

7.6 Updating the list of deployment packages

At the Mexico meeting in 2008, the group proposed modifications to the lists of deployment packages for the Generic Profile/Basic Sub-profile. The decision was made to modify the list of deployment packages, as follows:

- remove the Project Retrospective (i.e. lessons learned) deployment package and the Issue Tracking deployment package,
- reduce the scope of the Change Management deployment package to Version Control,
- rename the Coding deployment package Construction and Unit Testing,
- modify the scope of the Unit Testing deployment package to Integration and Tests,
- add a Product Delivery deployment package and a Self-Assessment deployment package.

Figure 7-7 illustrates the list of deployment packages that baselined at the Mexico meeting. The full set of deployment packages can be found in Volume II.

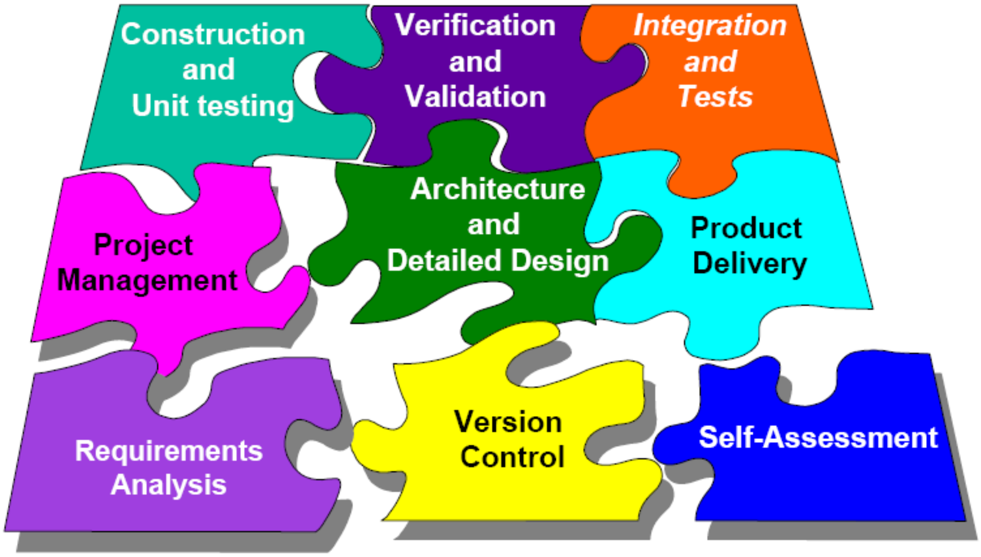


Figure 7-7 List of deployment packages agreed to at the Mexico meeting

Once all the deployment packages have been completed, the coverage matrices from each of them will be integrated into one coverage matrix for each standard or model. This coverage matrix, illustrated in Table 7-6, will be available on the author's Web site. This matrix can be used by a VSE to demonstrate to a customer the coverage of the practices implemented to ISO 9000 or CMMI standard.

Task	Requirement Analysis	Architecture and Detailed Design	Construction and Unit Testing	Software Testing	Verification and Validation	Version Control	Project Management	Product Delivery	Self Assessment
PM.1.1 Review the Statement of Work									

Table 7-7-5 Example of a Coverage Matrix for an Individual Standard or Model

7.7 Quality assurance of deployment packages

At the Mexico meeting, one delegate from Mexico, who happened to be a former graduate student of the author, volunteered to perform quality assurance testing of the deployment packages before they are used in pilot projects. To facilitate this task, the author modified a checklist used to perform formal inspection (see Figure 7-8). This checklist is sent to the authors of deployment packages to allow them to perform a verification of their work before sending their deployment package for quality assurance testing in Mexico.

Checklist – Deployment Package (DP)

Version 0.1, Dated February, 11th 2009

Adapted from: Gilb, Tom, Graham, Dorothy, Software Inspection, Addison-Wesley, 1993.
This checklist specifies the requirements that must be satisfied by the deployment packages (DP).

DP 1 (COMPLETE) – All processes, activities/tasks of Part 5 must be listed in at least one DP for a specific sub-profile.
DP 2 (RELEVANT) – All information must be relevant to the purpose of the document and the section in which it resides.
DP 3 (BRIEF) – Information must be stated briefly.
DP 4 (CLEAR) – Information must be clear to all reviewers.
DP 5 (CORRECT) – Information must be free of technical errors.
DP 6 (CONSISTENT) – Information must be consistent with all other information in the document and its source documents.
DP 7 (UNIQUE) – Ideas should be stated once only and thereafter be referenced.
DP 8 (REFERENCE) – References must be cited for derived process, activity/task
DP 9 (COMMENT) – All comments, notes, suggestions, or ideas that do not form part of the document must be clearly distinguished as such.
DP 10 (RISK) – Any known or suspected adverse consequences of any statement must be clearly indicated.
DP 11 (UNCERTAINTY) – Any known or suspected uncertainty or tolerances associated with quantitative values must be clearly indicated.
GD 12 (PURPOSE) – The DP should have an explicit statement of purpose.
DP 13 (STATUS) – A DP shall show its status clearly: INSPECTED (from the Document Inspection Process) or NOT INSPECTED. “Not inspected” status shall be assumed until the other status is claimed.

Figure 7-8 Checklist used to inspect deployment packages

Once the quality assurance has been completed, the deployment package is sent to the official repository of Working Group 24 at the Standards Council of Canada (SCC). An e-mail is automatically sent to all members notifying them of the availability of a new deployment package. Finally, the members of the VSE support network (explained in the next section) also post the new deployment package on their Web site.

7.8 Establishment of an international network of support centers

In 2006, the author proposed to the CETIC Center in Belgium that a network of international collaborators be established to support VSEs in their own region. It was agreed that the hub of the network would be the ETS. This means that every collaboration agreement is signed between a collaborator and the ETS.

An informal meeting was organized, by Canada, during the November 2008 interim meeting of WG24 in Mexico, to officially launch the Network of Support Centers, as well as to explain the purpose (see below), the objectives, the Collaboration Agreement, and the achievements of the new Network to new participants of WG24. Currently, the members of the Network are:

- Belgium – Centre d'Excellence en Technologies de l'Information et de la Communication (CETIC)
- Canada – École de technologie supérieure (ÉTS)
- Columbia – Parquesoft Foundation
- Finland – Tampere University of Technology, Pori
- France – Université de Bretagne occidentale (UBO)
- Ireland – LERO, The Irish Software Engineering Research Centre
- Luxembourg – Henri Tudor Public Research Centre
- Thailand – Federation of Thai Industries (F.T.I.)

The following organizations have shown an interest in becoming members of the Network:

- Hong Kong – Polytechnic University
- Mexico – Universidad Nacional Autónoma de México (UNAM)

The purpose of the Network is to promote, facilitate, and develop collaborative activities between institutions in the field of software engineering, information technology, and others to improve VSE capabilities, especially in software engineering and information technology. The main objectives of the Network are to accelerate deployment of standards and guides for VSEs and to accelerate the development and application of guides and deployment packages (e.g. through pilot projects).

Description of the contents of the Collaboration Agreement

This section presents the nine articles of the Collaboration Agreement.

Article 1: Purpose

This Agreement aims to promote, facilitate, and develop collaborative activities between the two institutions in the field of software engineering, information technology, and others to improve VSE capabilities, especially in software engineering and information technology.

The ÉTS and 'name of organization' hereby agree to promote all forms of cooperative activities they deem appropriate.

Article 2: Areas of Cooperation

Both parties, in order to achieve the objective set out in Article 1, agree to establish cooperation, especially in the following subjects:

6. Establishment of a network of international expertise in software engineering to help Very Small Entities (enterprises, departments, or projects with fewer than 25 people) involved in software development.

7. Establishment of an integrated catalog of their respective services to promote their regions.

Article 3: Specific Agreement

Some specific agreements in the form of rider(s) to this Agreement detailing cooperation and partnership, as well as financing, will be established as required on a case-by-case basis, subject to approval by the competent authorities of each institution.

Article 4: Consultations

The two partners will consult regularly in order to assess the development of joint activities, review the results and actions under way, and identify new areas of collaboration.

Article 5: Editors

The parties hereby appoint the following individuals to establish and track the initiatives covered by this Agreement:

For the ETS:

Mr. Claude Y. LAPORTE, Professor
Department of Software Engineering and Information Technology
Member of the Software Engineering Research Laboratory (GÉLOG)

For ‘name of organization’:
Name

Article 6: Intellectual Property

Intellectual property resulting from the work carried out within the framework of this Agreement will be subject to applicable statutory requirements and specific procedures undertaken by the parties to this effect. Those who will be involved in carrying out the work will receive due recognition, according to the institutional practices and the intellectual property policies of each party.

Article 7: Financing

The parties agree to do their best to secure the necessary funding for the implementation of this Agreement.

Article 8: Duration

The Agreement shall be valid for an initial period of three (3) years, beginning on the date of signature. It may be renewed through the agreement of both parties.

Article 9: Termination

The Agreement may be terminated by either party without prejudice by giving three (3) months notice. Specific collaboration may survive this Agreement if both parties agree to such collaboration in writing.

7.9 Development of a Process Asset Library and a Web site

Most software practitioners do not like paperwork, and many hate it. In order to attenuate this situation, an organization has to establish a system which allows each practitioner a rapid, user-friendly access to protected and shared documents that can be used or reused in a project to help produce the required documentation on a specific project. In this section, the way the documents are developed by the members of the working group is described.

A Process Asset Library (PAL) is an organized, indexed, searchable, and downloadable repository of process assets that is easily accessible by everyone who needs processes and process-related assets, such as guidance documents (e.g. deployment packages), forms, checklists, templates, examples, lessons learned related Web links, or other process support materials, such as training material. In addition, a PAL ensures that process assets are stored, shared, easily retrieved, and protected. Finally, individuals using the PAL may contribute to the library by making suggestions or providing artifacts.

The information in the PAL is a resource for helping VSEs effect the effort and cost reductions necessary to make a project successful. Assets are also useful to those who are defining, managing, and implementing processes.

Since VSEs cannot afford the overhead to develop and maintain a PAL, the author has developed a public PAL where any VSE can access the assets that are not covered by restrictive copyright clauses, but by clauses typical of Open Source resources. Two versions of the PAL were, in fact, developed, one in French and one in English. The author is the Webmaster responsible for the maintenance of the VSEs' PAL. Finally, this PAL can also be used as a model by a VSE to develop it own PAL, thereby reducing the effort and cost for the VSE.

The CMMI recommends the development of a PAL, as described by the Specific Practice 1.5 Establish the Organization's Process Asset Library (SEI 2006). The author developed the PAL using the suggested subpractices of the CMMI listed in Figure 7-9.

1. Design and implement the organization's Process Asset Library, including its structure and support environment.
2. Specify the criteria for including items in the library.
   The items are selected based primarily on their relationship to the organization's set of standard processes.
3. Specify the procedures for storing and retrieving items.
4. Enter the selected items into the library and catalog them for easy reference and retrieval.
5. Make the items available for use by the projects.
6. Periodically review the use of each item and use the results to maintain the library contents.
7. Revise the organization's Process Asset library as necessary.
   Examples of when the library may need to be revised include the following:
   - New items are added.
   - Items are retired.
   - Current versions of items are changed.

Figure 7-9 Subpractice 1.5 – Establish the organization's Process Asset Library

Eventually, the PAL will be controlled and maintained in accordance with a Maintenance Procedure and a Configuration Control Procedure. The latter will describe the following elements (adapted from Wiegers 2005):

- Identification of the appropriate contents of the collection,
- Development of a scheme for organizing and labelling the many documents we expected to encounter,
- Identification of subject matter experts in various software practice domains to evaluate, revise, and create suitable documents,
- Development of a plan to create and populate the repository, adjusting the plan as needed.

For those candidate documents that we accept, we will track the following milestone dates (adapted from Wiegers 2005):

- Accepted for inclusion in the PAL,
- Permission to include it granted by the owner,
- Converted to a Word document,
- All edits and quality assurance completed,
- Converted to its ultimate delivery format,
- Delivered to the Webmaster,
- Installed,
- Announcement (e.g. e-mail) sent to registered users.

The structure of the PAL is illustrated in Table 7-6.

Name of the Profile	Name of Sub-profile	Title of Deployment Package	Software Tools	Other Tools
Profile Name				
	Sub-profile Name			
	Sub-profile Name			

Table 7-6 Structure of the Process Asset Library

Table 7-7 lists the documents available on the Web for the Basic Sub-profile.

Name of the Profile	Name of Sub-profile	Title of Deployment Package	Software Tools	Other Tools
Generic Profile				
	Entry			
	Basic	Requirements Analysis Architecture and Detailed Design Construction and Unit Testing Integration and Tests Verification and Validation Version Control Project Management Product Delivery Self-Assessment	Traceability Matrix Project Status Report Card	Version Control Guide with SVN Configuration Management with SVN Version Control Guide with CVS Configuration Management with CVS Requirements Management with Xml-Based SRS Issue tracking with GForge Project Management with GForge Process Improvement with OpenOffice Calc
	Intermediate			
	Advanced			

Table 7-7 List of Artifacts of the Generic Profile Available on the Web

Currently, the PAL is located on the ETS infrastructure at the following addresses:

- French PAL: http://profs.logti.etsmtl.ca/claporte/VSE/index.html
- English PAL: http://profs.logti.etsmtl.ca/claporte/English/VSE/index.html

The participants in the Network are also gradually developing a PAL for VSEs, and are providing links to other sites. To date, the other sites are:

- CETIC: http://www.cetic.be/indexEN.php3
- LERO:http://www.lero.ie/research/internationalprojects/softwareprocessesforsmallenterprises/

- SIPA: http://www.sipa.or.th/en/home/index.php

Architecture of the Process Asset Library

The PAL is structured as follows:

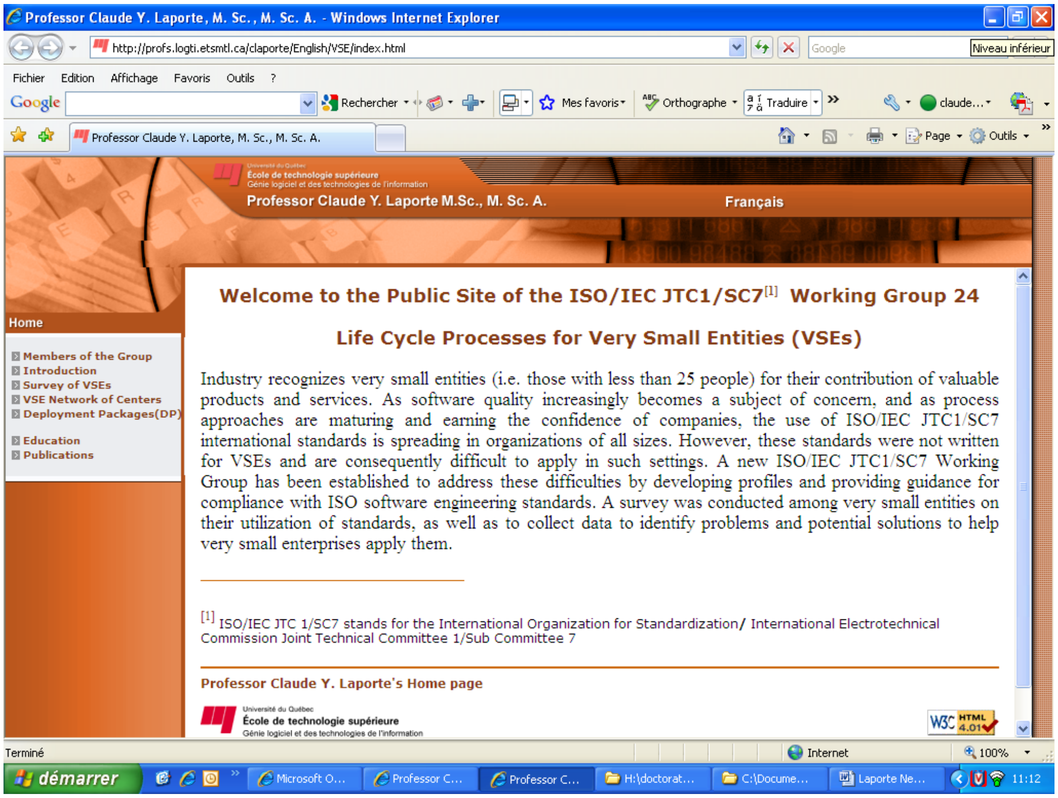
- A home page with a table of contents and the following introduction: "Industry recognizes very small entities (i.e. those with fewer than 25 people) for their contribution of valuable products and services. As software quality increasingly becomes a subject of concern, and as process approaches are maturing and earning the confidence of companies, the use of ISO/IEC JTC1/SC7 international standards is spreading in organizations of all sizes. However, these standards were not written for VSEs and are consequently difficult to apply in such settings. A new ISO/IEC JTC1/SC7 Working Group has been established to address these difficulties by developing profiles and providing guidance for compliance with ISO software engineering standards. A survey was conducted among very small entities on their utilization of standards, as well as to collect data to identify problems and potential solutions to help very small enterprises apply them."
- A page listing the members of ISO working group 24. For each member, the following information is provided:
  - Name of the country represented by the delegate
  - Name and e-mail address
  - Picture of the member

As an example, the following information is provided for the convenor of the working group:

- Tanin Uthayanaka (taninu@mozart.inet.co.th) is the Chief Operating Officer of Siamguru Co. Ltd. (a VSE). He is SW-CMM (Software Capability Maturity® Model) lead assessor, candidate CMMI® (Capability Model Integration) lead appraiser, and a member of the Thai Industrial Standards Institute – Software and Systems Engineering Standard Group.
- A page presenting the results of the survey conducted by the working group
- A page presenting the Network of Support Centers to support VSEs
- A page describing and listing the deployment packages for each profile

- A page listing the publications and communications made by the members of the working groups
- A page listing events for VSEs, such as training or briefing sessions, to be added in the future

Figure 7-10 shows the home page of the English PAL located on the author's Web site at the École de technologie supérieure.



Welcome to the Public Site of the ISO/IEC JTC1/SC7[1] Working Group 24

Life Cycle Processes for Very Small Entities (VSEs)

Industry recognizes very small entities (i.e. those with less than 25 people) for their contribution of valuable products and services. As software quality increasingly becomes a subject of concern, and as process approaches are maturing and earning the confidence of companies, the use of ISO/IEC JTC1/SC7 international standards is spreading in organizations of all sizes. However, these standards were not written for VSEs and are consequently difficult to apply in such settings. A new ISO/IEC JTC1/SC7 Working Group has been established to address these difficulties by developing profiles and providing guidance for compliance with ISO software engineering standards. A survey was conducted among very small entities on their utilization of standards, as well as to collect data to identify problems and potential solutions to help very small enterprises apply them.

[1] ISO/IEC JTC 1/SC7 stands for the International Organization for Standardization/ International Electrotechnical Commission Joint Technical Committee 1/Sub Committee 7

Professor Claude Y. Laporte's Home page

Université du Québec
École de technologie supérieure
Génie logiciel et des technologies de l'information

Figure 7-10 Home page of the Process Asset Library

Monitoring access to the Process Asset Library

The author has instrumented the French and English VSE Web sites with code provided by Google[67], called Google Analytics, to capture information about access to the sites. The services provided by Google Analytics are free, easy to install, and easy to use.

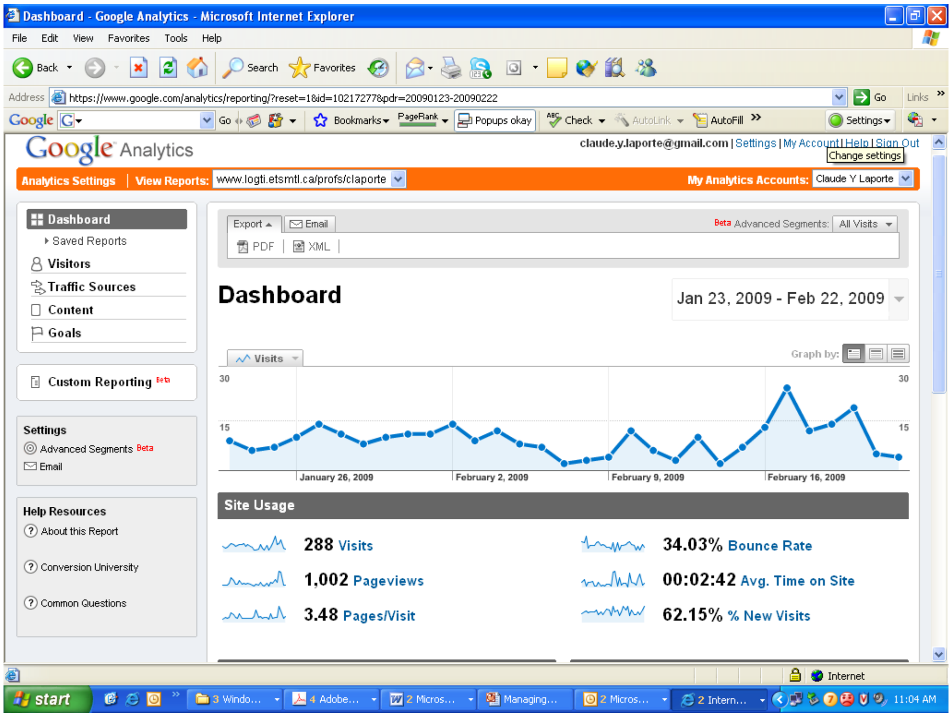
The following information is monitored and captured on a monthly basis by the Webmaster:

- Number of visits
- Percentage of new visits
- Number of countries

[67] http://www.google.com/analytics/

- Number of pages visited
- Number of accesses to VSE pages

Figure 7-11 illustrates the Google Analytics home page for the VSE Web site.


Dashboard - Google Analytics - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address https://www.google.com/analytics/reporting/?reset=1&id=10217277&pdr=20090123-20090222
Go Links
Google Analytics
claude.y.laporte@gmail.com | Settings | My Account | Help | Sign Out
Change settings
Analytics Settings | View Reports: www.logti.etsmtl.ca/profs/claporte
My Analytics Accounts: Claude Y Laporte
Dashboard
Saved Reports
Visitors
Traffic Sources
Content
Goals
Custom Reporting Beta
Settings
Advanced Segments Beta
Email
Help Resources
About this Report
Conversion University
Common Questions
Export
Email
PDF
XML
Beta Advanced Segments: All Visits
Dashboard
Jan 23, 2009 - Feb 22, 2009
Visits
Graph by:
30
15
January 26, 2009
February 2, 2009
February 9, 2009
February 16, 2009
Site Usage
Internet
start
11:04 AM

288 Visits

1,002 Pageviews

3.48 Pages/Visit

34.03% Bounce Rate

00:02:42 Avg. Time on Site

62.15% % New Visits

Figure 7-11 Web site access data

Table 7-8 lists the data on accesses to the PAL Web site between December 2008 and May 2009. The data on the right-hand side of the table show a large number of accesses to VSE pages. As the site will be populated with more documents, such as deployment packages and eventually case studies, the number of accesses should increase substantially.

Month	Number of visits	Percentage of new visits	Number of countries	Number of pages visited	Number of accesses to VSE pages
December 2008	180	61%	38	1140	575
January 2009	269	63%	33	876	521
February 2009	252	58%	35	1173	482
March 2009	197	63%	36	789	489
April 2009	171	65%	30	674	433
May 2009	203	66%	28	945	452

Table 7-8 Monthly Access to the VSE Web Site

7.10 Conducting pilot projects

Fenton states: 'Unfortunately software methods and techniques often find their way into standards even when there is no reported empirical, quantitative evidence of their benefit' (Fenton et al. 1994). In order to address this comment, the author proposed to the working group that pilot projects be conducted using draft versions of the standards. A pilot project is a method for exploring the value of a new technological concept via an objective study conducted in a somewhat realistic setting (adapted from Glass 1997).

Ed Yourdon wrote this a few years ago: 'For the past ten years, the data processing profession has been slowly learning that technology transfer is one of its biggest problems, if not the biggest problem.'[68] Pilot projects are an important means of reducing risks and learning more about the organizational and technical issues associated with the deployment of software engineering practices (e.g. deployment packages), ISO standards, and guides. A successful pilot project is also an effective means of building the adoption of new practices by members of a VSE.

Pilot projects should provide the working group with feedback that is useful for the modification of the draft standards before they are officially published by the ISO. Pilot projects should also provide qualitative and quantitative data to managers, systems integrators, and government organizations, to substantiate claims made by the new standards.

To be credible, the pilot projects should satisfy the following requirements (Fenton et al. 1994):

- The pilot project experiment has to be designed correctly,
- The pilot project has to be performed in a real situation. It is not a toy project, i.e. an artificial problem in an artificial situation,

[68] Forward, Agents of Change, by Barbara Bouldin, Englewood Cliffs, New Jersey, Yourdon Press, 1989.

- The measurements have to be appropriate to the goals of the experiment,
- The experiment has to be run for long enough.

Pilot projects are an important means of reducing risks and learning more about the organizational and technical issues associated with the deployment of new software engineering practices. A successful pilot project is also an effective means of building the adoption of new practices by an organization such as a VSE.

Canada proposed that pilot projects of the deployment packages be conducted before the documents are published by the ISO. It was agreed that, at a minimum, the following information would be collected:

- The effort (staff-hours) and time (days) to deploy by the VSE,
- The usefulness for the VSE,
- A verification of the understanding of the VSE,
- Self-assessment data – self-assessment at the beginning and end of the pilot.

Sometimes a pilot project is an initial implementation of an improvement, usually on a small and controlled scale, before management approves its installation in an organization (adapted from McFeeley et al. 1996). In our case, it will be the experimentation with the Deployment Packages before the publication of the standards by the ISO, that will put us, the members of the working group, in a better position to 'sell' the adoption of the standard to VSEs, to government agencies, and to systems integrators, such as Boeing or Sony, worldwide.

The following countries volunteered to conduct pilot projects in 2009: Belgium, Canada, Columbia, Ireland, and Mexico. The volunteers requested a set of guidelines so that pilot projects would be conducted similarly around the world. Canada agreed to provide a process for conducting pilot projects by the spring of 2009.

7.11 Description of the Pilot Project deployment package

The purpose of this document is to provide tailorable guidelines and materials in order to select and conduct a pilot project.

To develop the deployment package, the author referred to the CMMI (SEI 2006) and to a paper by Glass (Glass 1997). Figure 7-12 lists the typical work products and subpractices of Specific Practice 1.3 of the Organizational Innovation and Deployment Process Area of the CMMI.

Specific Practice 1.3-1 Pilot Improvements
The pilot process is deployed and technology improvements are evaluated to select which ones to implement. Pilots are performed to assess new and unproven major changes before they are broadly deployed, as appropriate.

The implementation of this specific practice may overlap with the implementation of the Implement the Action Proposals specific practice in the Causal Analysis and Resolution process area (e.g. when causal analysis and resolution are implemented organizationally or across multiple projects).

Typical Work Products
1. Pilot evaluation reports
2. Documented lessons learned from pilots

Subpractices
1. Plan the pilots.
   When planning pilots, it is critical to define quantitative criteria to be used for evaluating pilot results.
2. Review and obtain relevant stakeholder agreement on the plans for the pilots.
3. Consult with, and assist, the people performing the pilots.
4. Perform each pilot in an environment that is characteristic of the environment of broadscale deployment.
5. Track the pilots against their plans.
6. Review and document the results of pilots.

Pilot results are evaluated using the quantitative criteria defined during pilot planning. Reviewing and documenting the results of pilots usually involves the following:

- Deciding whether to terminate the pilot, replan and continue the pilot, or proceed with deploying the process and technology improvement
- Updating the disposition of process- and technology-improvement proposals associated with the pilot
- Identifying and documenting new process- and technology-improvement proposals as appropriate
- Identifying and documenting lessons learned and problems encountered during the pilot

Figure 7-12 Subpractices of the Organizational Innovation and Deployment Process Area (SEI 2006)

The tasks of the Pilot Project deployment package are:

- plan the pilot project,
- conduct the pilot project,
- evaluate the results of the pilot project.

The following paragraphs describe each task in more detail.

Task 1– Plan the Pilot Project

The goals for the pilot are established, evaluation guidelines are identified, and a pilot project plan is developed. The commitment of VSE management to the plan is obtained. Figure 7-13 lists the 12 steps required to perform this task.

Step 1: Define the characteristics and context of the VSE
Step 2: Define the problem to be addressed
Step 3: Select the technology to pilot and the project piloting the technology
Step 4: Examine the cost and benefit of the pilot project
Step 5: Identify the variables to be measured
Step 6: Define the criteria for success
Step 7: Elaborate the pilot evaluation criteria
Step 8: Define the mechanism for evaluating the pilot
Step 9: Define the stakeholder inputs needed and the means of obtaining them
Step 10: Define ways of gathering data during the pilot.
Step 11: Develop the pilot project plan
Step 12: Obtain commitment to the plan from the VSE management

Figure 7-13 Steps required to plan the pilot project

In step 3, titled 'Select the technology to pilot and the project piloting the technology', we select, with VSE management, a technology (e.g. deployment package) and the project that will pilot the technology. Some criteria to consider when selecting a pilot include (adapted from[69]) the following:

- The pilot project should be scoped such that it can be executed in a short time frame with the limited VSE resources.

- The pilot project should not on a critical path, such that failure would have a significantly adverse impact on the VSE.

- The pilot projects should have a very good chance to succeed.

- The participants in the pilot project should be advocates of the proposed technology, or at least be open-minded to it. They should be either 'innovators' or 'early adopters' (Rogers 2003):

    - **Innovators** – They are the first people to adopt a technology, and comprise about 2.5% of all likely consumers or users. They are driven by a desire to be rash and to do something daring. They will spend hours trying to get the technology to work and they do not need good quality documentation. They could also provide helpful criticism (Garcia et al. 2007).

    - **Early adopters** – They are more integrated into an organization's development culture. They are respected by their peers, and use new ideas discreetly. By making judicious innovation decisions, they decrease the uncertainty of a new idea by adopting it while personally informing colleagues of its success.

[69] http://www.sei.cmu.edu/productlines/frame_report/launch.inst.PL.htm

- The participants in a pilot project should not be part of either the 'late majority adopters' or 'laggards' categories, as defined by (Rogers 2003):
    - **Late majority adopters** – They are more skeptical, and their adoption may be the result of economic pressure or peer pressure. Most of the uncertainty about a new idea must be resolved before a late adopter will agree to try it. They will not volunteer for an improvement project (Garcia et al. 2007).
    - **Laggards** – They jump on the bandwagon only when they are certain that a new idea will not fail, or when they are forced to change by mandate from managers or customers. They avoid improvements until they have no other choice (Garcia et al. 2007).

In Step 6, titled 'Define the criteria for success', the deployment package provides the following list of criteria to help the coordinator and the owner of the VSE to identify the criteria for success, based on the following:

- A self-assessment performed at the beginning and end of the pilot
- Measurement of the effort required to conduct the pilot (from the pilot coordinator)
- Measurement of the effort required to conduct the pilot (from the VSE)
- Evaluation of the usefulness of the technology; for example, does it:
    - reduce cycle time (actual)
    - reduce redundant work
    - reduce rework
    - improve product stability
    - improve process control (standardize)
    - improve cost/schedule estimation (accuracy)
    - improve visibility/accountability of process
    - improve training and equipment investment effectiveness
    - improve quality/functionality/performance
    - shorten schedule/time to market
    - reduce cost

  - conform with regulatory/audit requirements
  - enhance the organization's reputation for service
  - increase competitive edge

- Verification of understanding on the part of the VSE
  - Was the technology used to its fullest extent?
  - Were any of the key features of the technology not used?
  - Were all the key features of the technology used?
  - Was the expected impact made or not?

- Capture of the lessons learned:
  - To improve the selection and planning of future pilot projects
  - To improve the way future pilot projects are conducted
  - To improve the technology piloted (e.g. deployment package)
  - To improve ISO/IEC 29110 IS and TR (e.g. Part 5)

In Step 11: Develop Pilot Project Plan, the pilot project coordinator is asked to develop a plan. The figure below lists the elements of the table of contents of that plan.

1. Purpose
2. Acronyms
3. Reference documents
4. Overview
5. Roles and responsibilities
6. Description of tasks, effort, and artifacts
7. Resources
8. Interfaces / Dependencies
9. Risks

Table of Contents of the Pilot Project Plan

Task 2 – Conduct the pilot project

The goal of this task is to conduct the pilot according to the plan and, if unexpected events occur, to adjust the plan accordingly.

Step 1: Conduct the pilot according to the plan
Step 2: Save data

Task 3 – Evaluate the results of the pilot project

The goals of this task are to evaluate the results of the pilot project and to improve future pilots.

Step 1: Conduct the evaluation
Step 2: Perform a cost/benefit analysis
Step 3: Produce the pilot project report
Step 4: Store all the information

In Step 3: Produce the Pilot Project Report, the coordinator is asked to produce a report. The figure below lists the elements of its table of contents.

1.Purpose
2.Acronyms
3.Reference documents
4.Overview
5.Description of tasks, effort, and schedule deviations
6. Resources
7. Interfaces/Dependencies
8. Risks
9. Report on lessons learned
    What went well
    What could have gone better
    What surprised us
    Lessons learned
Summary questions

Table of Contents of the Pilot Project Report

The deployment package provides the description of the responsibilities for the 3 main stakeholders of a pilot project: the Pilot Project Coordinator, the Management of the VSE, and the Participant(s) in the Pilot Project, as described in Table 7-9.

Role	**Definition**
Pilot Project Coordinator	The individual who will contact a VSE, explore the possibility of conducting a pilot project with its management, plan and conduct a pilot project, and produce a report on the pilot project.
Management of the VSE	A person, within a VSE, who has the authority to approve and allocate the internal resources to conduct the pilot project, and the authority to deploy the technology if the pilot project is successful.
Participant(s) in the Pilot Project	A person or persons, under the authority of the management of the VSE, who will be trained and who will participate in the project pilot.

Table 7-9 Definition of Key Roles during a Pilot Project

The deployment package provides a description of the definition of the artifacts, as listed in Table 7-10, used or produced during the pilot project.

Artefacts	Definition
Pilot Project Plan	Plan that defines the steps for selecting and conducting a pilot in a VSE
Deployment Package	A deployment package (DP) is a set of artifacts developed to facilitate the implementation of a set of practices, of a framework, in a VSE.
Pilot Project Report	This report collects information about conducting the pilot project. It also document the lessons learned, i.e. what went well during the pilot project and what could be improved in future pilot projects.
Effort Spreadsheet	The spreadsheet captures the effort of the project pilot coordinator and the effort of the participants in the pilot project.

Table 7-10 Definition of Artifacts

Any pilot project may fail. It has been reported that the probability of failure in a software process improvement (SPI) program is significant, approaching 70% (Statz et al. 1997). In order to help identify the potential risks, a list of risk factors is provided. Those risk factors are separated into thirteen categories of risk (see Figure 7-14). Once the risks have been identified, each risk should be mitigated by reducing its likelihood (probability) and/or its impact (consequence).

List of Software Process Improvement Risk Factors

Source: Joyce A. Statz, Ph.D., Don Oxley, Patrick O'Toole Identifying and Managing the Risks for Software Process Improvement, TeraQuest Metrics, Inc. Published in Crosstalk, Volume 10, No. 4, April 1997**.**

Mission and Goals

- Extent to which the process improvement fits in with the organization's mission and goals
- Relation to other process improvement efforts
- Anticipated work flow impact from SPI changes
- Alignment of the organization, shared vision of business, and need for change

Culture

- Attitude toward change, based on prior change efforts
- Experience with quality programs, level of success
- Action orientation for solving problems vs. political struggles
- Use of facts to manage the organization and business
- Patience with change; ability to spend time socializing
- Tool orientation – expectation that tools can solve the problems
- Level of "planfulness" – ability of the organization to plan
- Ability of organization members to interact with various levels of the organization openly at meetings
- Ability of organization members to manage meetings effectively
- Level of experience with defined processes in the organization

Organization Management

- Executive involvement in SPI effort
- Management support for SPI effort
- Management functioning as management teams
- Middle management participation in SPI efforts
- Credibility of management with respect to SPI change
- Ability of the organization to deal with personnel issues and develop staff
- Degree of empowerment in organization

Process Users

- Developer involvement in SPI efforts
- Level of acceptance by individuals of SPI efforts
- Training needs of users with respect to SPI results being met

Budget and Cost

- Manageability of SPI project size
- Sufficiently large dedicated SPI staff
- Sufficient budget for the SPI project
- Established cost controls for the SPI project

Schedule for SPI Development

- Realistic dates in the SPI plan
- Level of aggressiveness of the SPI schedule, as seen by SPI staff

Content of Deliverables

- Requirement stability for items in the SPI plan
- Complete and clear requirements for the SPI project
- Testability/pilot possibilities for each SPI activity
- Dependencies on other efforts for SPI project success

SPI Project Management

- Approach to planning and monitoring the SPI project
- Adequacy of impact assessment for the SPI project
- Supporting policies and standards for SPI efforts

- Type and amount of communication by SPI PM to SPI team and organization
- Commitment process for starting SPI activities and handling changes
- Ability and experience of SPI PM with projects of this type
- Attitude of SPI PM toward SPI efforts of the organization
- Authority of the SPI PM and support of the organization for the PM

SPI Development Process

- Experience of the SPI team using a defined SPI process
- Early defect identification by the SPI team
- Change control for SPI work products
- Defect tracking of SPI defects by the SPI team

SPI Development Environment

- Physical facilities for use by SPI action teams
- Hardware and software support for SPI action teams
- Amount of communication with organization by members of the SPI action teams
- Consultant support (advice, training) to the SPI action teams

SPI Staff

- Staff availability for the SPI action teams
- Mix of staff skills for the SPI action teams
- Respect for permanent SPI staff and for action teams
- Experience with organizational change among permanent SPI staff
- Training of the SPI action teams and permanent SPI staff
- Spirit and attitude of the SPI action teams and permanent SPI staff
- Productivity of permanent SPI staff and SPI action teams
- Personality fit of permanent SPI teams for process improvement work
- Expertise of the SPI action teams in the specific work domain

Maintenance of SPI Deliverables

- Complexity of deliverables from the SPI action teams
- Availability of deliverables from the SPI action teams by the organization
- Availability of process owners for defined processes

Organization Stability

- Organizational stability in structure and management

Figure 7-14 List of risk factors

Finally, like the other deployment packages, this one provides coverage matrices to ISO standards and the CMMI model. Table 7-11 illustrates a partially completed coverage matrix for the CMMI model.

Objective/ Practice of CMMI	**Coverage F/P/N**	**Title of the Task**	**Comments**
1. Plan the pilots.	**F**	Plan the pilot project	
2. Review and obtain relevant stakeholder agreement on the plans for the pilots.	**F**	Plan the pilot project	
3. Consult with and assist the people performing the pilots.	**F**	Conduct the pilot project	
4. Perform each pilot in an environment that is characteristic of the environment of a broad scale deployment.	**F**	Conduct the pilot project	
5. Track the pilots against their plans.	**F**	Conduct the pilot project	

Objective/ Practice of CMMI	Coverage F/P/N	Title of the Task	Comments
6. Review and document the results of pilots.	**F**	Evaluate the results of the pilot project	

Table 7-11 Example of the CMMI Coverage Matrix

The Pilot Project Deployment Package, the Pilot Project Plan template, the Pilot Project Report Template, and the Pilot Project Confidentiality Agreement can be found in Volume II.

7.12 Results of pilot projects

The members of the working group should be starting pilot projects during the summer of 2009. The first reports of the pilot projects will be presented and discussed at upcoming meetings of the working group in Peru and in Japan.

7.13 Case studies

Once a few pilot projects have been completed, the members of the Network will develop short case studies. The content of a case study could be as follows (adapted from Garcia et al. 2006):

- Context: domain, size, country, experience with process improvement (PI), and context factors that would help an organization understand its situation as it begins its improvement effort; description of the context before the improvement is implemented
- Initial reaction of the organization to proposals related to implementing the process improvement (e.g. a deployment package)
- Approach to overcoming an initial negative reaction: assuming that the first reaction to process improvement proposals is negative, an explanation of what was done (pilots, bringing in other executives, etc.) to overcome this reaction
- Actions: what actions were actually taken to implement improvements
- Results: qualitative and/or quantitative from the effort, preferably from the viewpoint of the executive sponsor
- Critical success factors: the factors that the participants believe contributed most concretely to the organization’s adoption success
- Lessons learned
- Future plans

Once developed and reviewed, the case studies will be communicated in conferences and published in the author's Web site PAL, as well as on the Web sites of the members of the Network of Support Centers for VSEs.

7.14 Establishment of an education interest group

So far, the means to accelerate the adoption of standards by VSEs has been described. A target audience, and an often forgotten one, in the area of software engineering standards comprise undergraduate and graduate students. Since many of them will work in environments such as VSEs, very small projects, or very small departments, it would be wise to address this audience in classrooms.

Academic institutions, in collaboration with their local enterprises, are best suited to determining which courses meet the needs and culture of their VSE's constituency. Since a few members of WG24 are from academia, the author has proposed to establish, within working group 24, an education interest group (EIG) having the following objectives:

- To help educators teach the future ISO standards for VSEs by developing educational material,
- To sensitize undergraduate and graduate students to the ISO standard for VSEs.

The EIG should participate in the International Cooperation on Education about Standardization (ICES) network. The ICES network was established in 2006 by people from industry, academia, and standards organizations interested in education about standardization. Their mission is to promote education about standardization and improve its quality and attractiveness to all stakeholders. The objectives of ICES are (ICES 2009):

- To develop and maintain an interdisciplinary network of people interested in education about standardization,
- To facilitate the development of policies and infrastructures to support education about standardization (e.g. in companies, industries, regions, nations, and worldwide),
- To seek cooperative relationships with organizations that provide training and education in this area,
- To professionalize education about standardization by, for example:
    - Tightening the link between up-to-date standardization research and education
    - Facilitating a repository for curricula

- Facilitating a repository for educational materials
- Facilitating peer review of educational materials
- Stimulating the development of guidelines, innovative approaches, and educational materials
- Organizing workshops.

The EIG could provide the following services:

- Collect and disseminate information through a newsletter or other publications,
- Prepare educational material, such as presentation material and case studies,
- Organize sessions at conferences,
- Sponsor conferences, symposia, and workshops,
- Organize working groups for education, research, and development,
- Serve as a source of technical information for other working groups of SC7, who want to develop an approach similar to that of WG24 in developing profiles for VSEs.

The educational material developed by academia could be used for other audiences, such as:

- Governmental organizations, which can promote or mandate the use of ISO standards to VSEs,
- System integrators, who can mandate the use of ISO standards to VSEs,
- Customers, who can mandate the use of ISO standards to VSEs,
- Owners of VSEs,
- Practitioners within VSEs,
- ISO SC7 working groups.

7.15 Conclusion

In the introduction to this chapter are listed the requirements developed by the working group (see Table 7-12) to facilitate the adoption of the new standards and technical reports by VSEs.

- R07 – Implementation of the set of workproducts (i.e. profiles, guides) must be affordable (very low initial cost of entry in terms of training).
- R08 – Use of the set of workproducts must be affordable and should be intuitive (i.e. consultant services should not be necessary).
- R15 – The set of workproducts should provide the whole spectrum of documents (from standards to educational material, i.e. tutorial information).
- R29 – The set of workproducts should propose examples of life cycles to facilitate choice.
- R33 – The set of workproducts should propose a definition of documents (as assemblies of workproducts). For example, templates (e.g. requirements templates – use cases)
- R37 – The set of workproducts should include compliance table checklists (assessment guide).
- R47 – The guide should include support for documentation, particularly templates.
- R52 – The guide should provide examples (e.g. plans, workproducts, and other deliverables).
- R56 – The guide should be compact (about 50 pages), excluding templates.
- R57 – The guide should be available free of charge on the Web

Table 7-12 Requirements to Facilitate the Adoption of ISO Standards for VSEs (MOS-023 2007)

To address these requirements, the following mechanisms have been presented in this chapter: the development of a Web site dedicated to VSEs, the establishment of an education interest group, or EIG, within WG24 (e.g. courses for undergraduate students), the development of a set of deployment packages, the conduct of pilot projects, the development of case studies, and the establishment of a Network of Support Centers for VSEs.

In the next chapter, the last phase of the 6-phase innovation process is presented, along with suggestions for future work and a conclusion.

Chapter 8

Conclusion and Future Work

8 Chapter 8 – Conclusion and Future Work

8.1 Introduction

This chapter presents the conclusion to this work and an outline of the author's future work. With regard to future work, the following topics are discussed: application of the concepts of profiles and deployment packages developed by WG24 to the domain of systems engineering for the development of products by small and very small entities; measurement of the adoption of the ISO standards and deployment packages by VSEs worldwide; establishment of a technology transfer center for VSEs at the ÉTS; development of profiles for critical software developers, e.g. for software for medical device developers and for scientific software developers; development of courses by the education interest group introduced in this work; and development of software plug-ins to facilitate and accelerate utilization of the deployment packages.

In previous chapters, the author presented the first five phases, of Rogers' 6-phase innovation process, as illustrated in Figure 8-1 (Rogers 2003). In the first section of this chapter, the author presents the last phase, titled consequences. This topic is important, since the consequences of the new ISO standard for VSEs have to be forecast and potentially either assessed or measured.

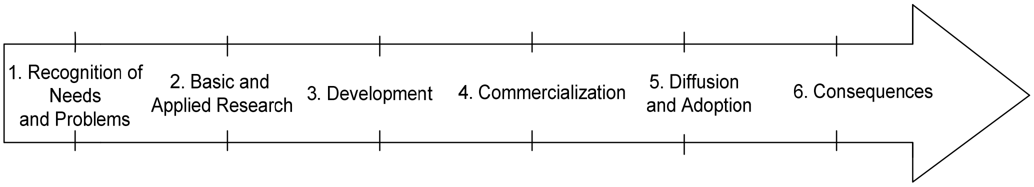

1. Recognition of Needs and Problems
2. Basic and Applied Research
3. Development
4. Commercialization
5. Diffusion and Adoption
6. Consequences

Figure 8-1 Phases of the Innovation Process (adapted from Rogers 2003)

8.2 Consequences of the adoption of innovations

The promoters of an innovation are often optimistic by nature, and tend to market and 'sell' their innovation without looking at possible consequences: e.g. undesirable, direct or indirect, anticipated or unanticipated. Moreover, the issues related to the consequences of an innovation are often forgotten or even deliberately ignored. Rogers offers the following reasons why there are few studies on the consequences of innovation (Rogers 2003):

- Change agents and agencies tacitly assume that the consequences of innovations will be positive.
- The consequences of an innovation usually manifest themselves over extended periods of time, and it may take a few years to observe them.

- Consequences are difficult to measure; for example, it is difficult for a researcher from one cultural environment to assess potential consequences in another cultural environment.

Feng lists a few issues related to the area of standardization (Feng 2003):

- There have not been many empirical studies on how standardization works in practice. Few have tackled implementation issues, and even fewer have tried to compare implementations across diverse cultural contexts, to see whether or not global standards really do work at a global level (and if they do, how).
- Technical experts, like everyone else, cannot see all sides of a problem, but instead adopt certain perspectives that necessarily color their interpretation of what 'the problem' is.
- Standard setting is viewed as intrinsically beneficial because it promotes coordination, but how this actually occurs and how this in turn shapes or constrains future technological development is something not well addressed in the economics literature.
- Standards can be thought of as semi-formal voluntary rules which, when successful, commit a group of developers to a particular technological design or trajectory (technological lock-in). Lock-ins are to be avoided, since the unintended consequences of a particular technology are never known in advance.
- Standard setting is an activity with potentially deep political and social implications. Unfortunately, in many studies the social significance of standardization is never made clear.

The author lists some potential consequences or side-effects that could result from the publication of the ISO 29110 standards in Table 8-1. Consequences of the following 3 strategies are presented:

- Imposing the standards on all the VSEs in a country
- Imposing the standards on all a customer's VSEs70
- Not imposing the standards on VSEs (laissez-faire)

[70] e.g. from a large enterprise or a government agency

Imposing the standards on all the VSEs in a country, or on all a customer's VSEs	• The standards could reduce performance, harming the competitiveness (e.g. cost, development time) of some VSEs by mandating the use of processes; • Some VSEs may decide to shut down, instead of complying with such regulation; • Some VSEs may lose valuable employees who do not want to use formal processes; • Some VSEs may decide to ignore the regulation; • Some VSEs might not be able to afford to implement the standard • because of lack of internal expertise; • Some VSEs might not be able to afford to implement the standard because of lack of available budgetary resources for training employees; • Some VSEs might not be able to afford the cost of buying the set of ISO documents; • Some VSEs might not be able to afford the cost of certification; • Some VSEs may not have the knowledge required to understand the activities and tasks required by the standards; • The standards may impose practices that are in opposition to the culture of a VSE (e.g. agile development process); • The standards could contain undetected contradictory requirements from one part to another. VSEs might not be able to make decisions about which requirements of the standards to implement; • The standards may degrade the efficiency of the development of projects (e.g. longer development time, higher overhead, higher cost); • The standards may reduce the flexibility of the management of projects by imposing too much control; • VSEs may not be eligible for contracts if they do not comply

	with the standards; • A VSE of a multi-national organization could be forced to put in place compliant processes which are different from those of other VSEs in the same organization, burdening that VSE with having to use the standards (e.g. VSEs may need 2 sets of processes: one to comply with national regulations and one to engage in development with other members of a multi-national company); • The VSEs may not be fluent enough in English to develop processes which are compliant with the standards; • VSEs may perceive the standards as a threat to innovation; • The standards may not be appropriate for a particular software domain (e.g. web development, game development); • The standards may impose practices which are different from/incompatible with a VSE's current practices (e.g. agile development, systems engineering practices, configuration management practices); • The standards do not take into account the requirements of one or more stakeholders (i.e. the standards developers did not know that these stakeholders existed (e.g. government regulatory agencies)); • VSEs have already invested resources in developing processes using another framework (e.g. the CMMI) and may not want to invest additional resources to comply with another framework; • VSEs cannot develop a product for one customer using one framework and a product for another customer using a different framework;
Not imposing the standards on VSEs (*laissez-faire*)	• Some VSEs may decide to ignore the standard, thereby slowly losing competitiveness; • Some VSEs may not know that such a standard exists, thereby missing an opportunity to become more competitive; • No government support is available to help VSEs adapt

	and implement the standards; • Each VSE would have to invest separately to implement the standard (e.g. purchase training courses, consulting); • Consulting companies would not invest in the development of expertise or courses to support a small number of VSEs; • Assessors may not be locally available to consult on, or assess, compliance with the standards; • Academia would not teach the standards in software engineering or computer science programs.

Table 8-1 Potential Negative Consequences of the ISO 29110 Standards

8.3 Measurement of the worldwide diffusion of the standards

Many researchers have noted that promising software tools and technologies, such as Computer-Aided Software Engineering Tools (CASE) and formal development techniques, have been developed, but were either not widely used or often abandoned (Green et al. 2000). For example, Iivari noted that, a year after their introduction, 70% of CASE tools have never been used by individuals in the IS organization, and only 5% are widely used, although not to the full capability of the tool (Iivari 1996).

A better understanding of these problems has led researchers to study the factors that drive or inhibit the adoption of an IT innovation, with the result that an IT diffusion model has been developed taking into account the human dimension as well as the technical dimension. A field study of a Software Process Improvement (SPI) technology was conducted, using the Personal Software Process (PSP) developed by Humphrey at the SEI. PSP is a 'self-improvement process to control, manage, and improve the way [software engineers] work.' (Humphrey 1997). The field study of industrial software practitioners was performed with a field survey of developers in multiple organizations (Green et al. 1998). One limitation of the survey was that only 12% of respondents were non-US practitioners. Therefore, since the respondents of the study are not representative of the future users of the ISO standards, the model may be culturally biased.

Since the SPI technology studied by Green, i.e. PSP, is quite close to the technology developed by WG24, the IT diffusion model could be used to guide the members of the Network of Support Centers in adopting the appropriate means to promote and accelerate the adoption of the ISO standards and deployment packages in VSEs worldwide.

The model developed by Green, illustrated in Figure 8-2, indicates that a key measure of success is the satisfaction of the users of a new IT technology. Satisfied users should influence the degree of use in the adopting organization (Green et al. 2006).

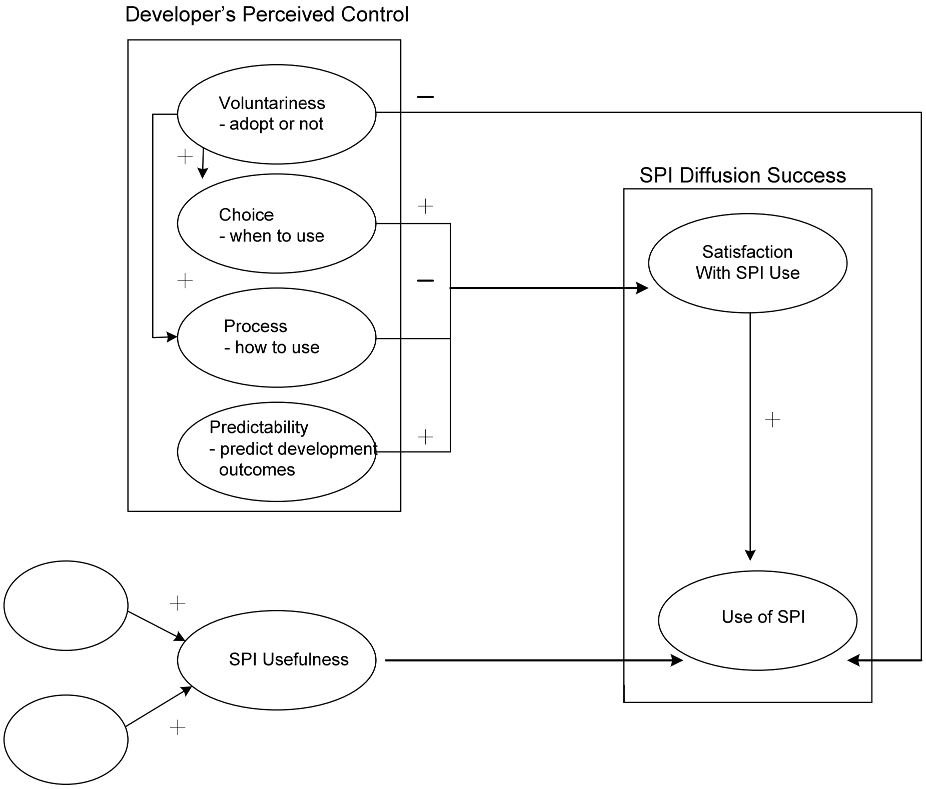


Figure 8-2 IT Diffusion Model (Green et al. 2006)

In a recent publication, Green (Green et al. 2006) offers a series of management guidelines for the selection and implementation of SPIs. The selection, training, and implementation guidelines are listed in Table 8-2. These guidelines could be used by the members of the Network of Support Centers in communicating locally to VSEs during the pilot projects of deployment packages and the deployment of the standards.

SPIs Selection Guidelines	Select SPIs that enable developers to predict effort and quality outcomes.
	Select SPIs that demonstrably increase software quality.
	Select SPIs that demonstrably improve developer productivity.
	Select SPIs that have a clear fit with the tasks and implementation environment on which they are to be used.
Training Guidelines	Train developers on how to determine when use of the SPIs is most appropriate.
	Train developers on the best process for using the SPI effectively.
	Demonstrate to developers the positive outcomes of SPI in terms of predictability or outcomes, increased software quality, and increased developer productivity.
	Set expectations for developers for continuous learning via SPI use.
Implementation Environment Guidelines	Involve developers in the adoption decision.
	Provide clear standards and practices for applying the SPI in the organization.
	Allow voluntariness for developers early in the implementation, but not later.
	Promote and assess both satisfaction and use as desired outcomes of SPI diffusion.

Table 8-2 Management Guidelines (Green et al. 2006)

The survey instrument developed by Green could be adapted, translated, and used to measure the adoption of the ISO standards in VSEs worldwide. To increase the number of respondents, the survey instrument can also be designed, like the author's the survey, to be completed on the Internet. It could be conducted about one year after the publication by the ISO of the standard and then on a yearly basis to measure the rate of adoption in different countries.

8.4 Development of courses for VSEs

In the previous chapter, the idea of establishing an education interest group (EIG) was presented. At a meeting of WG24 in India, four delegates volunteered to develop an initial set of courses. The following paragraphs describe the outline of an initial set of courses. The author used Bloom's taxonomy, as illustrated in Table 8-3, to describe the course content, since it is a well-known and frequently used classification of cognitive educational goals (Bloom 1956).

Bloom’s Taxonomy Level	Associated Keywords
Knowledge (K): Recall data	Defines, describes, identifies, knows, labels, lists, matches, names, outlines, recalls, recognizes, reproduces, selects, states
Comprehension (C): Understand the meaning, translation, interpolation, and interpretation of instructions and problems; state a problem in one’s own words.	Comprehends, converts, defends, distinguishes, estimates, explains, extends, generalizes, gives examples, infers, interprets, paraphrases, predicts, rewrites, summarizes, translates
Application (AP): Use a concept in a new situation or use an abstraction unprompted; apply what was learned in the classroom to novel situations in the workplace.	Applies, changes, computes, constructs, demonstrates, discovers, manipulates, modifies, operates, predicts, prepares, produces, relates, shows, solves, uses
Analysis (AN): Separate material or concepts into component parts so that its organizational structure may be understood; distinguish between facts and inferences.	Analyzes, breaks down, compares, contrasts, diagrams, deconstructs, differentiates, discriminates, distinguishes, identifies, illustrates, infers, outlines, relates, selects, separates
Synthesis (S): Build a structure or pattern from diverse elements; put parts together to form a whole, with emphasis on creating a new meaning or structure.	Categorizes, combines, compiles, composes, creates, devises, designs, explains, generates, modifies, organizes, plans, rearranges, reconstructs, relates, reorganizes, revises, rewrites, summarizes, tells, writes
Evaluation (E): Make judgments about the value of ideas or materials.	Appraises, compares, concludes, contrasts, criticizes, critiques, defends, describes, discriminates, evaluates, explains, interprets, justifies, relates, summarizes, supports

Table 8-3 Bloom’s Taxonomy (Bloom 1956)

The courses described below could be offered in the standard classroom format or through Webinars, and possibly as self-learning modules. For VSEs, it would be cheaper and more efficient to attend Webinar courses. Also, the Webinars could be offered worldwide to all participating VSEs of the Network of Support Centers.

The first course topics could be as follows:

- Introduction to ISO/IEC Software Engineering Standards
- Introduction to the ISO/IEC 29110 Standards, Technical Reports, and Deployment Packages for VSEs
- Development of a Software Engineering Process using ISO/IEC 29110 TR Part 5 – Engineering and Management Guide
- Software Development Using ISO/IEC 29110 TR – Engineering and Management Guide
- Self-Assessment of an ISO/IEC 29110-based Software Process
- Deployment of ISO/IEC 29110 Engineering and Management Guide in a VSE

These courses are briefly described in the next paragraphs.

8.4.1 Course 1– Introduction to ISO/IEC Software Engineering Standards

Figure 8-3 describes a course titled Introduction to ISO/IEC Software Engineering Standards. This course provides students with an introduction to the family of ISO/IEC Software Engineering Standards and describes the relationships between software engineering and systems engineering standards.

Title of Course: Introduction to ISO/IEC Software Engineering Standards

Course description
This course provides students with an introduction to the family of ISO/IEC Software Engineering Standards and describes the relationships between software engineering and systems engineering standards.

Objectives

	Taxonomy Level
Explain why ISO/IEC software engineering standards were developed	Comprehension
Explain the portfolio of ISO software and systems engineering standards and the relationships between systems engineering and software engineering ISO/IEC standards	Comprehension
Explain the ISO 9001 standards and associated guide for IT (ISO 90003)	Comprehension
Present the ISO/IEC 12207, 15289, 15504 standards	Comprehension
Show how compliance with an ISO/IEC standard is demonstrated using the ISO/IEC 15504 standard	Comprehension
Present the advantages and disadvantages of standards	Evaluation

Target Audience
The course is for anyone new to ISO/IEC software engineering standards or those needing a refresher on the subject, such as:

- Corporate engineering, manufacturing, and design staff
- Quality managers
- Government and public administration staff
- University faculty and students (engineering, computer science, business, public policy, law)
- Non-governmental organizations concerned with trade
- Standards development organizations staff

Strategy

- Presentations and class discussions
- Exercises

Figure 8-3 Introduction to ISO/IEC Software Engineering Standards

8.4.2 Course 2 – Introduction to the ISO/IEC 29110 Standards, Technical Reports. and Deployment Packages for VSEs

Figure 8-4 describes a course titled: Introduction to the ISO/IEC 29110 Standards, Technical Reports, and Deployment Packages for VSEs. This course explains the justification and steps that led to the development of ISO/IEC 29110 Standards, Technical Reports, and Deployment packages for VSEs.

Title of Course: Introduction to the ISO/IEC 29110 Standards, Technical Reports, and Deployment Packages for VSEs

Course description
This course explains the justification and steps that led to the development of ISO/IEC 29110 Standards, Technical Reports, and Deployment packages for VSEs.

Objectives

	Taxonomy Level
Present the characteristics of VSEs, and their problems and needs regarding standards	Comprehension
Present the results of the survey conducted to elicit and validate the set of needs and problems of VSEs about the use of standards	Comprehension
Present the justification and motivation that led to the establishment of an ISO working group dedicated to VSEs	Comprehension
Present the list of requirements derived by the ISO WG to develop standards for VSEs	Comprehension
Explain the strategy of the WG to develop standards and guidelines for VSEs	Comprehension
Explain the concept of Profiles and International Standardized Profiles (ISPs)	Comprehension
Explain the set of ISO/IEC 29110 standards and technical reports for the Generic Profile Group	Comprehension
Explain the concept of Deployment Packages	Comprehension
Explain the set of Deployment Packages of the Generic Profile Group	Comprehension

Target Audience
The course is for anyone new to ISO/IEC 29110 standards for VSEs, such as:

- Corporate engineering, manufacturing, and design staff
- Quality managers
- Government and public administration staff
- University faculty and students (engineering, computer science, business, public policy, law)
- Non-governmental organizations concerned with trade
- Standards development organizations staff

Strategy

- Presentations and class discussions
- Exercises
- Case studies (e.g. from pilot projects)

Figure 8-4 Introduction to the ISO/IEC 29110 Standards, Technical Reports, and Deployment Packages for VSEs (adapted from[71])

8.4.3 Course 3 – Development of a Software Engineering Process using ISO/IEC 29110 TR Part 5 – Engineering and Management Guide

Figure 8-5 describes a course titled Development of a Software Engineering Process using ISO/IEC 29110 TR Part 5 – Engineering and Management Guide. In this course, students will use the implementation and management tasks and product descriptions described in Part 5 to develop a usable set of processes using a selected process notation.

Title of Course: Development of a Software Engineering Process using ISO/IEC 29110 TR Part 5 – Engineering and Management Guide

Course description
Students use the implementation and management tasks and product descriptions described in Part 5 to develop a usable process, using a selected process notation.

[71] http://www.ip-shield.com/products_services.aspx#StandardsDevelopingOrganizations

Objectives

	Taxonomy Level
Explain ISO/IEC 29110 TR Part 5 – Engineering and Management Guide	Comprehension
Explain the notation(s) used to document a process	Comprehension
Document the process using the selected notation (e.g. deployment package notation, ETVX notation, etc.)	Synthesis
Conduct a walk-through of the documented process	Evaluation
Present the documented process to other students	Evaluation
Present the set of available Deployment Packages	Evaluation

Target Audience
The course is for anyone new to ISO/IEC 29110 standards for VSEs, or those needing a refresher on the subject, such as:

- Corporate engineering, manufacturing, and design staff
- Quality managers
- Government and public administration staff
- University faculty and students (engineering, computer science, business, public policy, law)
- Non-governmental organizations concerned with trade
- Standards development organizations staff

Strategy

- Presentations and class discussions
- Exercises
- Case studies (e.g. from pilot projects)

Figure 8-5 Development of a Software Engineering Process using ISO/IEC 29110 TR Part 5 – Engineering and Management Guide

8.4.4 *Course 4 – Software Development Using ISO/IEC 29110 TR – Engineering and Management Guide*

Figure 8-6 describes a course titled Software Development Using ISO/IEC 29110 TR – Engineering and Management Guide. In this course, students will use a process, compliant with Part 5, to develop a software component and propose improvements to deployment packages and Part 5.

Title of Course: Software Development Using ISO/IEC 29110 TR – Engineering and Management Guide

Course description
Students use processes compliant with Part 5 to develop a software component and propose improvements to deployment packages and Part 5.

Objectives

	Taxonomy Level
Use standards and technical reports for VSEs to develop a software application	Application
Use the set of Deployment Packages for the Generic Profile Group	Application
Evaluate the process and Deployment Packages used	Evaluation
Propose improvements to the Deployment Packages and Part 5	Evaluation

Target Audience
The course is for anyone new to ISO/IEC 29110 standards for VSEs, or those needing a refresher on the subject, such as:

- Managers and software developers of VSEs
- Assessors
- University faculties of computer science and software engineering departments
- University students of computer science and software engineering departments

Strategy

- Presentations and class discussions
- Team project using the set of Deployment Packages for the Generic Profile Group
- Case studies (e.g. from pilot projects)

Figure 8-6 Course Description – Software Development Using ISO/IEC 29110 TR – Engineering and Management Guide

8.4.5 Course 5 – Self-Assessment of an ISO/IEC 29110-based Software Process

Figure 8-7 describes a course titled Self-Assessment of an ISO/IEC 29110-based Software Process. This course will use the ISO/IEC standards, technical reports, and deployment packages for VSEs to assess the compliance of a software development process with the 29110 Standard.

Title of Course: Self-Assessment of an ISO/IEC 29110-based Software Process

Course description

This course uses the ISO/IEC 29110 standards, technical reports, and deployment packages for VSEs to assess the compliance of a software development process with the 29110 Standard and propose improvements.

Objectives

	Taxonomy Level
Present the ISO/IEC 29110 Part 3 Technical Report – Assessment Guide	Comprehension
Present the self-assessment deployment package	Comprehension
Assess an existing process using the deployment package	Evaluation
Write an assessment report of the process assessed	Synthesis
Propose improvements to process assessed and to deployment packages	Evaluation
Propose improvements to the self-assessment deployment package	Evaluation
Propose improvements to ISO/IEC 29110 Part 3	Evaluation

Target Audience

The course is for anyone who wants to perform compliance assessment, such as:

- Managers and software developers of VSEs
- Assessors
- University faculties of computer science and software engineering departments
- University students of computer science and software engineering departments
- Consultants performing technology transfer

Strategy

- Presentations and class discussions
- Team project using the Self-assessment Deployment Package
- Exercises
- Case studies (e.g. from pilot projects)

Figure 8-7 Course Description – Self-Assessment of an ISO/IEC 29110-based Software Process

8.4.6 Course 6 – Deployment of ISO/IEC 29110 Engineering and Management Guide in a VSE

Figure 8-8 describes a course titled Conduct Pilot Projects to Transfer the ISO/IEC 29110 Standards to a VSE. This course will use the ISO/IEC 29110 standards, technical reports, and deployment packages for VSEs to deploy a set of practices and conduct a pilot project in a VSE using a deployment package.

Title of Course: Deployment of ISO/IEC 29110 Engineering and Management Guide in a VSE

Course description
This course will use the ISO/IEC 29110 standards, technical reports, and deployment packages for VSEs to deploy a set of practices and conduct a pilot project in a VSE using a deployment package.

Objectives

	Taxonomy Level
Perform an assessment of the VSE to identify a set of practices to be deployed	Evaluation
Conduct pilot project using the Pilot Project Deployment Package	Application
Produce report on the pilot project	Application
Propose improvements to development process of the VSE	Evaluation
Propose improvements to Pilot Project Deployment Package	Evaluation
Propose improvements to ISO/IEC 29110 standards and technical reports	Evaluation

Target Audience
The course is for anyone who wants to perform a technology transfer to a VSE, such as:

- Managers and software developers of VSEs
- University faculties of computer science and software engineering departments
- University students of computer science and software engineering departments
- Consultants performing technology transfer

Strategy

- Presentations and class discussions
- Team project using the Pilot Project Deployment Project
- Exercises (TBD)
- Case studies (e.g. from pilot projects)

Figure 8-8 Course Description – Deployment of ISO/IEC 29110 Engineering and Management Guide in a VSE

8.5 Development of self-learning and e-learning course modules

In many VSEs, staff have learned software engineering by trial and error. We propose that a set of self-learning modules be developed to facilitate learning in missing or weak areas of knowledge. The IEEE Computer Society has developed a certification program, titled Certified Software Development Associate (CSDA), which is intended to be the first step to becoming a software development professional. To earn the CSDA credential, an individual must pass a comprehensive examination, in which the candidate's knowledge of software engineering is assessed. The CSDA product allows individuals to perform a self-assessment of their knowledge before taking the exam, based on the Software Engineering Body of Knowledge (ISO 2005a). The SWEBOK is divided into fifteen knowledge domains, as listed in Table 8-4.

Content Domain I: Software Requirements **Content Area 1: Software Requirements Fundamentals** Content Area 2: Requirements Process Content Area 3: Requirements Elicitation Content Area 4: Requirements Analysis Content Area 5: Requirements Specification Content Area 6: Requirements Validation Content Area 7: Practical Considerations **Content Domain II: Software Design** Content Area 1: Software Design Fundamentals Content Area 2: Key Issues in Software Design	**Content Domain X: Software Quality** Content Area 1: Software Quality Fundamentals Content Area 2: Software Quality Management Processes Content Area 3: Practical Considerations **Content Domain XI: Software Engineering Professional Practices** Content Area 1: Professionalism Content Area 2: Codes of Ethics Content Area 3: Group Dynamics/Psychology Content Area 4: Communication Skills Content Area 5: Intellectual Property,

Content Area 3: Software Structure and Architecture
Content Area 4: Human-Computer Interface Design
Content Area 5: Software Design Quality Analysis and Evaluation
Content Area 6: Software Design Notations
Content Area 7: Software Design Strategies and Methods

Content Domain III: Software Construction

Content Area 1: Software Construction Fundamentals
Content Area 2: Managing Construction
Content Area 3: Practical Considerations
Content Area 4: Construction Tools
Content Area 5: Construction Technologies
Content Area 6: Product Documentation

Content Domain IV: Software Testing
Content Area 1: Software Testing Fundamentals
Content Area 2: Test Levels
Content Area 3: Test Techniques
Content Area 4: Human-Computer Interface Testing and Evaluation
Content Area 5: Test-Related Measures
Content Area 6: Test Process

Module 2: Software Maintenance and Management

Content Domain V: Software Maintenance
Content Area 1: Software Maintenance Fundamentals
Content Area 2: Software Maintenance Process
Content Area 3: Key Issues in Software Maintenance
Content Area 4: Techniques for Maintenance

Content Domain VI: Software Configuration Management
Content Area 1: Management of the SCM Process
Content Area 2: Software Configuration Identification
Content Area 3: Software Configuration Control
Content Area 4: Software Configuration Status Accounting
Content Area 5: Software Configuration Auditing
Content Area 6: Software Release Management and Delivery
Content Area 7: Software Configuration Management Tools

Content Domain VII: Software Engineering Management
Content Area 1: Initiation and Scope Definition
Content Area 2: Software Project Planning
Content Area 3: Software Project Enactment
Content Area 4: Review and Evaluation
Content Area 5: Closure
Content Area 6: Software Engineering Measurement
Content Area 7: Software Management Tools

Confidentiality, Security

Content Domain XII: Software Engineering Economics
Content Area 1: Software Engineering Economics Fundamentals
Content Area 2: For-Profit Decision Making
Content Area 3: Not-for-Profit Decision Making
Content Area 4: Present Economy
Content Area 5: Estimation, Risk, and Uncertainty
Content Area 6: Multiple-Attribute Decisions

Module 4: Software Engineering Foundations

Content Domain XIII: Computing Foundations
Content Area 1: Programming Fundamentals
Content Area 2: Use and Support for Enabling Techniques, Such as Abstraction
Content Area 3: Data Structures/Representation, Algorithms, and Complexity
Content Area 4: Problem-Solving Techniques
Content Area 5: Computer Organization
Content Area 6: Basic User Human Factors
Content Area 7: Basic Developer Human Factors
Content Area 8: Operating System Basics
Content Area 9: Database Basics and Data Management
Content Area 10: Network Communication Basics
Content Area 11: Distributed and Parallel Computing
Content Area 12: Concepts of Programming Languages
Content Area 13: Debugging Tools and Techniques
Content Area 14: Secure Coding

Content Domain XIV: Mathematical Foundations
Content Area 1: Basic Logic
Content Area 2 Sets, Functions, and Relations
Content Area 3: Proof Techniques
Content Area 4: Basic Counting
Content Area 5: Graphs and Trees
Content Area 6: Discrete Probability
Content Area 7: Grammars
Content Area 8: Finite State Machines, Regular Expressions
Content Area 9: Numerical Precision, Accuracy, and Errors
Content Area 10: Number Theory
Content Area 11: Algebraic Structures

Content Domain XV: Engineering Foundations
Content Area 1: Empirical Methods and Experimental Techniques
Content Area 2: Statistical Analysis
Content Area 3: System Development
Content Area 4: Engineering Design
Content Area 5: Root Cause Analysis
Content Area 6: Standards
Content Area 7: Modeling, Simulation, and Conceptual Prototyping

Module 3: Software Engineering Supporting Practices **Content Domain VIII: Software Engineering Process** Content Area 1: Process Implementation and Change Content Area 2: Process Definition Content Area 3: Process Assessment Content Area 4: Process and Product Measurement Content Area 5: Software Engineering Process Tools **Content Domain IX: Software Engineering Methods** Content Area 1: Modeling Content Area 2: Types of Models Content Area 3: Analysis Content Area 4: Development Methods	Content Area 8: Tool and Platform Selection Content Area 9: Measurement Content Area 10: Theory of Measurement Content Area 11: Goal Question Metric (GQM) Paradigm

Table 8-4 Content of the CSDA Knowledge Modules (IEEE 2009)

Use of this feature could help the software staff or management of a VSE identify the candidate's weaknesses in the various areas of the SWEBOK. Once weaknesses have been identified, the individual can elect to attend a formal class, undertake readings on his own, or acquire the knowledge through a self-learning module. The development of a self-learning module could also be designed to provide the knowledge required for the successful application of a deployment package. For example, it is well known that many software developers face difficulties in the architecture and detailed design phases of a project. A self-learning module, provided at very low cost, or even at no cost, could greatly enhance the abilities of a software developer who cannot attend formal classes, but could acquire the required knowledge at his own pace.

8.6 Development of plug-in modules to support ISO standards

A set of plug-in modules similar to the RUP Plug-In could be developed. The RUP Plug-In provides guidance for software development teams who use RUP and are looking for effective ways to apply them to small teams and small projects. The plug-in modules would be developed to facilitate the utilization of the set of Deployment Packages, and could also be designed to be integrated into a suite of CASE tools similar to the IBM Rational suite of tools.

8.7 Establishment of a software engineering support center for VSEs at the ÉTS

The author proposed to the senior management of the ÉTS in early 2009 the establishment of a center dedicated to VSEs. The proposed mission for the center is to accelerate technology transfer to small and very small structures in Québec developing software products or software-based systems, or to provide IT services to make them more competitive, both at the national level and internationally, by developing and deploying software engineering practices tailored to their needs.

To fulfill this mission, the center would have the following objectives:

- Identify, promote, and disseminate best practices in software engineering and services for very small entities;
- Accelerate the process of technology transfer in software engineering for VSEs;
- Provide information and technical and strategic information to managers of VSEs, outsourcers, and Government of Québec agencies;
- Participate in the development of international standards for VSEs;
- Promote international standards for VSEs in Québec;
- Promote research in software engineering for VSEs;
- Promote training and development courses on ISO standards for VSEs.

The table of contents (in French) of a business plan developed by the author for the establishment of a support center for VSEs at the ETS is illustrated in Figure 8-9. The business plan can be found in Appendix B (in French).

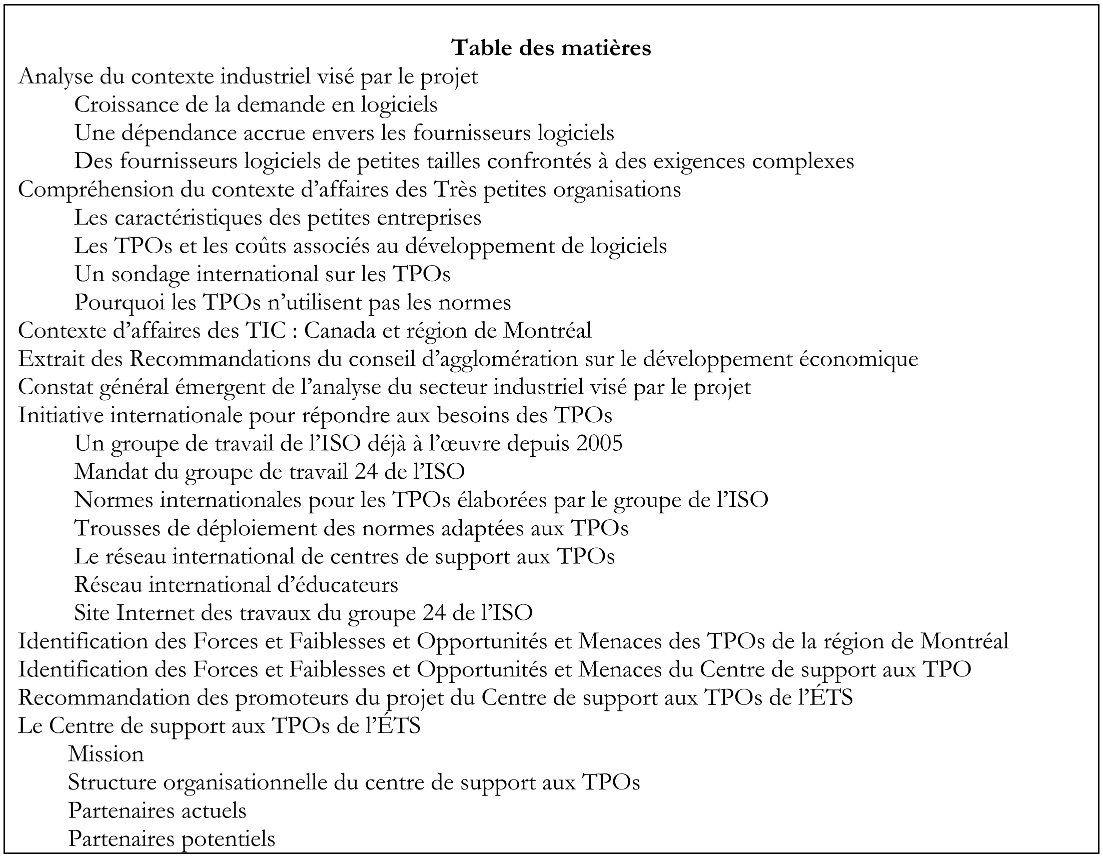


Table des matières

- Analyse du contexte industriel visé par le projet
    - Croissance de la demande en logiciels
    - Une dépendance accrue envers les fournisseurs logiciels
    - Des fournisseurs logiciels de petites tailles confrontés à des exigences complexes
- Compréhension du contexte d'affaires des Très petites organisations
    - Les caractéristiques des petites entreprises
    - Les TPOs et les coûts associés au développement de logiciels
    - Un sondage international sur les TPOs
    - Pourquoi les TPOs n'utilisent pas les normes
- Contexte d'affaires des TIC : Canada et région de Montréal
- Extrait des Recommandations du conseil d'agglomération sur le développement économique
- Constat général émergent de l'analyse du secteur industriel visé par le projet
- Initiative internationale pour répondre aux besoins des TPOs
    - Un groupe de travail de l'ISO déjà à l'œuvre depuis 2005
    - Mandat du groupe de travail 24 de l'ISO
    - Normes internationales pour les TPOs élaborées par le groupe de l'ISO
    - Trousses de déploiement des normes adaptées aux TPOs
    - Le réseau international de centres de support aux TPOs
    - Réseau international d'éducateurs
    - Site Internet des travaux du groupe 24 de l'ISO
- Identification des Forces et Faiblesses et Opportunités et Menaces des TPOs de la région de Montréal
- Identification des Forces et Faiblesses et Opportunités et Menaces du Centre de support aux TPO
- Recommandation des promoteurs du projet du Centre de support aux TPOs de l'ÉTS
- Le Centre de support aux TPOs de l'ÉTS
    - Mission
    - Structure organisationnelle du centre de support aux TPOs
    - Partenaires actuels
    - Partenaires potentiels

Figure 8-9 Table of contents of the business plan for the establishment of a support center for VSEs at the ÉTS (Laporte 2009b)

8.8 Systems engineering for small and very small entities

The INCOSE (International Council on Systems Engineering) has been developing and improving a systems engineering handbook for many years. The handbook defines the practices of systems engineering (SE) for students and practicing professionals (INCOSE 2007). The handbook is also consistent with ISO/IEC 15288 – Systems engineering – System life cycle processes standard (ISO 2008c).

A paper about the work of WG24, co-authored by the author of this thesis, had been presented in 2008 at an INCOSE conference (Laporte 2008). This presentation triggered the idea, of the INCOSE Board of Directors, of setting up a working group to develop a set of guidelines, similar to the guidelines developed by WG24, for developers of products in very small and medium-sized enterprises.

A new working group was mandated in 2009, by INCOSE, to apply systems engineering to product development for small and very small entities. The working group, created in April 2009, co-chaired by the author, is composed of eleven members from Canada, France, Germany, and the US.

At the first meeting of the working group, the author proposed an approach similar to that developed by WG24, i.e. the conducting of a survey, the development of a set of requirements, the creation of profiles and deployment packages, and the conducting of pilot projects. The members of the working group agreed with this proposition.

The initial goals of the working group were (INCOSE 2009) to:

- Improve or make product development efficient by using systems engineering methodology
- Elaborate tailored practical guidance to apply to very small and medium-sized enterprises (VSMEs) in the context of the prime contractor or subcontractor of commercial products
- Contribute to standardization

The initial set of deliverables is the following:

- Roadmaps and deployment packages (process, check list, templates, training support, coverage matrices with standards)
- Lessons learned from pilot projects
- Case studies
- Mapping of tailored activities with the INCOSE Systems Engineering (SE) Handbook and ISO Standards (e.g. ISO/IEC 15288), as well as other recommended frameworks (e.g. CMMI)
- Supplements to the INCOSE Systems Engineering Handbook
- Technical communication with INCOSE membership through papers, presentations, and Webinars

The author, while participating in this working group, will try to make sure that the work of the INCOSE working group is harmonized with the future ISO standards for VSEs, such that entities already using ISO standard 29110 can easily integrate the newly developed guidelines for systems engineering into their practices.

Finally, it is possible that ISO/IEC JTC1 SC7 will decide to launch an ISO working group, similar to WG24, for the development of roadmaps and profile groups for the ISO/IEC 15288 systems engineering standard for VSEs.

8.9 Development of profiles and deployment packages for ISO/IEC 20000

It has been mentioned in a previous chapter that work has started toward the development of an initial profile and associated deployment modules for the ISO/IEC 20000 standard.

The author, in collaboration with ISO/IEC 20000 experts, could pursue the development of roadmaps, profiles, and deployment packages similar to the work performed by WG24. There are thousands of VSEs worldwide that could apply this ISO standard to their daily activities. But, they are facing issues similar to those facing software developers in VSEs.

8.10 Application of standards for VSEs in courses of the ÉTS

In Appendix A, the author presented two graduate courses where students are required to carry out a project in an organization. Once the standards for VSEs have reached the final revision stage, the students will be taught about the new standards and will use them, and the deployment packages, to carry out their project.

The graduate students, who will be under my supervision for their research project, having already taken the two graduate courses described in Appendix A (i.e. MGL 805, MGL 950), and apply this knowledge to their project (MGL 940) in an organization.

Finally, the undergraduate course in software quality assurance (LOG 330) will also teach the new standards for VSEs and require the students to use the ISO 29110 standards in their team project assignment.

8.11 Application of the WG24 approach to other software domains

So far, WG24 has developed a Profile Group for developers of generic software applications. There are many domains, or application fields, that could benefit from the development of a specific profile group, such as scientific software, embedded software, real-time software, critical software, and software maintenance. A few of these domains are discussed below.

8.11.1 Scientific software development

Examples of scientific software are nuclear reactions, and high-performance computing to simulate physical phenomena, such as earthquakes, nuclear reactions, and climate changes. Some applications require thousands of processors on a single project.

The author had a graduate student develop software processes for the wind turbine research laboratory at the ÉTS (the project is described in Appendix A). One lesson learned from this project was that, from the point of view of the researcher, software is only a research tool. It is most often not an objective of a research team to commercialize the software developed during a research project.

While most software engineers expect an up-front list of requirements before starting to develop software, a researcher expects requirements to emerge during the research project. As the requirements emerge, graduate students develop the required functionalities. As the project evolves, new requirements are coded, most probably by new graduate students, which are often poorly documented or not documented at all. With regard to the software engineering process, the needs of researchers are therefore very different from those of most VSEs.

Some of the characteristics of scientific software are (adapted from Basili et al. 2008) are the following:

- Scientists either write or customize much of the code themselves, or occasionally have their graduate students perform these tasks, since the average software developer does not understand the application domain;
- Research laboratories are often multidisciplinary environments;
- Developers receive their software training from other scientists;
- Many software programs are not designed to be large, but grow and evolve in parallel with the research project;
- Scientists are not typically trained in software engineering methodologies, their expertise residing rather in the science of their field of research;
- Requirements are mostly emergent, and the emergence of requirements is intertwined with evaluation and testing is cursory;
- The life cycle could be long, e.g. a 30-year life cycle for some nuclear simulation software.

At first glance, scientists may be receptive to using a 'light' process similar to the Entry profile presented in a previous chapter. This profile was designed for a project having a size of about 6 person-months, which is quite representative of what a graduate student can accomplish by working in a research laboratory. Once the practices associated with this profile are implemented in a research laboratory, the scientist should be in a position to assess whether or not this profile meets his needs. Otherwise, if needed, the Basic Profile practices can then be implemented in the laboratory.

A graduate student is currently developing a software process, using the Basic Profile, for a research laboratory (Centre de recherche en neuropsychologie et cognition) at the Université de Montréal (Guillemot 2009). Once this project has been completed, a post-mortem will be conducted to evaluate the costs and benefits of the approach.

8.11.2 Embedded system development

An embedded computer system is one that is part of a larger system and performs some of the requirements of that system. An example of an embedded computer system is a computer system used in an aircraft or rapid transit system (ISO/IEC 24765). The world market for embedded systems is approximately 160 billion Euros, involving approximately 3 billion embedded units delivered per year and a compound annual growth of 9 percent (Ebert et al. 2009). As illustrated in Figure 8-10, both the size of the software and the number of applications deployed each year vary by many orders of magnitude. For example, some cars have 100 Mbytes of software running, and a car may contain from 30 to 70 embedded systems that communicate with one another (Ebert et al. 2009).

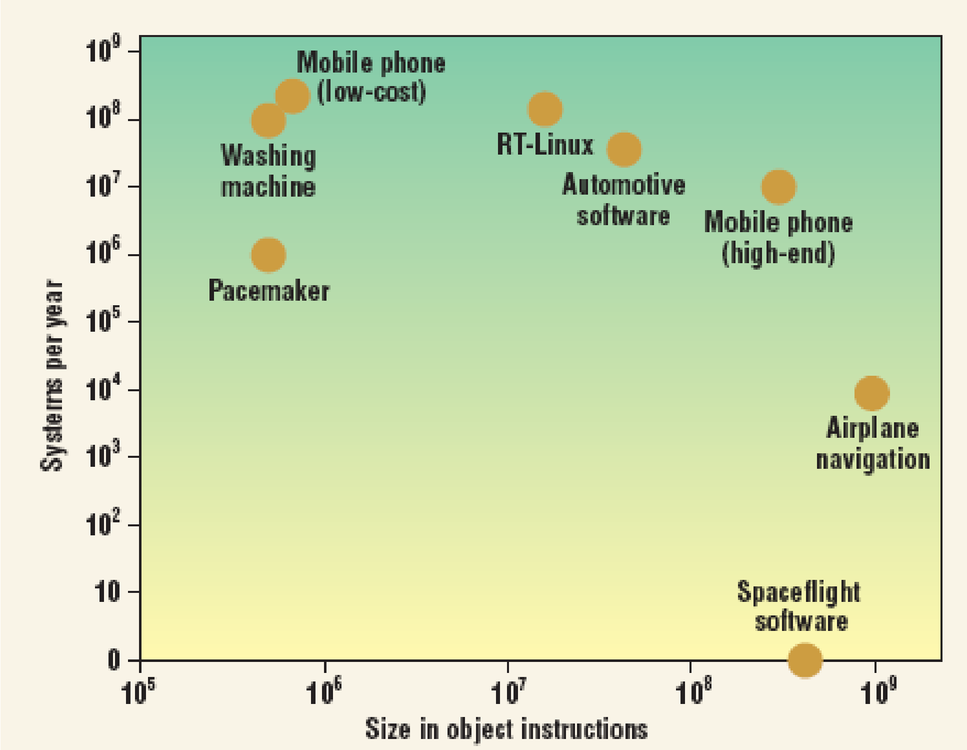


Figure 8-10 Embedded software size and deployment (Ebert et al. 2009)

Some applications, like automotive or airplane navigation systems, are already regulated by standards. Others, like washing machines and home electronic devices (digital cameras or video recorders), are not. The future ISO standards could be a nice niche for some of these applications.

Among the characteristics of embedded systems are (adapted from Mäkäräinen 2000) the following:

- Consist of computers affiliated with products for implementing control, communication, usage, and other intelligent functions;

- Utilize mechanics, electronics, and hardware and software technologies that are closely related to one another;
- Contain hardware and software components which are often developed concurrently;
- Often have no keyboard and a limited display;
- Often have real-time requirements, i.e. correctness is partially a function of time;
- Must be robust, i.e. their behavior must always be controlled, even during system failure;
- Often have time, memory, or power-consumption constraints;
- Use software that is often developed in assembly language and in very primitive software engineering environments, with almost no support for testing, code measurement, etc.;
- Locating problems is difficult because of the tight relationship between hardware and software;
- Making changes to some of them after delivery is impossible or very costly, as some they might be in an inaccessible environment, or released in a huge volume, for example;
- Frequently, they provide no access to the target system for developers, in which case development is done in a development/simulation environment that behaves differently from the target environment, since some aspects of the target environment are hard to simulate;
- Locating errors is very time-consuming, because the individual who found the error is typically not the one who reports it;
- New versions of the development tools are typically delivered several times during the long maintenance phase of some systems, such as a subway system which may operate for up to 40 or 50 years;
- Traceability information between a change request and the modified document and associated software is often missing;

- Because some systems have a very long life, knowledge must be managed so that the organization remembers how to use the languages and tools of very old components.

Use of the Basic Profile Group is proposed as the baseline for developing a roadmap that meets the requirements of embedded software systems.

8.11.3 Critical software development

Many embedded systems also have critical software components. Critical software is defined as software which, if it fails, could have an impact on safety or cause major financial or social loss (IEEE 610.12 1990).

A generic standard, IEC 61508, addresses the functional safety of systems, primarily those developed using electrical, electronic, and computer technology (IEC 1998). This standard incorporates four Safety Integrity Levels, or SILs. An SIL is defined as the likelihood of a safety-related system satisfactorily performing the required safety functions under all the stated conditions, within a standard period of time. The definitions of the four SILs of the IEC 61508 standard are given, as illustrated in Table 8-5, in terms of the probability of failure, or, in the high demand case, as the probability of failure per hour of operation.

Safety Integrity Level (SIL)	**Continuous or high demand operation**
1	>= 10-6 to 10-5
2	>= 10-7 to 10-6
3	>= 10-8 to 10-7
4	>= 10-9 to 10-8

Table 8-5 Probability of failure corresponding to each SIL (IEC 1998)

A proposed strategy to address this domain could be the addition of practices to the current Generic Profile Group standards, guides, and deployment packages, having in mind compliance with the requirements of critical standards, such as the DO-178B for the aerospace domain (RTCA 1992).

8.12 Conclusion

The purpose of this thesis, submitted for the Degree of Doctor of Philosophy by Published Work to the Université de Bretagne Occidentale, was to demonstrate that the author’s contributions have had an impact in software engineering (SE) and to the development, deployment, and utilization of international SE standards, specifically those for VSEs, in Canada and internationally.

The impact of the author's published contributions has been shown in the thesis, which documents:

- a large number of published works and contributions covering a 29-year period;
- contributions to the military, to industry and academia;
- contributions to the establishment of the Applied Software Engineering Center in Montréal;
- contributions of the author as the editor of Working Group 24 of ISO/IEC JTC1 SC7, and as the Canadian delegate to that group, which is mandated to develop International Standards (ISs) and Technical Reports (TRs) for VSEs;
- the broad acceptance of the draft ISs and TRs developed by Working Group 24;
- contributions, beyond those as editor of Working Group 24, to accelerate and facilitate the diffusion of ISs and TRs to VSEs worldwide, and their utilization by those entities;
- the infrastructure developed to accelerate and facilitate the deployment and utilization of future standards by VSEs worldwide;
- contributions to the establishment of a technology transfer center for VSEs at the École de technologie supérieure (Montreal);
- contributions of the author as the co-editor of Working Group 20 of ISO/IEC JTC1 SC7, and as the Canadian delegate to that group;
- the research performed by SE graduate students under the author's supervision or co-supervision.

Table 8-6 and Table 8-7 list the contributions discussed in this thesis. Appendix A describes those accomplishments in more detail to enable the reader to better understand the author's involvement and motivation in the development of international standards for VSEs.

Topic	Edited Book, Book Chapter, Thesis	Refereed Journal Paper	Edited Journal Paper	Refereed Conference Proceedings Paper
Software and Systems Engineering for industry	(Laporte 1980) (Laporte 1986a) (Laporte 1995b) (April et al. 2005) (April et al. 2009)	(Laporte et al. 1986a) (Laporte et al. 1988) (Laporte et al. 1989) (Laporte et al. 1999d) Laporte et al. 1996b) Laporte et al. 1997b) (April et al. 2005) (April et al. 2006a) (April et al. 2006b) (Berrhouma et al. 2009) (Côté et al. 2004a) (Côté et al. 2005) (Corrales et al. 2007) (Laporte et al. 2003) (Laporte et al. 2005d) (Laporte et al. 2008f)	(Laporte 1985) (Laporte et al. 1998d) (Laporte et al. 1999a) (Laporte 2004a)	(Laporte et al. 1985) (Laporte et al. 1986b) (Laporte et al. 1986c) (Laporte 1994) (Laporte 1996) (Laporte et al. 1996) (Laporte et al. 1997a) (Laporte 1998) (Laporte et al. 1998a) (Laporte et al. 1998b) (Laporte et al. 1998c) (Laporte et al. 1999b) (Laporte et al. 1999c) (Laporte 2004b) (Laporte 2007e) (Laporte et al. 2007b) (Laporte et al. 2007c) (Bourque et al. 2004) (Corrales et al. 2007) (Côté et al. 2004b) (Suryn et al. 2002) (Suryn et al. 2004) (Laporte et al. 2008f)
Software Engineering for Education	(April and Laporte 2010[72])	(Laporte et al. 2007e)		(Laporte et al. 2007d)
Software Engineering for Very Small Entities (VSEs)	(Laporte et al. 2008c)	(Laporte 1995) (Laporte et al. 2005a) (Laporte et al. 2006a) (Laporte 2007a) (Laporte et al. 2007a) (Laporte et al. 2008a) (Laporte et al. 2008b) (Deniger et al. 2006) (Desfossés et al. 2008) (Habra et al. 2008) (Rheault et al. 2006)	(Laporte et al. 2006d) (Laporte et al. 2006e) (Laporte et al. 2007a)	(Laporte 2005a) (Laporte et al. 2005b) (Laporte et al. 2006b) (Laporte et al. 2006c) (Laporte et al. 2008d) (Laporte et al. 2008e) (Laporte 2007d) (Desharnais et al. 2007) (Stambollian et al. 2006)

Table 8-6 Refereed or Edited Published Works

[72] In preparation

Topic	Contribution to ISO Standards	Contribution to ISO Technical Reports	Technical Reports[73] and Contribution to Technical Reports	Keynote Speaker and other Communications
Software and Systems Engineering for industry			(SECOR 1990a) (SECOR 1990b) (SECOR 1992) **(**Laporte et al. 1993)	(Laporte 1981a) (Laporte 1981b) (Laporte 1981c) (Laporte 1981d) (Laporte 1982) (Laporte 1986a) (Laporte 1986b) (Laporte 1991) (Laporte 1992) (Laporte 1997) (Laporte 1999a) (Laporte 1999b) (Laporte 1995a) (Laporte 2001)
Software Engineering for Education	(ISO/IEC 2008a)	(ISO/IEC 2005a)		
Software Engineering for Very Small Entities (VSEs)	(ISO/IEC 2009b) (ISO/IEC 2009d)	(ISO/IEC 2009a) (ISO/IEC 2009c) (ISO/IEC 2009e)	(Laporte et al 2005c) (Laporte et al. 2009g) (Laporte 2009b) (Laporte 2009c) (Laporte 2009d) (BAR-026-2005a) (BAR-026-2005b) (BAR-026-2005c) (BAR-026-2005d) (BAR-026-2005e) (BAR-026-2005f) (BAR-026-2005g) (BAR-026-2005h) (BAR-026-2005i) (Vasquez 2009) (Arenas 2009) (Alexandre et al. 2009) (Palza 2009) (Guillemot 2009) (Sanyakorn 2009) (O'Connor 2009) (Varkoi 2009)	(Laporte 2006a) (Laporte 2006b) (Laporte 2007b) (Laporte 2007c) (Laporte 2008a) (Laporte 2008b) (Laporte 2009a)

Table 8-7 Other Contributions

In Chapter 2, industry trends and challenges were set out. The characteristics of VSEs were presented to better understand why they need standards adapted to their needs. This chapter covered phase 1 of the 6-phase model of the innovation process developed by Rogers on the recognition of needs and problems.

[73] Deployment Packages are included in this category

Phase 2, on basic and applied research, was covered in chapter 3, titled Overview of Process Improvement Initiatives for VSEs, and in chapter 5, which described the survey conducted to gain a better understanding of the problems and needs of VSEs regarding their use of standards.

Phase 3, on development, was covered in chapter 4, titled The Development Process of International Standards, and in chapter 6, titled The Development of International Standards for VSEs.

Phase 4, on commercialization, and phase 5, on diffusion and adoption, were covered in chapter 7, titled Development of the Means to Accelerate the Adoption and Utilization of International Standards by VSEs.

The final phase, phase 6, on consequences, was covered in this chapter.

Further work will be needed to develop additional profile groups, standards, guides, and deployment packages for systems engineering, for additional software application domains, such as embedded systems, and to enhance the productivity and quality of software developed and maintained by VSEs.

The thesis demonstrates that the author has contributed to software engineering and to the development and utilization of international software engineering standards, specifically for VSEs, in Canada and internationally.

References

(Abou 2006) Abou E Fattah, Mounir, *Projet d'amélioration de la compétitivité des petites entreprises Québécoises de développement et de maintenance de logiciels*, Masters Thesis, École de technologie supérieure, Montréal, Canada, Project co-directed by Professor J-M Desharnais and C.Y. Laporte of ÉTS, 2006.

(Abran et al. 2004) Abran, A., and Moore, J.W. (Exec. Eds.), Bourque, P. and Dupuis, R. (Eds.) *Guide to the Software Engineering Body of Knowledge*, 2004 Version, IEEE Computer Society Press and to be published as ISO Technical Report 19759, 2004, (Available free of charge at www.swebok.org and)

(AFNOR 2003) *Ingénierie systèmes, Processus de vie des systèmes*, Norme Française, AFNOR, NF Z 67-288, 2003.

(Alexandre et al. 2009) Alexandre, S., Laporte, C.Y., Requirements Analysis Deployment Package, Technical Report, Laporte et al. (Eds.) 2009, http://profs.logti.etsmtl.ca/claporte/English/VSE/index.html

(Anacleto et al. 2004) Anacleto, A., von Wangenheim, C.G., Salviano, C.F., Savi, R. (2004). *Experiences gained from applying ISO/IEC 15504 to small software companies in Brazil*, 4th International SPICE Conference on Process Assessment and Improvement, Lisbon, Portugal, pp. 33-37.

(APCQ 1997) APCQ, *If We Only Knew What We Know: Identification and Transfer of Internal Best Practices,* American Productivity & Quality Center, 123 North Post Oak Lane, Houston, Texas, USA., 1997.

(April et al. 2005) April, A., Laporte, C.Y., *An Overview of Software Quality Concepts and Management Issues*, in: Measuring Information Systems Delivery Quality, Idea Group Inc, Hershey, PA, 2005.

(April et al. 2006a) April, A., Laporte, C.Y., What is the service level of your CMMS (Computerized Maintenance Management System) Supplier, Maintenance and Asset Management, Vol. 21, No. 2, Summer 2006, pp 34-38.

(April et al. 2006b) April, A., Laporte, C.Y., Software Quality, dans Measuring Quality Requirements in Information System, Idea Group Inc, Hershey, PA, 2006, pp 28- 54.

(April et al. 2009) April, A. Laporte, C. Y., *An Overview of Software of Software Engineering Process and its improvement*, United States of America, Idea Group Publishing, Encyclopedia of information Science and Technology, 2nd edition, 2009, p 1-6, encyclopediaIST@igi-bal.com, Deborah Yahnke

(April et al. 2010) April, A., Laporte, C.Y., *Assurance qualité logicielle*– in preparation.

(Arenas 2007) Arenas, L., *Small Entrepreneurships Software Enterprises Quality Process Assessment*, ParqueSoft Colombia, Real Case, Columbia, http://parquesoft.com

(Arenas 2009) Arenas, L., Integration and Test Deployment Package, 2009. http://profs.logti.etsmtl.ca/claporte/English/VSE/index.html

(Armour 2001) P.G. Armour (2001) The Laws of Software Process, *Communications of the ACM*, Vol. 44, No. 1, pp. 15-17.

(Bagert 2004) Bagert, D., *Licensing and Certification of Software Professionals*, Advances in Computers (60) 2004.

(BAR-003-2005) Minutes of Meeting, ISO/IEC JTC1 SC7 WG24, Bari, Italy, October 24-28, 2005

(BAR-026-2005a) Survey Questionnaire, English Version, Laporte et al. (Eds.) 2005,

(BAR-026-2005b) Survey Questionnaire, French Version, Laporte et al. (Eds.) 2005.

(BAR-026-2005c) Survey Questionnaire, German Version, Laporte et al. (Eds.) 2005.

(BAR-026-2005d) Survey Questionnaire, Thai Version, Laporte et al. (Eds.) 2005.

(BAR-026-2005e) Survey Questionnaire, Turkish Version, Laporte et al. (Eds.) 2005.

(BAR-026-2005f) Survey Questionnaire, Russian Version, Laporte et al. (Eds.) 2005.

(BAR-026-2005g) Survey Questionnaire, Spanish Version, Laporte et al. (Eds.) 2005.

(BAR-026-2005h) Survey Questionnaire, Portuguese Version, Laporte et al. (Eds.) 2005.

(BAR-026-2005i) Survey Questionnaire, Korean Version, Laporte et al. (Eds.) 2005.

(Basili et al. 1994) Basili, V., Green, S. 1994. *Software Process Evolution at the SEL*, IEEE Software, July.

(Basili et al. 2008) Basili, V., Cruzes, D., Carver, J., Hochstein, L., Hollingsworth, J., Zelkowitz, M., Shull, F., Understanding the High-Performance-Computing Community, IEEE Software, Vol.25, Number 4, p29-36.

(Bate et al. 1995) Bate, R., Dorothy K., Curt W., Armitage*, J., A Systems Engineering Capability Maturity Model*, version 1.1, Software Engineering Institute, CMU/SEI-95-01, November, 1995.

(Batista et al. 2000) J. Batista and A. Dias de Figueiredo, *SPI in a Very Small Team: A Case with CMM*, Software Process: Improvement and Practice, vol. 5, no. 4, 2000, pp. 243-250.

(Bégnoche 2008) Bégnoche, L., *Amélioration des processus de développement logiciel dans les très petites organization*s**,** Masters Thesis, École de technologie supérieure, Montréal, Canada 2008

(Belkebir 2003) Belkebir, Youssef, Analysis and Improvement of the Role Definitions of the Software Engineering Process of the Competency Center of Bombardier Transportation, 2003. Co-directed with Professor Pierre Bourque of ETS.

(Benediktsson et al. 2001) Benediktsson, O., Hunter, R. B., McGettrick, A. D., *Processes for software in safety critical systems*, Software Process: Improvement and Practice, Volume 6, Issue 1, March 2001, Pages: 47-62

(Bencherif 2006) Bencherif, Khaled, Development and Deployment of a Process for the Engineering of Safety-Critical Software for an Unmanned Aerial Vehicle. Masters Thesis, École de technologie supérieure, Montréal, Canada 2006.

(Bégnoche 2008) Bégnoche, Luc, Développement de trousses de déploiement de processus pour les très petites entreprises du secteur logiciel, Masters Thesis, École de technologie supérieure, Montréal, Canada 2008.

(BER-003 2008) Minutes of Meeting, ISO/IEC JTC1 SC7 WG24, Berlin, Germany, 19-23 May 2008.

(Berrhouma 2007) Berrhouma, Nabil, *Development and Deployment of a Cost of Software Quality Measurement System at Bombardier Transportation,* Masters Thesis, École de technologie supérieure, Montréal, Canada, 2007.

(Berrhouma et al. 2009) Berrhouma, N., Laporte, C.Y., Doucet, M., April, A., *Mesure du coût de la qualité logicielle d'un projet d'envergure de la société Bombardier Transport*, Revue Génie Logiciel, Number 88, March 2009, pp 47-57.

(BK1-006 2005) Laporte, C.Y., *Action Items after the meeting-Presentation*, First Meeting of Special Working Group, Bangkok, Thailand, March 2005.

(BK1-032 2005) *Requirements Document*, First Meeting of Special Working Group, Bangkok, Thailand, March 2005.

(BK1-014 2005) *Preliminary list of VSE ISO Survey*, Second Meeting of Special Working Group, Bangkok, Thailand, September 2005.

(BK3-003 2006) Minutes of Meeting, ISO/IEC JTC1 SC7 WG24, Bangkok, Thailand, 15-19 May 2006.

(Block 1981) Block, P. 1981. Flawless Consulting, Pfeiffer & Company.

(Bloom 1956) Bloom, B., (ed.), Taxonomy of Educational Objectives: The Classification of Educational Goals, Mackay, 1956.

(Bluteau 2005) Bluteau, C., *Preliminary Questions List of VSE Survey*, École de technologie supérieure, Montréal, Canada, September 2005.

(Bluteau 2007) Bluteau, C. Étude et élaboration des besoins de très petites entreprises en développement de logiciels sur l'utilisation des normes en génie logiciel de l'organisation internationale de normalisation (ISO), Masters Thesis, École de technologie supérieure, Montréal, Canada 2007.

(Boehm 2006) Boehm, B., *A view of 20th and 21st Century Software Engineering,* Proc. ICSE'06, May 20–28, 2006, Shanghai, China

(Bourque et al. 1999) Bourque, P., Dupuis, R., Abran, A., Moore, J.W., and Tripp, L.L. *The Guide to the Software Engineering Body of Knowledge*, *IEEE Software* (16:6) 1999b, pp. 35-44.

(Bourque et al. 2002) Bourque, P., Dupuis, R., Abran, A., Moore, J.W., Tripp, L.L., and Wolff, S. *Fundamental Principles of Software Engineering – A*

Journey, Journal of Systems and Software (62:1) 2002, pp. 59-70.

(Bourque et al. 2004) Bourque, P., Belkebir, Y., Laporte, C.Y., Doucet, M., and Palza, E. *Coverage Analysis and Improvement of the Role Definitions of the Bombardier Software Engineering Process*, 17th International Conference on Software & Systems Engineering and their Applications – ICSSEA, Paris, 2004.

(Bridges 1991) Bridges, W. 1991. Managing Transitions. Addison Wesley.

(BSI 1981) British Standard Institute, *British Standards Guide: A Standard for Standards*, London, 1981.

(CAN 2004) Canadian National Committee on the International Organization for Standardization (CNC/ISO), *Responsibilities and Procedures Section D*, Canadian Advisory Committees (CACs), Standards Council of Canada, CAN-P-2014, April 2004

(CAN 2006) CNC/ISO Responsibilities and Procedures Section E-2 Canadian Convenors of ISO and JTC1 Working Groups, Standards Council of Canada, CAN-P-2016, March 2006

(CEI/IEC 1998) 61508: 1998–2000, *Functional safety of electrical/electronic/programmable electronic safety-related systems*. Parts 1–7 (IEC 1998–2000); International Electrotechnical Commission, Geneva, CEI/IEC.

(Charette 2005) Charette, R. N. 2005. *Why software fails*. Spectrum (September): 42-49.

(Churchill et al. 1983) Churchill, L., *The Five Stages of Small Business Growth,* Harvard Business Review, May-June 1983.

(Cignoni 1999) Cignoni, GA. *Rapid software process assessment to promote innovation in SME's*, Proceedings of Euromicro 99, Italy, 1999.

(Coallier 2003) Coallier, F., *International Standardization in Software and SystemsEngineering*, Crosstalk, Journal of Defense Software Engineering. pp 18-22.

(Coleman et al. 2008) Coleman, G.; O'Connor, R.; *Investigating Software Process in Practice: A Grounded Theory Perspective,* Journal of Systems and Software, Vol. 81, No. 5, pp 772-784, 2008.

(Conner 1992) Conner, D., *Managing at the Speed of Change*, Willard Books, 1992.

(Corrales et al. 2007) Corrales, Y., Laporte, C.Y., *Étude de cas: Évaluation de la migration d'une architecture logicielle d'une société de commerce électronique,* Revue Génie Logiciel, September 2007, Number 82, p 40-54.

(Côté et al. 2004a) Côté M. A., Suryn W., Martin R., Laporte, C.Y.**,** *Evolving a Corporate Software Quality Assessment Exercise: A Migration Path to ISO/IEC 9126*. Software Quality Professional, Vol. 6, No 3, ASQ, 2004.

(Côté et al. 2004b) Côté M-A., Suryn W., Martin R., Laporte C.Y., "The Analysis of the Industrial Applicability of Software Product Quality ISO Standards: The Context of MITRE's Software Quality Assessment Exercise". Actes-12th International Software Quality Management & INSPIRE Conference (BSI) 2004, Canterbury, Kent, UK 5-7 Avril 2004.

(Côté et al. 2005) Côté M. A., Suryn W., Laporte C.Y, Martin R., The Evolution Path for Industrial Software Quality Evaluation Methods Applying ISO/IEC 9126:2001 Quality Model: Example of MITRE's SQAE Method. Software Quality Journal, 13, 17–30, 2005. Springer Science + Business Media, Inc

(CRIM 1992) *Applied Software Engineering Center, Proposal to Industry*, Science and technology Canada and Ministère de l'Industrie, du Commerce et de la Technologie du Québec, February 1992.

(Crosby 1979) Crosby, Philip B. *Quality Is Free: The Art of Making Quality Certain*. New York: Mentor, New American Library, 1979.

(DACS 2003) DACS, *Software Acquisition Gold Practice* – Plan for Technology Insertion, V1.1, 2003

(Daskalantonakis 1994) Daskalantonakis, M.K., *Achieving Higher SEI Levels*, IEEE Software, July 1994.

(Davies et al. 2003) Davis, N. & Mullaney, J., *The Team Software Process (TSP) in Practice: A Summary of Recent Results* (CMU/SEI-2003-TR-014). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003.

(DoD 1994) Mil-Std-498, *Military Standard-Software Development and Documentation*, US Department of Defense, December 5, 1994.

(Deniger 2005) Deniger, Frédéric, *Development and Deployment of a Project Management Framework for a Multi Media Company*, Masters

Thesis, École de technologie supérieure, Montréal, Canada 2005.

(Deniger et al. 2006) Deniger, F., Laporte, C.Y., *Development and Deployment of a Project Management Framework in a Multi-Media and Post Production Company*, Revue Génie Logiciel, Number 77, June 2006, pp 44-53.

(Desfossés 2006) Desfossés, *Yves, Development of an ISO 9001 Certification Kit for Small Software Organizations of Québec*, Masters Thesis, École de technologie supérieure, Montréal, Canada Project co-directed with Professor Alain April, ÉTS, 2006.

(Desfossés et al. 2006) Desfosses, Y. B, Laporte, C Y, April, A., Berrhouma, Nabil, *Méthode d'amélioration des services de TI, basée sur ITIL, dans les entreprises québécoises,* Revue Génie Logiciel, Septembre 2008, number 86, pp 47-59.

(Desharnais et al 2007) Desharnais, J.-M.; Laporte, C.; Stambollian, A.; Zarour, M.; Habra, N.; Renault, A., *Initiating Software Process Improvement with a Light Model for Small Entreprise : Our Experience*, in III Taller Internacional de Calidad en Technologias de Information et de Communications, Cuba, February 15-16 , 2007. Publié sur CD.

(Dion 1993) Dion, R., *Process improvement and the corporate balance sheet.* IEEE Software, 10(4), 28–35. Figure abstracted from IEEE Software.

(Dunaway 1996) Dunaway, D., Master, S., *CMM-Based Appraisal for Internal Process Improvement (CBA-IPI): Method Description*, Software Engineering Institute, CMU/SEI-96-TR-007, 1996

(Dyba 2005) Dyba, T., An Empirical Investigation of the Key Factors for Success in Software Process Improvement, IEEE Trans. Software Eng., vol. 31, no. 5, 2005, pp. 410-424.

(Dyba et al. 2008) Dyba, T., Dingsøyr, T., *Empirical studies of agile software development: A systematic review*, Information and Software Technology 50 (2008) 833–859

(Ebert et al. 2009) Ebert, C., Salecker, J., *Embedded Software - Technologies and Trends*, IEEE Software, May/June 2009.

(EC 2002) Support Services For Micro Small And Sole Proprietor's Businesses, Results from the Study "Support Services for

Micro, Small and Sole Proprietor's Businesses" on behalf of the European Commission, DG Enterprise, April 2002, Vienna

(EC 2005) European Commission. *The new SME definition. User guide and model declaration.*, Europe: Enterprise and Industry Publications, 2005. Declaration, available at: http://europa.eu.int/comm/enterprise/enterprise_policy/sme_definition/sme_user_guide.pdf

(EC 2006) SMEs and standardisation in Europe, 23 good practices to promote the participation of craft and SME enterprises in standardisation and the use of standards, EIM Business & Policy Research, European Commission, EIM Business & Policy Research, Zoetermeer, October 2006.

(Epstein 94) Epstein, R.G., *The Case of the Killing Robot*, West Chester University, 1994, epstein@golden.wcupa.edu.

(Estrin et al. 2003) Estrin, L., Foreman, J.T., Garcia, S., *Overcoming Barriers to Technology Adoption in Small Manufacturing Enterprises*, Software Engineering Institute, CMU/SEI-2003-TR-012, 2003

(Europe 2000) *The European Observatory for SMEs*, Sixth Report, 2000

(Fagan 1976) Fagan, M.E. *Design and code inspections to reduce errors in program development.* IBM System Journal, 15(3), 182–211, 1976.

(Fenton et al. 1994) Fenton, N., Pfleeger, S., Glass, R., *Science and Substance: A Challenge to Software Engineers*, IEEE Software, Vol. 11, No. 4 July 1994, pp. 86-95.

(Feng 2003) Feng, P., Studying Standardization : A Review of the Literature, in Proc. 3rd IEEE Conference on Standardization and Innovation in Information Technology, Delft, The Netherlands, 2003

(Fogle et al. 2001) Fogle, S., Loulis, C., Neuendorf, B., *The Benchmarking Process: One Team's Experience*, IEEE Software Sep/Oct 2001.

(Ford et al. 1996) Ford, G., and Gibbs, N.E. *A Mature Profession of Software Engineering*, CMU/SEI-96-TR-004, Software Engineering Institute, Pittsburgh, USA, January, 1996.

(Foisy 2007) Foisy, Marco, Development and Deployment of a Software Test Process for the Wind Mill Research Laboratory of École de Technologie Supérieure, Masters Thesis, École de technologie supérieure, Montréal, Canada 2007.

(Ferris 2006) Ferris, T., *Cross-Cultural Issues Associated with the Application of ISO/IEC 15288 Standard*, INCOSE 2006 - 16th Annual International Symposium Proceedings.

(Fuller et al. 2006) Fuller, G. Keith, Vertinsky, I., *Market response to ISO 9000 certification of software engineering processes.*(International Organization for Standardization). International Journal of IT Standards and Standardization Research 4.2 (July-Dec 2006): 43(12).

(Garcia 2005) Garcia, S., *Thoughts on Applying CMMI in Small Settings*, Presentation to the Montréal SPIN, 2005.

(Garcia et al. 2006) Garcia, S., Graettinger, C., Kost, K., eds., *Proc. 1st Int'l Research Workshop for Process Improvement in Small Settings*, special report CMU/SEI-2006-SR-001, Software Eng. Institute, 2006; www.sei.cmu.edu/pub/documents/06.reports/ pdf/06sr001.pdf.

(Garcia et al. 2007) Garcia, S., Turner, R., *CMMI Survival Guide,* Addison Wesley Professional, 2007.

(Garcia et al. 2008) Garcia, S., Graettinger, C., Camody, C., *Prototype for a Field Guide for Improving Processes in Small Settings*, Software Engineering Institute, March 2008.

(Gauthier 2004) Gauthier, R., *Une force en mouvement*, La Boule de Cristal, Centre de recherche informatique de Montréal, January 22th 2004.

(Gilb et al. 1993) Gilb, T., & Graham, D. (1993). *Software inspection.* Addison-Wesley, Wokingham, U.K., ISBN: 0-201-63181-4

(Ginsberg et al. 1995) Ginsberg, M.; Quinn, L.; *Process Tailoring and the Software Capability Maturity Model*, Software Engineering Institute, CMU/SEI-94-TR-024, November 1995.

(Glass 1997) Glass, R., L., Pilot Studies: What, Why, and How, J. Systems and Software, 1997; 36: 85-97

(Gotterbarn 1999) Gotterbarn, F. *How the new software engineering code of ethics affects you.* IEEE Software, 16(6), 58–64, 1999.

(Gotterbarn et al. 1999) Gotterbarn, D., Miller, K., & Rogerson, S. *Computer society and ACM approve software engineering code of ethics*. IEEE Computer, 32(10), 84–88, 1999.

(Grady 1997) Grady, R.: Successful Software Process Improvement. Prentice Hall.

(Green et al. 1998) Green, G., Hevner, A., Perceived Control of Software Developers and Its Impact on the Successful Diffusion of Information Technology, Special Report CMU/SEI-98-SR-013, Apr. 1999;

(Green et al. 2000) Green, Gina C., and Alan R. Hevner. *The Successful Diffusion of Innovations: Guidance for Software Development Organizations*, IEEE Software 17.6 (Nov-Dec 2000): 96(8).

(Green et al. 2005) Green, Gina, Alan Hevner, and Rosann Collins. *The Impacts of Quality and Productivity Perceptions on the Use of SPIs*, Information and Software Technology Journal, 47(8), pp 543-553, June 2005.

(Green et al. 2006) Gina, G., Hevner, A., Collins. R., *Developer-Driven Quality: Guidelines for Implementing Software Process Improvements*, Measuring Quality Requirements in Information Systems, E. Duggan and H. Reichgelt, Eds., Idea Group, Inc., 2005.

(Guillemot 2009) Guillemot, F., Développement et implantation d'un processus de génie logiciel pour un groupe de recherche en neuroscience et d'une trousse de déploiement pour la conception logicielle inspirée de la future norme ISO pour les très petites structures. Projet de maîtrise en génie logiciel, École de technologie supérieure. En cours de rédaction.

(Guillemot 2009) Guillemot, F., Architecture and Detailed Design Deployment Package, Technical report, Laporte et al. (Eds.), 2009, http://profs.logti.etsmtl.ca/claporte/English/VSE/index.html

(Habra et al. 1999) Habra, N., Niyitugabira E., Lamblin A.C., Renault A., *Software Process Improvement for Small Structures: First Results of a Micro-Evaluation Framework*, in Proceedings of the European Conference on Software Process Improvement SPI'99, Barcelona, Spain, December 1999.

(Habra et al. 2008) Habra, N., Alexandre, S., Desharnais, J-M, Laporte, C.Y., Renault, A., *Initiating Software Process Improvement in Very Small Enterprises Experience with a Light Assessment Tool*, Information and Software Technology, Volume 50, Issues 7-8, juin 2008, Pages 763-771.

(Hadden 1998) Hadden, R., *Key Practices to the CMM: Inappropriate for Small Projects*, Panel, Proceedings of the Software Engineering Process Group Conference, Chicago, 1998.

(Haley 1996) Haley, Software process improvement at Raytheon. *IEEE Software, 13*(6), 33–41. *Figure abstracted from IEEE Software,* 1996

(Hareton et al. 2006) *CMM Fast Track (CMMFT) Programme for Hong Kong SME Software Companies,* In: Proceedings of the First International Research Workshop for Process Improvement in Small Settings, pp. 208–215. Carnegie Mellon University, Pittsburgh (2006)

(Harrison 2004) Harrison, W., *Best Practices: Who Says?*, IEEE Software, Jan/Feb 2004

(Hofer 2002) Hofer, C.; Software Development in Austria: Results of an Empirical Study among Small and Very Small Enterprises, Proceedings of the 28th Euromicro Conference, 2002: 361-366

(Hofstede 1997) Hofstede, G., *Culture and Organizations – Software of the Mind.* McGraw-Hill, 1997

(Hofstede et al. 2002) Hofstede, G.J., Pedersen, P.B., Hofstede, G., *Exploring Culture -Exercises, Stories and Synthetic Cultures*, Intercultural Press, 2002

(Holland 1998) Holland, D., *Document inspection as an agent of change.* In A. Jarvis & L. Hayes (Eds.), *Dare to be excellent.* Upper Saddle River, NJ: Prentice Hall, 1998.

(Humphrey 1997) Humphrey, W.S., *Introduction to the Personal Software Process*, Addison-Wesley, 1997.

(Humphrey 2000) Humphrey, W.S., *Introduction to the Team Software Process*, Addison Wesley, 2000.

(Humphrey 2002) Humphrey, W.S., *Winning with Software - An Executive Strategy*, Addison Wesley, 2002.

(Humphrey 2008) Humphrey, W.S., *The Process Revolution*, Crosstalk-The Journal of Defense Software Engineering, August 2008, pp 4-6.

(HYD-003 2009) Minutes of Meeting, ISO/IEC JTC1 SC7 WG24, Hyderabad, India, 25-29 May 2009.

(Iberle 2002) Iberle, K. *But Will it Work for Me?*, Proceedings of the Pacific Northwest Software Quality Conference, Portland, USA, 2002

(ICES 2009) ICES, International Cooperation on Education about Standardization, 2009, http://www.standards-education.org/

(IEEE 1990) IEEE, Standard Glossary of Software Engineering Terminology, 610.12, IEEE, 1990.

(IEEE 1986) IEEE, IEEE Guide for Software Quality Assurance Planning, 983, IEEE, 1986.

(IEEE 1999) IEEE, *Software Engineering Code of Ethics and Professional Practice*, IEEE-CS/ACM, 1999, Http://www.computer.org/certification/ethics.htm

(IEEE 2002) IEEE Std 730-2002. IEEE Standard for Software Quality Assurance Plans

(IEEE 2003) IEEE, *Software Engineering Standards Collection* CD-ROM, 2003, http://www.computer.org/cspress/.

(IEEE 2008a) IEEE Std 1028-2008, *IEEE Standard for Software Reviews*: IEEE.

(IEEE 2009) Certified Software Development Associate (CSDA), IEEE Computer Society, 2009. http://www2.computer.org/portal/web/csda/about

(Iivari 1996) Iivari, J. *Why Are CASE Tools Not Used?* Communications of the ACM (39:10), October 1996, pp. 94-103.

(INCOSE 2007) *Systems Engineering Handbook - A Guide for System Life Cycle Processes and Activities*, INCOSE (International Council on Systems Engineering), Version 3.1, August 2007.

(INCOSE 2009) *Meeting Minutes « Se For VSMEs » Project*, April 16, 2009, Kick Off Meeting.

(ISO 1995) ISO/IEC 12207 (1995), Information technology – Software life cycle processes, International Organization for Standardization/ International Electrotechnical Commission: Geneva, Switzerland.

(ISO 1998) ISO/IEC TR 10000-1 (1998), Information technology – Framework and taxonomy of International Standardized Profiles – Part 1: General principles and documentation framework, Fourth Edition, International Organization for Standardization/ International Electrotechnical Commission: Geneva, Switzerland.

(ISO 2003) ISO/IEC 15504-2 (2003) Information technology -- Process assessment -- Part 2: Performing an assessment, International Organization for Standardization/ International Electrotechnical Commission: Geneva, Switzerland.

(ISO 2003-2005) ISO/IEC 15504 (2003-2005). ISO/IEC IS 15504: Information technology – Process assessment, Part 1 to Part 5, International Organization for Standardization/ International Electrotechnical Commission: Geneva, Switzerland.

(ISO 2004a) ISO/IEC, Information Technology – Process Technology, 15504 – Parts 1 to 5, International Organization for Standardization/International Electrotechnical Commission: Geneva Switzerland.

(ISO 2004b) ISO/IEC 90003, International Standards Organization. Software Engineering: *Guidelines for the application of ISO9001:2000 to computer software*, ISO/IEC Standard 90003:2004. (2004). International Organization for Standardization/International Electrotechnical Commission: Geneva Switzerland.

(ISO 2004c) ISO/IEC JTC1 SC7, Ad Hoc Report – Software Engineering Standards for Small and Medium Enterprise http://142.137.17.56/Labo_Recherche/Lrgl/sc7/N3051-N3100/07N3060%20%20004_07BRIxx-%

(ISO 2005a) ISO/IEC TR 19759:2005, Software Engineering Body of Knowledge, Technical Report, International Organization for Standardization/ International Electrotechnical Commission: Geneva, Switzerland. Available free of charge at: http://standards.iso.org/ittf/PubliclyAvailableStandards/c033897_ISO_IEC_TR_19759_2005(E).zip

(ISO 2005b) ISO/IEC JTC1/SC7 N3288, *New Work Item Proposal – Software Life Cycles for Very Small Enterprises,* May 2005. http://www.jtc1-sc7.org/.

(ISO 2005c) Resolutions – JTC1/SC7 Plenary Meeting, ISO/IEC JTC1/SC7 N3274, Helsinki, Finland, May 2005.

(ISO 2005d) ISO/IEC 17799 Information technology - Security techniques - Code of practice for information security management, International Organization for Standardization/ International Electrotechnical Commission: Geneva, Switzerland.

(ISO 2005e) ISO/IEC 20000- Information Technology — Service Management, International Organization for Standardization/International Electrotechnical Commission: Geneva Switzerland.

(ISO 2005f) The Global Use of Environmental Management System by Small and Medium Enterprises Executive Report, ISO/TC207/SC1/ Strategic SME Group, May 2005

(ISO 2006a) ISO/IEC 15289 Systems and software engineering - Content of systems and software life cycle process information products (Documentation)

(ISO 2006b) Standards, big benefits for small business, ISO Focus, International Organization for Standardisation (ISO), Vol. 3, Sept 2006.

(ISO 2007a) ISO/IEC JTC 1 Directives, 5th Edition, Version 3.0, International Organization for Standardization/ International Electrotechnical Commission: Geneva, Switzerland.

(ISO 2007b) ISO/IEC TR 24774:2007, Software and systems engineering — Life cycle management- Guidelines for process description, International Organization for Standardization/ International Electrotechnical Commission: Geneva, Switzerland.

(ISO 2008a) ISO/IEC 24773: 2008, Software engineering — Certification of software engineering professionals — Comparison framework, International Organization for Standardization/ International Electrotechnical Commission: Geneva, Switzerland.

(ISO 2008b) ISO/IEC 9001:2008, Quality management systems — Requirements, International Organization for Standardization/ International Electrotechnical Commission: Geneva, Switzerland.

(ISO 2008c) ISO/IEC 15288:2008, Systems engineering - Systems life cycle processes, International Organization for Standardization/ International Electrotechnical Commission: Geneva, Switzerland.

(ISO 2008d) ISO/IEC 12207:2008, Information technology – Software life cycle processes, International Organization for Standardization/ International Electrotechnical Commission: Geneva, Switzerland.

(ISO 2009a) ISO/IEC 29110 - Lifecycle Profiles for Very Small Entities (VSEs) – Part 1: Overview. International Organization for Standardization/International Electrotechnical Commission: Geneva, Switzerland,

(ISO 2009b) ISO/IEC 29110 -Lifecycle Profiles for Very Small Entities (VSEs) – Part 2: Framework and Taxonomy. International Organization for Standardization/International Electrotechnical Commission: Geneva, Switzerland.

(ISO 2009c) ISO/IEC 29110 - Lifecycle Profiles for Very Small Entities (VSEs) – Part 3: Assessment Guide. International Organization for Standardization/International Electrotechnical Commission: Geneva, Switzerland.

(ISO 2009d) ISO/IEC 29110 - Lifecycle Profiles for Very Small Entities (VSEs) – Part 4-1: Specification – Basic VSE Profile. International Organization for Standardization/International Electrotechnical Commission: Geneva, Switzerland.

(ISO 2009e) ISO/IEC 29110 - Lifecycle Profiles for Very Small Entities (VSEs) – Part 5-1: Management and Engineering Guide – Basic VSE Profile. International Organization for Standardization/International Electrotechnical Commission: Geneva, Switzerland.

(ISO 2009f) ISO/IEC 24765: 2009, Systems and Software Engineering Vocabulary, International Organization for

Standardization/International Electrotechnical Commission: Geneva, Switzerland.

(ISO 2009g) ISO/IEC JTC 1 SC7 Information, Hyderabad, India, May 2009.

(ISO 2009h) SC7 Secretariat Report to the SC7 AG Meeting, Hyderabad, India, May 24, 2009.

(ISO 2009i) ISO/IEC JTC1 Information, Hyderabad, India, May 24, 2009.

(Johnson et al. 1998) Johnson, D.; Brodman, J.; *Applying the CMM to Small Organizations and Small Projects*, Proceedings of Software Engineering Process Group Conference, Chicago, 1998

(Jorgenson et al. 2006) Jorgenson, D.W., and C.W. Wessner, Eds. *Measuring and Sustaining the New Economy, Software, Growth, and the Future of the U.S. Economy*. Washington, D.C.: National Academies Press, 2006.

(Kabli 2009) Kabli, Samia, Conception, réalisation et mise à l'essai de trousses de déploiement pour faciliter et accélérer l'implémentation de la norme ISO/CEI 20000 par les très petites structures, École de technologie supérieure, 2009.

(Katzenbach 1994) Katzenbach, J. R., Smith, D.K., *The Wisdom of Teams,* Harper Business, 1994.

(Kasunic 2005) Kasunic, M. *Designing an Effective Survey*, Handbook CMU/SEI-2005-HB-004, Software Engineering Institute, 2005.

(Kautz et al. 2000) Kautz, K., Hansen, H.W., Thaysen, K.: Applying and adjusting a software process improvement model in practice: the use of the IDEAL model in a small software enterprise. In: ICSE 2000, Limerick, Ireland, pp. 626–633 (2000)

(Korneck 2008) Korneck, A.J., *Computing Curricula for the 21st Century*, IEEE Distributed Systems Online (vol. 9, no 2, 2008), art. no 0802-mds2008020002

(Krasner 1998) Krasner, H., *Using the cost of quality approach for software*, Crosstalk - The Journal of Defense Software Engineering, volume 11, number 11, November 1998.

(Kroll et al. 2003) Kroll, P.; Kruchten, P.; *The Rational Unified Process Made Easy – A Practionner's Guide to the RUP.*; Addison-Wesley, 2003, ISBN0-321-16609-4

(Kruchten 2003) Kruchten, P. *The Rational Unified Process: An Introduction*, (3rd ed.), Addison-Wesley, 2003.

(Kruchten 2004) Kruchten, P. *Putting the "Engineering" into Software Engineering*, Australian Software Engineering Conference, 2004, pp. 2-8.

(Kruchten 2005) Kruchten, P. Casting the Software Design in the Function-Behavior-Structure Framework, IEEE Software (22:2) 2005, pp. 52-58.

(Kurniawati et al. 2005) Kurniawati, F., Jeffery, R., *The Use and Effects of an Electronic Process Guide and Experience Repository: A Longitudinal Study*, J. Information and Software Technology, 2005, pp. 1-12.

(Kulpa et al. 2008) Kulpa, M., Johnson, K.A., *Interpreting the CMMI: A Process Improvement Approach*, CRC Press.

(Land 1997) Land, S. K. Results of the IEEE Survey of Software Engineering Standards Users. *Software Engineering Standards Symposium and Forum, 1997. Emerging International Standards. ISESS 97,* Walnut Creek, CA, June 1-6, pp. 242 – 270, 1997.

(Lambert 1992) Lambert, G., 1992, *"Bénéfices de l'évaluation du Software Engineering Institute"*, Minutes of Symposium 92 GIRICO, Montreal, May.

(Laporte 1978a) "Un nouveau concept de carburation", Collège militaire royal de Saint Jean, Qué., September 13, 1978.

(Laporte 1978b) "A New Concept of Carburation", Defence Research Establishment, Ottawa, Ont., July 13, 1978.

(Laporte 1978c) "A New Concept of Carburation", Conférence du Génie du Matériel Terrestre, Borden, Ontario, May 1978.

(Laporte 1980) Laporte, C.Y., *Réalisation d'un système d'acquisition de données pour banc d'essaie de moteurs à combustion interne*, Mémoire de Maîtrise ès sciences (physique), Université de Montréal, Québec, 1980.

(Laporte 1981a) "Voice Interactive Systems and their Applications to Avionics", Ottawa Chapter of the Old Crows Association, Ottawa, Ont., Dec. 1981.

(Laporte 1981b) "Voice Interactive Systems and their Applications to Avionics", Director Aerospace Combat System, Ottawa, Ontario, Dec. 1981.

(Laporte 1981c) "Voice Interactive Systems and their Applications to Avionics", Aerospace Engineering Conference, Greenwood, Nova Scotia, Oct 1981.

(Laporte 1981d) "Man/Machine Interface and Voice Input/Output", Montréal Chapter "Instrument Society of America", Montréal, 24 September 1981.

(Laporte 1981e) "Voice Synthesis and its Applications", Director Avionics and Simulator Engineering, Ottawa, Ont., May 1981.

(Laporte 1982) "Voice Input/output and their Applications", Defence and Civil Institute of Environmental Medicine, Canadian Forces Base Toronto, Toronto, Ontario, August 6, 1982.

(Laporte 1985) Laporte, C.Y., "A Description of the Architecture of a Voice I/O Evaluation System and its Application in a Military Jet Aircraft", Speech Tech 1985, New York, 22 24 avril 1985. Actes - Speech Tech '85, Vol. 1, Number 2, 1985, pp 313 316.

(Laporte et al. 1985) Laporte, C.Y., Houle, J.L., "La réalisation d'un terminal graphique pour le contrôle du traffic aérien d'aéroports militaires canadiens". Actes-Third Int. Symposium of Applied Informatics", Grindelwald, Suisse, 19 22 février 1985.

(Laporte 1986a) Laporte, C.Y., *Réalisation d'un système à interaction vocale pour un avion militaire à hautes performances*. Mémoire de Maîtrise es Sciences Appliquées, Informatique appliquée et Ordinateur, École Polytechnique, Montréal, 1986.

(Laporte 1986b) "La synthèse et la reconnaissance de la parole et ses applica¬tions dans un avion", Experimental Aircraft Association, St Hubert, 8 oct. 1986.

(Laporte et al. 1986a) Laporte, C.Y., Houle, J. L., *La réalisation d'un terminal graphique pour le contrôle du traffic aérien d'aéroports militaires canadiens*, IASTED Journal Control and Computers, Vol. 14, Number 2, 1986, pp 38- 43.

(Laporte et al. 1986b) Laporte, C.Y., Houle, J.L., The Architecture of a Voice Interactive Evaluation System and its Application to Radio Frequency Selection in a Military Aircraft. Proceedings-

International Society for Mini and Microcomputers (ISMM), Mimi 1986, IEEE, Beverley Hills, California, Feb. 1986.

(Laporte et al. 1986c) Laporte, C.Y., Houle, J.L., "Design of a Voice Interactive System for Command Functions in a Military Aircraft", Actes - IASTED Fourth Int. Symposium of Applied Informatics, Innsbruck, Autriche, Fév. 1986.

(Laporte et al. 1988) Laporte, C.Y., Houle, J. L., *A Distributed Computer System for the Development of Voice Interactive Applications*, Int. Journal of Mini and Microcomputers, Vol. 10, No. 2, 1988.

(Laporte et al. 1989) Laporte, C.Y., Houle, J. L., *A Voice Interactive Evaluation System for Command Functions in Military Aircraft*, Microcomputer Applications, Vol. 9, No. 1, 1989.

(Laporte 1991) "A Canadian Software Engineering Initiative", Saskatchewan Ada Association, Régina, Saskatoon, Mai 23-24, 1991.

(Laporte 1992) "The Applied Software Engineering Centre", Aerospace Industries Association of Canada, Ottawa, 14 avril 1992.

(Laporte et al. 1993) Laporte, C.Y., Martel, Y., Ste-Marie,J., Tremblay, J., Khalil, D., Denoncourt, M., Demers, P., Hawthorne, L., Jennings, M., Basque, R., Poulin, L., *Oerlikon Aerospace-Report of Findings and Recommendations,* Oerlikon Aerospace, Saint-Jean-sur-Richelieu, 1993.

(Laporte 1994) Laporte, C.Y., ¨Process Improvement and the Management of Change¨, Actes - IEEE Computer Society Workshop on Software Engineering Technology Transfer, Avril 28-29, 1994.

(Laporte 1995a) ¨La démarche et l'expérience de Oerlikon Aérospatiale¨, Journée de formation, Association professionnelle des informaticiens et informaticiennes du Québec, 22 novembre 1995.

(Laporte 1995b) Laporte, C.Y., *Software Process Engineering Activities in Quebec*, dans : Metrics in Evolution, A German-Quebec Workshop, GMD - Forschungszentrum Informationstechnik GmgH, Report No. 254, Monika Mullerburg, Alain Abran, editors. ISBN 3-486-23589-3, Published by: R. Oldenbourg Verlag GmbH, Munich, 1995.

(Laporte 1996a) Laporte, C.Y., *Software Process Engineering Activities in Québec*, Proceedings - Software Process 96 Conference, Brighton U.K., December 1996.

(Laporte 1996b) Laporte, C.Y., Maturation du génie logiciel au Québec : où en sommes-nous ?, L'Expertise informatique, Vol. 2, No 1, été 1996, pp 2 - 9.

(Laporte et al. 1996) Laporte, C.Y., Papiccio, N., *Software and Systems Engineering Process Improvement at Oerlikon Aerospace,* Proceedings -Vision96 Process Improvement Symposium, Montréal, Oct. 8-10, 1996.

(Laporte 1997) ¨Software and Systems Engineering Process Development and Integration at Oerlikon Aerospace¨, Université du Québec à Montréal, 26 septembre 1997.

(Laporte et al. 1997a) Laporte, C.Y., Papiccio, N., *Development and Integration of Engineering Processes at Oerlikon Aerospace.* Proceedings - International Council on Systems Engineering- INCOSE, Los Angeles, August 3-7, 1997.

(Laporte et al. 1997b) Laporte, C.Y., Papiccio, N., L'ingénierie et l'intégration des processus de génie logiciel, de génie systèmes et de gestion de projets, Revue Génie Logiciel, Vol. 46, 1997.

(Laporte 1998) Laporte, C.Y. Development and Integration Issues about Software Engineering, Systems Engineering and Project Management Processes, Proceedings - Software Process Conference, Monte Carlo, December 1-4, 1998.

(Laporte et al. 1998a) Laporte, C.Y., Guay, A., Tousignant, J., *The Application of a Systems Engineering Process to the Re-Engineering of an Air Defense System*, Proceedings-8th Annual International Symposium of the International Council on Systems Engineering, Vancouver, July 26-30, 1998.

(Laporte et al. 1998b) Laporte, C.Y., Trudel, S., Addressing the People Issues of Process Improvement Activities at Oerlikon Aerospace, Software Process-Improvement and Practice, Vol. 4, 1998.

(Laporte et al. 1998c) Laporte, C.Y., Papiccio, N., Trudel, S., *A Software Factory for the Canadian Government Year 2000 Conversion Program*, Proceedings-Software Process Conference, Monte Carlo, December 1-4, 1998.

(Laporte et al. 1998d) Laporte, C.Y., Papiccio, N., *Software and Systems Engineering Process Development and Integration at Oerlikon Aerospace*, Software Process Newsletter, IEEE Computer Society, Committee on Software Process, Technical Council on Software Engineering, No.11, Winter 1998.

(Laporte 1999a) "La gestion et la création de la connaissance", Colloque de l'Association professionnelle des informaticiens et informaticiennes du Québec, Montréal 17 juin 1999.

(Laporte 1999b) "Development and Integration Issues about Software Engineering, Systems Engineering and Project Management Processes", Université du Québec à Montréal, Séminaire du département d'informatique, 29 janvier 1999.

(Laporte et al. 1999a) Laporte, C.Y., Trudel, S., *Development Integration and Implementation of Engineering Processes at Oerlikon Aerospace*, International Council on Systems Engineering–Insight, Vol. 2 issue 3, Fall 1999.

(Laporte et al. 1999b) Laporte, C.Y., Boucher, G., *Implementing Risk Management as part of a Systems Engineering Process*. Proceedings - 9th Annual International Symposium of the International Council on Systems Engineering, Brighton, U.K., 1999.

(Laporte et al. 1999c) Laporte, C.Y., Trudel, S., *Addressing the People Issues when Developing and Implementing Engineering Processes.* Proceedings - 9th Annual International Symposium of the International Council on Systems Engineering, Brighton, U.K., 1999.

(Laporte et al. 1999d) Laporte, C.Y., Trudel, S., Addressing People Issues when Developing and Implementing Engineering Processes, Crosstalk - Journal of Defense Software Engineering, November 1999.

(Laporte 2001) 'Le triangle de la qualité des logiciels: le personnel, le processus et le produit', Colloque, Ministère de la Santé et des Services sociaux du Québec, 13 juin 2001.

(Laporte et al. 2003) Laporte, C.Y., Boucher, G., Risk Management Applied to the Re-Engineering of a Weapon System, Crosstalk - Journal of Defense Software Engineering, January 2003.

(Laporte 2004a) Laporte, C.Y., *INCOSE 3rd Annual Conference for Space*, Insight, Publication of the International Council on Systems Engineering (INCOSE), Vol. 7, Issue 3, October 2004.

(Laporte 2004b) Coverage Analysis and Improvement of the Role Definitions of the Bombardier Software Engineering Process. International Conference Software & Systems Engineering and their Applications (ICSSEA), Paris, December 2004.

(Laporte 2005) *Adaptation of ISO/IEC Software Engineering Standards for Very Small Enterprises*, Software Engineering Process Group Conference - Latin America, November 11th 2005, Guadalajara (Mexico).

(Laporte et al. 2005a) Laporte, C.Y., April, A., Renault, A., *L'application des normes de génie logiciel dans les très petites entreprises: Historique et premiers résultats*, Revue Génie Logiciel, Number 75, December 2005, pp 7-12.

(Laporte et al. 2005b) Laporte, C.Y., Renault, A., Desharnais, J. M., Habra, N., Abou El Fattah, M., Bamba, J. C., *Initiating Software Process Improvement in Small Enterprises: Experiment with Micro-Evaluation Framework*, *SWDC-REK,* International Conference on Software Development, University of Iceland, Reykjavik, Iceland May 27 - June 1, 2005, pp 153-163.

(Laporte et al 2005c) Laporte, C.Y., Bluteau, C., *Communications Plan to Survey Very Small Enterprises*, École de technologie supérieure, Montréal, Canada, September 2005.

(Laporte et al 2005d) Laporte, C.Y., Bourque, P., Belkebir, Y., Doucet, M., "Amélioration de la définition des rôles du processus de génie logiciel de la société Bombardier Transport", Revue Génie Logiciel, Numéro 72, mars 2005, pp 43-52.

(Laporte 2006a) The Application of International Software Engineering Standards in Very Small Enterprises, Software Engineering Workshop, July 5-7th 2006, Cartagena de Indias (Columbia).

(Laporte 2006b) The Application of International Software Engineering Standards in Very Small Enterprises, Parquesoft, July 11th 2006, Cali (Columbia).

(Laporte et al. 2006a) Laporte, C.Y., Desharnais, J.M., Abou El Fatah, M., Bamba, J.C., Renault, A., Alexandre, S., Habra, N., *L'amélioration des* *processus logiciels pour de petites entreprises : nos expériences avec les micro-évaluations*, Revue Génie Logiciel, Number 79, December 2006, pp 2- 9.

(Laporte et al. 2006b) Laporte, C.Y., April, A, Renault, A., *Applying ISO/IEC Software Engineering Standards in Small Settings: Historical Perspectives and Initial Achievements*, SPICE Conference, May 4-5, 2006, Luxembourg, pp 57-62.

(Laporte et al. 2006c) Laporte, C.Y., April, A., *Applying Software Engineering Standards in Small Settings: Recent Historical Perspectives and Initial Achievements*, in: Proceedings of the First International Research Workshop for Process Improvement in Small Settings, 2005. Software Engineering Institute, Carnegie Mellon University, CMU/SEI-2006-Special Report-001, January 2006, p 39-51.

(Laporte et al. 2006d) Laporte, C.Y., April, A, Renault, A, *The application of International Software Engineering Standards in Very Small Enterprises*, Revue canadienne de l'IEEE, Number 53, Fall 2006, Issue 53, pp 14-15.

(Laporte et al. 2006e) Laporte, C.Y., Renault, A., Alexandre, S., Uthayanaka, T., *The Application of ISO/IEC JTC 1/SC7 Software Engineering Standards in Very Small Enterprises,* ISO Focus, International Organization for Standardisation (ISO), Vol 3, Septembre 2006, pp 36-38.

(Laporte 2007a) Laporte, C.Y., *Standards Application in Very Small Enterprises*, IEEE Computer, October 2007, Volume 40, number 10, pp 22-23

(Laporte 2007b) *Le développement de normes internationales de génie logiciel pour les très petites entreprises*, Journée qualité informatique, Centre de recherche informatique de Montréal (CRIM), Québec, Decembre 13th 2007.

(Laporte 2007c) The Application of International Software Engineering Standards in Very Small Enterprises, Ottawa Software Quality Association, Ottawa SPIN, June 14th 2007.

(Laporte 2007d) *The Application of International Software Engineering Standards in Very Small Enterprises*, Software Engineering Process Group

Conference, Software Engineering Institute, March 26-29 2007, Austin, Texas.

(Laporte 2007e) Improvement of Software Engineering Performances An Experience Report at Bombardier Transportation – Total Transit Systems Signaling Group. Actual problems of systems and software engineering Seminar, International Center for Informatics and Electronics (ICIE), Moscow, May 18 2007.

(Laporte et al. 2007a) Laporte, C.Y., April, A., Renault, A., *Applying ISO/IEC JTC 1/SC7 Software Engineering Standards in Very Small Enterprises*, Crosstalk, Journal of Defense Software Engineering, February 2007, Vol. 20, No 2, pp 29-30.

(Laporte et al. 2007b) Laporte, C.Y., Doucet, M., Bourque, P., Belkébir, Y., *Utilization of a Set of Software Engineering Roles for a Multinational Organization,* Product Focused Software Development and Process Improvement, PROFES 2007, Riga (Latvia), July 2-4 2007.

(Laporte et al. 2007c) Laporte, C.Y., Doucet, M., Roy, D., Drolet, M., Improvement of Software Engineering Performances An Experience Report at Bombardier Transportation – Total Transit Systems Signalling Group, INCOSE (International Council on Systems Engineering) Seventeenth International Symposium, San Diego (U.S.A.), June 24-28 2007.

(Laporte et al. 2007d) Laporte, C.Y., April, A., Benchérif, K., *Teaching Software Quality Assurance in an Undergraduate Software Engineering Program,* Industrial automation 2007, ÉTS, Montréal, June 11-13 2007.

(Laporte et al. 2007e) Laporte, C.Y., April, A., Benchérif, K., Teaching Software Quality Assurance in an Undergraduate Software Engineering Program, Software Quality Professional, ASQ, Vol. 9, Issue 3, 2007, p 4-10.

(Laporte 2008a) *Le développement de normes internationales de génie logiciel pour les très petites entreprises*, Journées de l'Aber Wrac'h, Séminaire annuel du Laboratoire d'Informatique des systèmes Complexes, Aber Wrac'h, Bretagne, France, May 29th 2008.

(Laporte 2008b) The Development of International Standards for Very Small Entities: Historical Perspectives and Achievements, Mexico City, Mexico, November 12th 2008.

(Laporte et al. 2008a) Laporte, C.Y., Alexandre, S., Renault, A., *The Application of International Software Engineering Standards in Very Small Enterprises*, Software Quality Professional, ASQ, Vol. 10, Number 3, June 2008, p 4-11.

(Laporte et al. 2008b) Laporte, C.Y., Alexandre, S, Renault, A., *Developing International Standards for Very Small Enterprises*, IEEE Computer, Mars 2008, p 82-85.

(Laporte et al. 2008c) Laporte, C.Y., Renault, A., Alexandre, S., *Applying ISO/IEC Software Engineering Standards in Very Small Enterprises,* in Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies, Idea Group Inc, Hershey, PA, 2008, pp 42-70.

(Laporte et al. 2008d) Laporte, C.Y., Alexandre, S., O'Connor, R., *A Software Engineering Lifecycle Standard for Very Small Enterprises,* in R.V. O'Connor et al. (Eds.): EuroSPI 2008, CCIS 16, pp. 129–141.

(Laporte et al. 2008e) Laporte, C.Y., Alexandre, S., Renault, A., Crowder, K.V., *The Development of International Standards for Very Small Enterprises*, INCOSE (International Council on Systems Engineering) Seventeenth International Symposium, Amsterdam, June 15-19 2008.

(Laporte et al. 2008f) Laporte, C.Y., Roy, R., Novieli, R., La gestion des risques d'un projet de développement et d'implantation d'un système informatisé au Ministère de la Justice du Québec, Revue Génie Logiciel, mars 2008, numéro 84, p 2-12.

(Laporte et al. 2009g) Laporte, C.Y., Vasquez, A., Deployment Package Template, Technical report, 2009. http://profs.logti.etsmtl.ca/claporte/English/VSE/index.html

(Laporte 2009a) The Development of International Standards for Very Small Entities: Historical Perspectives, Achievements and Way Forward, NASSCOM Quality Summit, Hyderabad, India, May 2009.

(Laporte 2009b) Plan d'affaires, Centre de support en génie logiciel aux Très petites organisations, École de technologie supérieure, June 19th 2009.

(Laporte 2009c) Laporte, C.Y., Select and Conduct Pilot Projects, Deployment Package, Technical report, 2009. http://profs.logti.etsmtl.ca/claporte/English/VSE/index.html

(Laporte 2009d) Laporte, C.Y., Product Delivery Deployment Package, Technical report, Laporte et al. (Eds.), 2009. http://profs.logti.etsmtl.ca/claporte/English/VSE/index.html

(Leveson et al. 1993) Leveson, N, Turner, C, *An Investigation of the Therac-25 Accidents*, IEEE Computer*,* 1993; Vol. 26, Issue 7; p18-41.

(Lindvall et al. 2003) Lindvall, M., Komi-Sirviö, S., Costa, P., Seaman, C., *A State of the Art Report: Embedded Software Maintenance*, Data and Analysis Center for Software, Rome, New York, 2003.

(Long 2008) Long, L., *The Critical Need for Software Engineering Education*, Crosstalk - The Journal of Defense Software Engineering, January 2008, pp 6-10.

(Leung et al. 2001) H.K.N. Leung and T.C.F. Yuen, *A Process Framework for Small Projects*, Software Process: Improvement and Practice, vol. 6, no. 2, 2001, pp. 67-83.

(LUX-003 2008) Minutes of Meeting, ISO/IEC JTC1 SC7 WG24, Luxembourg, 2-6 October 2006.

(Magee et al. 2004) Magee, S., Thiele, D., *Engineering Process Standards: State of the Art and Challenges*, IEEE, IT Pro, September/October 2004, p 38- 44.

(Mäkäräinen 2000) Mäkäräinen, M., Software Change Management Process in the Development of Embedded Software, VTT Publications, pp. –185, 2000

(McCaffery et al. 2005) McCaffery FH, McFall DJ, Donnelly P, Wilkie FG, Sterritt R, *A Software Process Improvement Lifecycle Framework for the Medical Device Industry*, Proceedings of 12th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2005), Greenbelt, MD, USA, 3-8 April, 2005, Pages 273-280

(McCaffery et al. 2006) McCaffery, F., Smite, D., Wilkie, F. G., McFall, D., *A Proposed Way for European Software Industries to Achieve Growth Within the*

Global Marketplace, Software Process Improvement And Practice, *Softw. Process Improve. Pract.* 2006; 11: 277–285

(McCaffery et al. 2007) McCaffery, F., Taylor, P., Coleman, G.: *Adept: A Unified Assessment Method for Small Software Companies.* IEEE Software 24(1), 24–31 (2007)

(McConnell 1997) McConnell, S. *Software's Ten Essentials*, *IEEE Software* (14:2) 1997, pp. 143-144.

(McConnell 1999a) McConnell, S. *Software Engineering Principles*, *IEEE Software* (16:2) 1999, pp. 6-8.

(McConnell 1999b) McConnell, S. After the Gold Rush: Creating a True Profession of Software Engineering, Microsoft Press, 1999.

(McConnell 2004a) McConnell, S. *Professional Software Development*, Addison-Wesley, 2004a.

(McConnell 2004b) McConnell, S. *Code Complete*, (2nd ed.), Microsoft Press, 2004b.

(McFall et al. 2003) McFall, D., Wilkie, F. G., McCaffery, F., Lester, N. G., Sterritt, R. (2003). *Software processes and process improvement in Northern Ireland.* 16th International Conference on Software & Systems Engineering and their Applications, Paris, France, December 1-10, ISSN: 1637-5033.

(McFeeley 1996) McFeeley, B. (1996). *IDEAL: A User's Guide for Software Process Improvement.* Handbook CMU/SEI-96-HB-001. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PE, USA.

(MEX-003 2008) Minutes of Meeting, ISO/IEC JTC1 SC7 WG24, Mexico, Mexico, November 10-14, 2008

(Mil 1978) Mil-STD-1679, *Software Development*, US Military Standard, 1978.

(Montoni et al. 2008) Montoni, M., Santos, G., Vasconcellos, J., Figueiredo, S., Cabral, R., Cerdeiral, C., e Katsurayama, A., Lupo, P., Zanetti, D., Rocha, A., *Application of the SPI-KM Approach to Support the Implementation of the MPS Model in Small and Medium-Sized Enterprises in Brazil*, Software Quality Professional, ASQ, Volume 11, Issue 1, 2008, p 34-45

(Moore et al. 2001) Moore, J.W., Bourque, P., Dupuis, R., Abran, A., and Tripp, L.L. *Software Engineering Body of Knowledge Project*, in: *Encyclopedia* *of Software Engineering*, Marciniak, J. (ed.), Wiley, 2001, pp. 1401-1409, (Available as a sample article at: http://www.mrw.interscience.wiley.com/ese/index.html).

(Moore 2002) Moore, G. Crossing the Chasm: Marketing and Selling Technology Products to Mainstream Customers (revised ed.). New York: HarperCollins Publishers. 2002.

(Moore 1999) Moore, J., An Integrated Collection of Software Engineering Standards, IEEE Software, Nov 1999.

(Moore 2006) Moore, J., *The Road Map to Software Engineering: A Standards-Based Guide*, Wiley-IEEE Computer Society Press, 2006.

(MON-003 2007) Minutes of Meeting, ISO/IEC JTC1 SC7 WG24, Montréal, Canada, Oct. 29 - Nov. 2, 2007.

(MOS-003 2007) Minutes of Meeting, ISO/IEC JTC1 SC7 WG24, Moscow, Russia, 21-25 May 2007.

(MOS-023 2007) *Consolidated Requirements Document*, ISO/IEC JTC 1 WG24 Meeting, Moscow, May 2007.

(Nambisan 2002) Nambisan, S., *Software firm evolution and innovation–orientation*, J. Eng. Technol. Manage. 19 (2002) 141–165.

(Northrop et al. 2006) Northrop, L., Feiler, P., Gabriel, R. P., Goodenough, J., Linger, R., Longstaff, T., *Ultra-Large-Scale Systems: The Software Challenge of the Future*. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2006

(NORMAPME 2003) *SMEs Issues Concerning Standardisation*, NORMAPME, October 2003. www.normapme.com

(NMX 2005) NMX-059-NYCE, Information Technology-Software-Models of Processes and Assessment for Software Development and Maintenance. Part 01: Definition of Concepts and Products; Part 02: Process Requirements (MoProSoft); Part 03: Guidelines for Process Implementation; Part 04: Guidelines for Process Assessment (EvalProSoft). Ministry of Economy, Mexico, 2005.

(OCDE 2002) Organisation for Economic Co-operation and Development (OECD) *Small and Medium Enterprise (SME) Outlook Report*, 2002.

(O'Connor 2009) O'Connor, R., Project Management Deployment Package, Technical report, Laporte et al. (Eds.), 2009.

http://profs.logti.etsmtl.ca/claporte/English/VSE/index.html

(Oktaba 2006) Oktaba, H.: MoProSoft®: A Software Process Model for Small Enterprises. In: Proceedings of the First International Research Workshop for Process Improvement in Small Settings, pp. 93–101. Carnegie Mellon University, Pittsburgh (2006)

(Okataba et al. 2007) Oktaba, H., Felix G., Mario P., Francisco R., Francisco P. and Claudia, A.; *Software Process Improvement: The Competisoft Project*, IEEE Computer, October 2007, Vol. 40, No 10

(Okataba et al. 2008) Oktaba, H., Piattini, M., Editors, *Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies*, Idea Group Inc, Hershey, PA. 2008

(Palza 2009) Palza, E., Verification and Validation Deployment Package Technical report, Laporte et al. (Eds.), 2009, http://profs.logti.etsmtl.ca/claporte/English/VSE/index.html

(Parnas 1998) Parnas, D.L. *Software Engineering Programmes Are Not Computer Science Programmes*, CRL Report no. 361, McMaster University, Hamilton, Ontario, April, 1998.

(Paulk et al 1993) Paulk, M., Curtis, B., Chrissis, M. B., Weber, C. (1993). *Capability Maturity Model for Software*, Version 1.1, Software Engineering Institute, CMU/SEI-93-TR-24, February 1993.

(Perry 1994) Perry, Dr. William J., *Specifications and Standards – A New Way of Doing Business*, Memorandum from Secretary of Defense, 1994

(Pfleeger et al. 1994) Pfleeger S.L., Fenton N.E., Page P, *Evaluating software engineering standards*, IEEE Computer, 27(9), 71-79, Sept, 1994.

(Pfleeger 1998) Pfleeger, S, L., *Understanding and Improving Technology Transfer in Software Engineering, A State-of-the-Art-Report*, Data & Analysis Center for Software (DACS), Rome, N.Y., U.S. 1998, http://www.dacs.dtic.mil

(Pfleeger 1999) Pfleeger, S., *Understanding and improving technology transfer in software engineering*. Journal of Systems and Software 47 (2,3), 111–124, 1999.

(Pfleeger et al. 2003) Pfleeger, S.L., Wang, M., Adamson, D., Bloom, G., Butz, W., Fossum, D., *Facilitating Technology Transfer of Federally Funded R&D*, RAND Science and Technology Policy Institute, Arlington, VA, January 2003.

(Phongpaibul et al. 2005) Phongpaibul, M., Boehm, B., *Improving Quality Through Software Process Improvement in Thailand: Initial Analysis.* Proceeding of 3-WoSQ, ICSE 2005, 17 May, 2005.

(Pino et al. 2008) Pino, F., Garcia, F., Piattini, M.: *Software Process Improvement in Small and Medium Software Enterprises: A Systematic Review.* Soft. Quality Journal 16(2), 237–261 (2008)

(Pino et al. 2009) Pino, F.J., Hurtado, J., Vidal, J., García, F., Piattini, M., *A process for driving process improvement in VSEs*, ICSP 2009, LNCS 5543, pp. 342–353, 2009.

(PMI 2008) A Guide to the Project Management Body of Knowledge, Project Management Institute, 2008.

(Popick et al. 1996) Popick, P.R., Shear, S.A., *Ten lessons Learned from Implementing Integrated Product Teams*, Proceedings, International Council on Systems Engineering 6th Annual International Symposium, Boston, July 7-11 1996.

(Poulin 1996) Poulin, L., *Software: Process Risks Identification, Mapping and Evaluation - The S:PRIME Approach,* communications presented to the European SEPG of June 1996 (Amsterdam) and at the 1996 Systems Engineering Workshop (Santiago), March 1996.

(Pressman 2004) Pressman, R. *Software Engineering – A Practitioner's Approach*, 6th edition, McGraw-Hill, 2004

(Punter et al. 2009) Punter, T.,Krikhaar, R.L., Bril, R.J., *Software engineering technology innovation – Turning research results into industrial success.* J. Syst. Software (2009), doi:10.1016/j.jss.2008.12.035

(RTCA 1992) RTCA/DO-178B. 1992. Software Considerations in Airborne Systems and Equipment Certification, Washington, RTCA Inc.

(Radice 1985) Radice, R., *A Programming Process Architecture*, IBM Systems Journal, vol. 24, no. 2, 1985

(Radice 2002) Radice, R.. *High quality low cost software inspections.* Andover, MA: Paradoxicon, 2002.

(Redwine et al. 1985) Redwine, S.T., Riddle, W.E., 1985. *Software technology maturation.* Proceedings of the Eighth International Conference on Software Engineering. IEEE Computer Society Press, Los Alamitos, California, pp. 189-200.

(Revankar et al. 2006) Revankar, A., Mithare, R., Nallagonda, V.M., *Accelerated Process Improvements for Small Settings*, in: Proceedings of the First International Research Workshop for Process Improvement in Small Settings, 2005. Software Engineering Institute, Carnegie Mellon University, CMU/SEI-2006-Special Report-001, January 2006, p 117-125.

(Reifer 2004) Reifer, D., *Industry Software Cost, Quality and Productivity Benchmarks.* DACS Newsletter, Volume 7, Number 2, 2004.

(Rheault 2005) Rheault, Frédéric, *Software Process Performance Improvement Project for a Government Organization in Quebec*. Masters Thesis, École de technologie supérieure, Montréal, Canada. Co-directed with Professor Michel Plaisant of UQAM, 2005.

(Rheault et al. 2006) Rheault, F., Laporte, C.Y., *A Software Process Performance Improvement Program at a Governmental Institution of the Province of Québec.* Programme d'amélioration de la performance des processus logiciels dans une société d'État québécoise. Revue Génie Logiciel, Number 76, March 2006, pp 17-34.

(Richardson et al. 2007) Richardson, I., Wangenheim, C.G., *Why are Small Software Organizations Different?* IEEE Software 24(1), 18–22 (2007)

(Rocha et al. 2007) Rocha1, A., Montoni, M., Weber, K., Araújo, E., *A Nationwide Program for Software Process Improvement in Brazil*, Sixth International Conference on the Quality of Information and Communications Technology, 2007

(Rogers 2003) Rogers, Everett M., *Diffusion of Innovations*, fifth edition, Free Press, New York, 2003.

(Rout 1997) Rout, T.P. *Educating Software Engineers for Quality*, Pacific NorthWest Software Quality Conference, Portland, Oregon, 1997, pp. 14-23.

(Rout et al. 2000) Rout, T., Tuffley, A., Cahill, B., Hodgen, B. *The Rapid assessment of software process capability*, Proceedings of SPICE 2000, Limerick, Ireland, pp. 47-55.

(Royce 1970) Royce, W. *Managing the Development of Large Software Systems*, Reprinted in 9th International Conference on Software Engineering, IEEE Computer Society Press, 1970, pp. 328-338.

(Ruokolainen et al. 2007) Nambisan, S., *Software firm evolution and innovation–orientation*, J. Eng. Technol. Manage. 19 (2002) 141–165.

(Sanyakorn 2009) Sanyakorn, B., Version Control Deployment Package Technical report, Laporte et al. (Eds.), 2009. http://profs.logti.etsmtl.ca/claporte/English/VSE/index.html

(SC7 2004) ISO/IEC JTC1/SC7, Vision of Liaison Outcomes: IEEE CS and ISO/IEC JTC 1/SC7, June 2004, SC7 N3069 (available at http://www.JTC1-SC7.org)

(Schulmeyer et al. 1999) Chulmeyer, G.G., McManus, J.I., *Handbook of Software Quality Assurance*, Prentice Hall, Third edition, 1999.

(SECOR 1990a) A feasibility Study for a Canadian Software Engineering Center to Serve the Aerospace and Defense Sectors, SECOR (Montréal), April 1990.

(SECOR 1990b) The Markets for a Canadian Software Engineering Center to Serve the Aerospace and Defense Industries, SECOR (Montréal)' August 1990.

(SECOR 1990c) Canadian Software Engineering Consortium-Business Plan, SECOR (Montréal) December 1990.

(Sédion 2007) Sédion, Yacouba, *Development and Deployment of a Project Management Office at Vircom Inc*, Masters Thesis, École de technologie supérieure, Montréal, Canada 2007.

(SEI 2006) SEI, Capability Maturity Model Integration, v1.2, Software Engineering Institute, Pittsburgh, USA, 2006.

(Sheard 2001) Sheard, S., *Evolution of the Frameworks Quagmire*, IEEE Computer, July 2001, pp. 96-98.

(Shintani 2006) Shintani, K, *Empowered Engineers are Key Players in Process Improvement*, Presentation at the First International Research Workshop for Process Improvement in Small Settings, Software Engineering Institute, CMU/SEI-2006-SR-01, Pittsburgh, PA, 2006

(Siddall 1996) Siddall, Donald, R., *The Team Notebook*, Publication in process, 1996

(Socarras 2009) Socarras, R., Développement et mise en œuvre d'un plan d'amélioration des processus de génie logiciel pour le groupe de développent de systèmes embarqués chez Gemalto Canada Inc. Projet de maîtrise en génie logiciel, École de technologie supérieure. En cours de rédaction.

(SPC 1992) SPC, *Process Definition and Modeling Guidebook*, Software Productivity Consortium, SPC-94095-CMC, January, 1992

(SPC 1995) SPC, *A Tailorable Process for Systems Engineering*, Software Productivity Consortium, SPC-94095-CMC, January, 1995

(SPC 1996) SPC, *Integrated Systems and Software Engineering Process*, Software Productivity Consortium, SPC-96001-CMC, May 1996.

(SPC 1998) *Interpreting the CMM for Diverse Software Environments*, Software Productivity Consortium, SPC-98023-MC, October 1998.

(SPIRE 2000) The SPIRE Handbook - *Better, Faster, Cheaper, Software Development in Small Organisations*, Centre for Software Engineering, Dublin City University Campus, Dublin 9, Ireland, 2000.

(Stambollian 2006) Stambollian, Anabel, *Amélioration de la performance des processus logiciels dans les petites et moyennes entreprises*. Masters Thesis, École de technologie supérieure, Montréal, Canada Project co-directed with Professor J-M Desharnais of ÉTS, 2006.

(Stambollian et al. 2006) Stambollian, A., Laporte, C.Y., Renault, A., Desharnais, J. M., Habra, *Initiating Software Process Improvement in Small Enterprises: Experiment with Micro-Evaluation Framework*, SPICE Conference, May 4-5, 2006, Luxembourg, pp 97-105.

(Standish 2003) Standish Group, The Chaos Report, 2003.

(Staples et al. 2007) Staples, M., Niazi, M., Jeffery, R, Abrahams, A., Byatt, P., Murphey, R., *An exploratory study of why organizations do not adopt CMMI,* The Journal of Systems and Software 80 (6), 883-895, 2007.

(Statz et al. 1997) Statz, J., Oxley, D., O'Toole, P., *Identifying and Managing the Risks for Software Process Improvement*, Crosstalk, Volume 10, No. 4, April 1997

(Stelzer et al. 1997) Stelzer, D., Mellis, W., Herzwurm, G., *A critical look at ISO 9000 for software quality management*, Software Quality Journal 6, (1997) 65–79

(Stelzer et al. 1998) Stelzer, D., Mellis, W., *Success Factors of Organizational Change in Software Process Improvement*, Softw. Process Improve. Pract. 4, 227–250 (1998)

(Suryn et al. 2002) Suryn, W., Bourque, P., Abran, A., Laporte, C.Y., "Software Product Quality Practices Quality Measurement and Evaluation using TL9000 and ISO/IEC 9126". Actes - IEEE Computer Society, Tenth International Workshop – Software Technology and Engineering Practice, Montréal, 2002.

(Suryn et al. 2004) Suryn W., Abran A., Laporte C.Y., "An Integrated Life Cycle Quality Model for General Public Market Software Products". Actes-12th International Software Quality Management & INSPIRE Conference (BSI) 2004, Canterbury, Kent, UK 5-7 Avril 2004.

(Sutton 2000) Sutton, S.M., *The Role of Process in a Software Start-up*, IEEE Software July/August 2000, p33-39

(Tomayko 2001) Tomayko, J.E. Milestones in Software Engineering, in: Encyclopedia of Software Engineering, Marciniak, J. (ed.), Wiley, 2001.

(Thanasankit et al. 2000) Thanasankit, T., Corbitt B., Cultural Context and its Impact on Requirements Elicitation in Thailand. The Electronic Journal on Information Systems in Developing Countries, http://www.ejisdc.org, 2000.

(Thailand 2005) *Thai Quality Standard*, Association of Thai Software Industry, Presentation to the Special Working Group Meeting, Bangkok, Thailand, March 2005.

(Turner 2003) Turner, R., *Seven Pitfalls to Avoid in the Hunt for Best Practices*, IEEE Software, Jan/Feb 2003.

(Weber et al. 2005) Weber, K., Araújo, E., Rocha, A., Machado, C., Scalet, D., Salviano, C.: *Brazilian Software Process Reference Model and Assessment Method*. In: Computer and Information Sciences, pp. 402–411. Springer, Berlin (2005)

(Zelkowitz 1995) Zelkowitz, Marvin V., *Assessing software engineering technology transfer within NASA*, NASA technical report NASA-RPT-

003095, National Aeronautics and Space Administration, Washington, DC, January 1995.

(Varkoi et al. 1999) Varkoi T, Mäkinen T. (1999). Software process improvement network in the Satakunta region – SataSPIN. In Proceedings of the EuroSPI'99, Pori, Finland.

(Varkoi 2009) Varkoi, T., Self-Assessment Deployment Package, Technical report, Laporte et al. (Eds.), 2009. http://profs.logti.etsmtl.ca/claporte/English/VSE/index.html

(Vasquez 2009) Vasquez, A. , Construction and Unit Testing Deployment Package, Technical Report, Laporte et al. (Eds.) 2009, http://profs.logti.etsmtl.ca/claporte/English/VSE/index.html

(Vincenti 1990) Vincenti, W.G. *What Engineers Know and How They Know It* – Analytical Studies from Aeronautical History, Johns Hopkins, Baltimore and London, 1990.

(Weinberg 1997) Weinberg, G., *Quality Software Management*, Vol 4: Anticipating Change. Dorset House, 1997.

(Wiegers 1996) Wiegers, K.E. *Creating a Software Engineering Culture*, Dorset House Publishing, New-York, 1996.

(Wiegers 2003) Wiegers, K.E. *Software Requirements*, (2nd ed.), Microsoft Press, 2003.

(Wiegers 2005) Wiegers, K., Software Process Improvement handbook, 2005. Document available from www.processimpact.com

(Wilkie et al. 2005) Wilkie F. G., McFall, D. and McCaffery, F., *An evaluation of CMMI process areas for small- to medium-sized software development organisations*, Software Process: Improvement and Practice, Volume 10, Issue 2, Date: April/June 2005, Pages: 189-201

(Wilkie et al. 2005) Wilkie F. G., McFall, D. and McCaffery, F., *The Express Process Appraisal Method*, Proceedings of 5th International SPICE Conference (SPICE 2005), Klagenfurt, Austria, 27-29 April 2005, Austrian Computer Society, ISBN 3-85403-190-4, pp. 27-36.

(Wilkie et al. 2007) Wilkie, F. G., Mc Caffery, F., McFall, D., Lester, N., Wilkinson, E., *A Low-overhead method for software process appraisal*, Software Process: Improvement and Practice Volume 12, Issue 4, July/August 2007, Pages: 339-349

(Zelkowitz 1995) Zelkowitz, M.V., 1995. *Assessing software engineering technology transfer within NASA*. NASA Technical report NASA-RPT-003095. National Aeronautics and Space Administration, Washington, DC.

Appendix A

Historical Perspectives and Accomplishments

1 Appendix A - Historical Perspectives and Accomplishments

1.1 Introduction

This appendix describes the accomplishments of the author in order for the reader to better understand the involvement and motivations of the author in the development of international standards for Very Small Entities (VSEs). This description is presented in two parts: the accomplishments of the author between 1976 and 2000 and the era between 2000 to the present time.

1.2 Accomplishments from 1976 to 2000

At the graduating from the Collège militaire royal (CMR) de Saint-Jean in 1973, a physics professor invited me, if the author wanted, to solicit a position of military professor at CMR. The Department of National Defense (DND) would totally sponsor graduate studies at a university of my choice anywhere in the world. In 1976, my request to pursue graduate studies was approved by DND. The objective was to introduce in the undergraduate program of CMR two new courses in digital electronics: a course on logic circuit design and a course on micro processor system design. The author decided to pursue graduate studies at Université de Montréal where, as his project, he designed and built a micro-computer based data acquisition system (Laporte 1980). In those days, developing a 4-kilobyte assembler software was quite a challenge.

The undergraduate students had to do a capstone project during the last 2 semesters of their program. Some of them, having developed an interest in micro-computer, started to request for projects where they would design and build a small system. After their graduation, some of them posted to the DND Headquarter contacted the author to request the development of prototypes or evaluation systems. They were willing to fund these small projects providing that they were documented well enough for their projects using the current Military Standards developed for the US Navy (Military 1978). As an example, one project consisted in designing a special display system, titled Radar Situation Display, for the radar control system of military airports (Laporte et al. 1986a). This project had been conducted as part of a 60 million$ larger project to modernize air traffic control systems of military airports. The project consisted in receiving data from the airport terminal, filtering the data in order to display targeted located within a radius of 32 nautical miles and lower than 10,000 feet altitude.

When the college started a new undergraduate program in computer science in the late 80's, the author was invited by a dean to obtain a Master degree in computer engineering to be eligible for tenure, as a civilian professor, at the college. The author then completed, at the electrical and computer engineering department of École Polytechnique de Montréal, a project in designing a computer-aided voice control system for military high performance aircrafts (Laporte 1986b). This project was later delivered to the military and published in two refereed journals (Laporte 1986c, Laporte 1989).

At that time, the author made many visits to the Software Engineering Institute (SEI) of Carnegie Mellon University and quickly realized that software engineering will soon be a required discipline in the Canadian DND. The military had already acquired a fleet of sophisticated jet fighters (F-18), anti-submarine and patrol aircrafts and frigates. When visiting his colleague at the Headquarter, it became evident that DND would be acquiring sophisticated systems having one thing in common: large critical computer-based systems, requiring high availability, high maintainability and very long operational lifecycles. As an example, the jetfighter, acquired in the early 80's are presently being upgraded to stretch their operation well into the 2020's. The author had then proposed to the authority of the military college the establishment of a Defense Software Engineering Center.

1.2.1 Project Manager for the establishment of a Defense Software Engineering Center.

The author was nominated in June 1988, by the dean of science and engineering, as project manager for this project. A steering committee was established mainly from senior executives of large defense and aerospace contractors, such as Bombardier Aerospace, Oerlikon Aerospace, CAE, SPAR Aerospace and Lockheed Martin (then called Paramax), of the Montréal area. A 250,000$ feasibility study was sponsored by 13 defense and aerospace companies and the Québec and federal governments with the participation of the Collège militaire royal de Saint-Jean. The objectives of the study were as follow (SECOR 1990a):

- To determine the markets, products and services for a Canadian Software Engineering Center.
- To analyze the international competitors to such a center.
- To determine the precise mission, the overall role, the objectives and the responsibilities of such a center.
- In light of the foregoing, to assess the degree of feasibility of such a center, particularly its financial and economic prospects.

- To layout an action plan which will allow the Center to achieve its objectives.

The feasibility study also aimed to achieve the following more specific objectives:

- To determine the role and importance of software engineering in the defense and aerospace sectors.
- To evaluate the current state of software engineering in the defense and aerospace sectors as well as the major problems faced as compared to the situation in other countries.
- To identify the principal problems which the defense and aerospace sectors must face without such a center.
- To layout the feasible options for such a center as a function of the role it would play. These options would include funding and organizational scenarios.
- To select one option and to analyze the requisite investment costs, ongoing operating costs and economic impact.

As part of the feasibility study, a first exposure to the software process assessment methodology developed by the Software Engineering Institute (SEI) was done in Montréal in the summer of 1989. Two members of the technical staff of the SEI conducted a one-day workshop at École Polytechnique of Montréal. The workshop was attended by 50 persons. The participants came mainly from defence, aerospace and finance organizations, of both the private and public sectors. During the workshop, the participants answered the SEI questionnaire that was used to conduct formal assessments (Humphrey 1987). The questionnaires were compiled, and the results, as illustrated in Table A-1, were that 93% of the participants to this workshop worked for organizations at the initial maturity level (level 1) and the remaining 7% were at the repeatable level (level 2) of the maturity scale. Although the assessment of organizations according to the SEI's approach would have been far more stringent, these results remain nevertheless indicative of the situation prevailing at that time.

As a comparison, the United States conducted similar workshops and gathered data from 113 projects (Humphrey 1989). The assessment workshop results as of January 15, 1989, indicate that the majority (86%) of the participants reported projects at the initial level (level 1). Fourteen per cent (14%) of the participants reported projects at the repeatable level (level 2) and one per cent (1%) reported projects at the defined level (level 3).

Maturity Level	US Results	Results of the Workshop
1	86%	93%
2	13%	7%
3	1%	0%
4	0%	0%
5	0%	0%

Table A-1 Process Maturity Profiles of US and Canadian high-tech firms

These results meant that software development in Canada was more often than not an ad hoc process, without formal procedures, cost evaluations or planning and, even where procedures do exist; there are no management mechanisms to ensure their use. A consequence of this lack of process maturity was that the federal government and large Canadian corporations awarded millions of dollars worth of software engineering contracts to US and other foreign companies.

Given the importance of information technologies as a productivity factor, the lack of competence in software engineering had a direct impact on the competitiveness of Canadian companies in all sectors of activity.

Canadian companies therefore did not qualify for the preliminary selection process established for US government suppliers. As an example, the US Department of Defense stated that to qualify as a prime contractor, a company must have processes compliant with the requirements of level 3 on the basis of the SEI SW-CMM model.

Following the tutorial held at École Polytechnique, some organizations decided to conduct software process assessments and improvement activities.

Encouraged by the results of the feasibility study (SECOR 1990a, SECOR 1990b), the study's sponsors decided in 1990 to draw up a five-year business plan (SECOR 1990c) aimed at creating a software engineering centre, the mission of which would be to assume a leadership role the technological level and to assist industry, where such an expertise is required, to improve their competencies in software engineering. In 1991, the Applied Software Engineering Centre became a division of CRIM.

The mission entrusted to the Applied Software Engineering Centre was to provide access to and training in the best software engineering managerial and technical solutions. Its target clients comprise companies and agencies that rely on information technology to improve the productivity and quality of their services and products. ASEC offered four main categories of services: services related to software engineering process such as software process assessment, auditing of suppliers' competencies and advising, training, awareness to new technologies by means of appropriate activities, as well as implementation of and relevant support to specific interest groups. ASEC is also part of a network of similar centres subsidised by the federal government.

The author participated, as a military of the rank of Major (Commandant) as the instigator of this project, to each phase that led to the establishment of the Applied Software Engineering Center (ASEC). The author was nominated as interim director of ASEC while the selection process of a civilian director was conducted.

The business plan described the sales and financial projections as follow:

- Sales of services are expected to generate around $200,000 in the first year and $2.4 million over a five-year period.
- Operating expenses and costs related to creating projects and providing services will exceed $1 million in the first year and total nearly $6.7 million over five years.

A small portion of this deficit will be offset by contributions from the Centre de recherche informatique de Montréal (CRIM). The Applied Software Engineering Centre and CRIM rely on funding from the Québec and federal governments, which will provide $1 million and $3 million respectively over five years, to help finance these activities.

The decision to ask the federal government for a proportionally larger grant is justified by the fact that the Québec government had already given CRIM a $25 million grant over five years. It was also mitigated by the national vocation of the founding member of ASEC, as well as by the Centre's mission which was to offer services throughout Canada.

The author also participated, in 1990, to a Software Process Assessment using the SEI's assessment method at CAE Electronics, in collaboration with Bombardier-Canadair. This division of CAE Electronics is responsible for the maintenance of the software of the Canadian Armed Force's CF-18 fleet. For this assessment, it was decided that the assessment team would be composed of representatives from the customer's organization as well as representatives from the assessed organization. The on-site assessment was performed in February of 1991 and the action plan was published in September. The costs of process assessment and improvement activities (Lambert 1992)) are in Table A-2. The data illustrates clearly the high cost of performing an assessment, i.e. 40,000$ for the assessor fees and about 400 hours of labour. These costs do not account for the costs of the action plan (500 hours) and the estimated cost of improvements (2,500 hours). For example, at a rate of 100$ per hour, the cost of performing an assessment and putting in place the improvement actions would be of 380,000$. Only large organizations can afford these costs, most VSEs cannot afford just the cost of the assessment, let alone the larger costs of improvements.

Assessment training and consulting cost:		Cdn $40,000
Labour:		
	Training	160 hours
	On-site assessment	240 hours
	Action plan elaboration	500 hours
	Action plan implementation	2,500 hours

Table A-2 Assessment and Improvement Costs (Lambert 1992)

Table A-3 lists organizations known by the author that are actively involved in software process engineering activities. So far, most assessments were performed by large organizations, using the SEl's approach. ASEC performed at least five assessments since April 1994 and expects to conduct another five in 1996-97. Since in Québec the number of small and medium organizations outnumbers the number of large organizations, we expect a growing use of the Software: Process Risk Identification Mapping and Evaluation method (S:PRIME) (Poulin 1996). Finally, it is expected that SPICE will become and ISO standard in 1998. It is possible that organizations choose to wait two or three years before deciding whether to adopt this type of assessment or stay with the SEI's approach. It is also possible that the SEI decides to map its maturity model to the SPICE framework. It is worth mentioning that the SEI is collaborating to the development of a System Engineering Capability Maturity Model (CMM). This CMM is using a framework nearly identical to the SPICE framework for the mapping of process areas and maturity levels (Bate 1995).

Organization	Sector	Year	Activity
CAE Electronics and Bombardier-Canadair	Defence	1991	SEI - SPA (1)
Loral Canada	Defence	1991	SEI - SPA
Hydro-Québec	Utility	1993	Internal assessment using SEI - CMM
Oerlikon Aerospace	Defence	1993	SEI - SPA
Scotia Bank (Montréal-Trust)	Finance	1993	SEI - SPA
CAE Electronics	Energy Management	1993	SEI - SPA
Hydro-Québec- IREQ	Utility - Research	1994	Internal assessment using CMM
Ericsson	Telecommunications	1994	SEI - CBA / IPI (3)
CAE Electronics and Bombardier-Canadair	Defence	1994	SEI - CBA / IPI (4)
Canadian Marconi	Defence	1994	SEI - CBA / IPI
M3i	Network Management	1994	S:PRIME
Hydro-Québec	Utility Automatisms	1995	SPICE
IST Group	Information Systems	1995	S:PRIME
CRIM	Research & Development	1995	S:PRIME

Table A-3 Software Process Activities in Québec

NOTES :

8. SEI - SPA: Software Engineering Institute Software Process Assessment with third party.

9. Internal assessment using SEI - CMM conducted without participation of third party.

10. SEI - CBA/IPI: SEI - CMM based-assessment: Internal Process Improvement with third party together with additional practices.

11. SEI - CBA/IPI: SEI - CMM based-assessment: Internal Process Improvement with third party.

THE MONTRÉAL SOFTWARE PROCESS IMPROVEMENT NETWORK

Montréal was the host of a SPIN (Software Process Improvement Network). Essentially, a SPIN is an interest group composed of software professionals from industry, government, academia, professional organizations, and consulting agencies. The SPIN provides a forum for the free and open exchange of information on software process improvement. The SEI provides some support to the SPIN. In fact, the SPIN in Montréal is part of an international network of interest groups called "SPIN for Software Process Improvement Network". The Montréal SPIN was co-founded and managed in 1993 by the author. Its mission is to facilitate the understanding, the adoption and the deployment of proven or innovation solutions for software process improvement. Each year, the SPIN organizes events such as tutorials, workshops and round tables. The SPIN is affiliated to the Applied

Software Engineering Centre; the meetings are generally held at ASEC facilities. In addition, the SPIN benefits from the administrative services offered by ASEC (e.g. mailing, reservation, accounting). The co-operation between the Montreal-SPIN, ASEC and the SEI will give rise to an international symposium on software process improvement to be held in Montreal in October 1996. This symposium will aim at gathering managers, professionals and contributors intervening in the continuous implementation and improvement of system and software processes. It represents a unique opportunity to perfect participants' knowledge and enrich their vision by sharing their experience and concerns on subjects such as investing, stakes, risks, profits and international trends re process improvement.

1.2.2 Project Manager for the development of a graduate program in software engineering

The author had also proposed, to the authority of the military college as well as to civil servants of the Department of Industry, Science and technology Canada, the development of a graduate program in software engineering. He was then nominated, by the dean of science and engineering of CMR, as project manager for this new program. The author received immediate and full support from the general officers responsible for the engineering and maintenance of military equipments of DND. The graduate program in software engineering, the first in Canada, had its first military officers graduated in1992. The author was also given a grant in 1991, of 25,000$, from the Department of Industry, Science and technology Canada to conduct a feasibility survey for the development of a multi-campus graduate program in software engineering in the Montréal area. Once the study was successfully completed, a steering committee composed of the deans of the computer science and engineering faculties, was established by the Centre de recherche informatique de Montréal (CRIM).

1.2.3 Senior Analyst at Oerlikon Aerospace

This section describes the professional duties, from 1992 to 1999, at Rheinmetall Canada (then called Oerlikon Aerospace) as a senior analyst responsible to coordinate the development and deployment of software engineering and systems engineering processes and project management processes. The system consists of five technology/product families: processing and display; platform system; sensors and effectors; command, control, communication and intelligence; and readiness system (e.g. training, simulators and test). About 60 software engineers and systems engineers are involved in the development and maintenance of the system. The software programs were written in a variety of languages ranging from assembler to Ada. The software programs have been documented using military standards such as 1679, 2167A and 498. Over 20 software engineers maintain the software assets. The software is divided in four domains: weapon software, command, control and communication software, simulation software, and instrumentation software.

Since working in for a defense manufacturer is quite unique, the author will start by a brief description of the system manufactured and maintained by his former employer. Then a description of the processes that had been developed to support the weapon system will be presented.

DESCRIPTION OF THE AIR-DEFENSE ANTI-TANK SYSTEM[74]

The Air-Defense Anti-Tank System (ADATS), as illustrated in figure A-1, is a low level short range air defence system, capable of engaging both air and surface targets. It was manufactured by Rheinmetall Canada (formerly Oerlikon Aerospace) in Québec. The first system was delivered in 1988. 36 systems were delivered, ending in 1994. The Royal Thai Air Force has one shelter-based system, linked to a Skyguard fire control system.

The ADATS is a self-contained system in an unmanned turret with FLIR (Forward-Looking Infrared) and TV sensors, laser rangefinder and designator, a search radar, and eight missiles. This turret is mounted on the vehicle, which is in current applications a modified armoured personnel carrier chassis. When the radar has detected a target, it is tracked by the passive optical sensors, which are immune to anti-radiation missiles and electronic counter measures. When the target is in range, a missile is fired, and guided along a digitally coded laser beam. The missile's anti-armour/anti-aircraft warhead is detonated either by a proximity fuze or by a mechanical impact fuze.

COMMAND AND CONTROL

The ADATS network coordinates up to six ADATS spaced at distances up to 20km. Any ADATS can be the network master controller and the network can link with other command facilities in real time. Real-time data exchange includes airspace control data, weapon control orders and fire control orders, target identification data, individual system status and vehicle position, threat prioritization and optimized weapon allocation, engagement status, weapon status and jammer triangulation data. The six-unit network can engage up to 48 air or ground targets.

ADATS MISSILE

The ADATS missile can engage all types of low-level threats, including attack helicopters exposed at stand-off ranges at extremely low altitudes. The system has a 10km range against air or ground targets. The missile has laser beam riding guidance and the laser

[74] The technical information in this section is adapted from: http://www.army.forces.gc.ca/lf/English/2_display.asp?product=65&more=65

guidance grid is digitally encoded for precision and immunity to countermeasures. The missile has a speed of Mach 3+ and manoeuvrability of 60g.

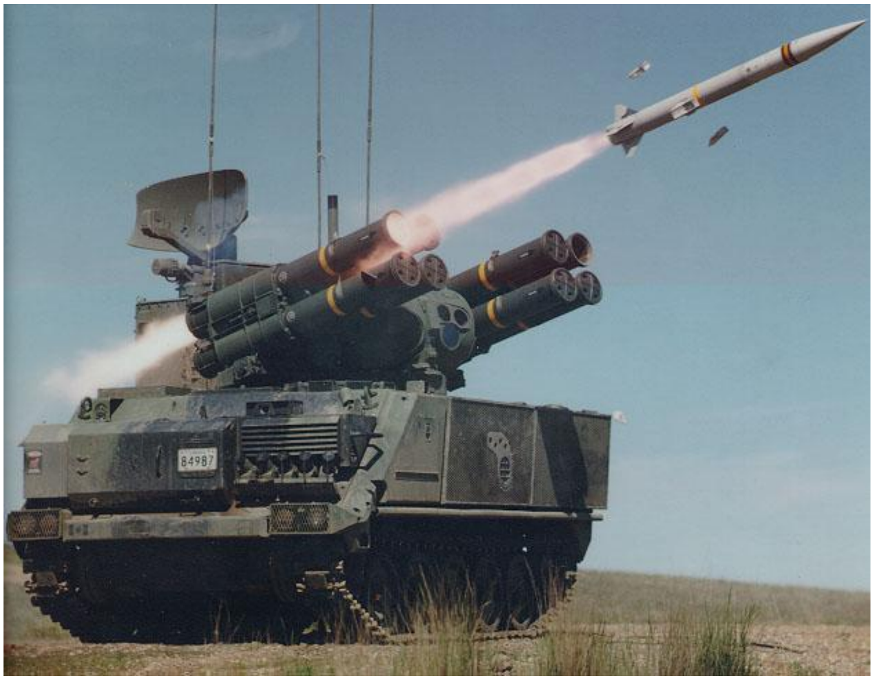


Figure A-1 ADATS System[75]

ENGAGEMENT SEQUENCE

Initial target detection to missile launch takes less than five seconds. The engagement sequence begins with target detection and turret slew using radar, FLIR and television systems against air targets and FLIR and television against ground targets. The tracker search and target acquisition sequence is carried out using the FLIR and the television sighting system.

Missile launch and guidance uses FLIR and television target tracking and carbon dioxide laser beam riding missile guidance. The time required to launch a second missile following completion of the first engagement is less than two seconds.

The ADATS in is set to be replaced by the Multi-Mission Effects Vehicles (MMEV) now in development. The MMEV will be set on the General Dynamics Land Systems 8x8 LAV III vehicle chassis and feature long-range non-line-of-sight anti-tank missiles, a new 3D radar systems and infra-red vision equipment. The replacement is set to take place in 2010.[76]

[75] http://www.army.forces.gc.ca/lf/English/2_display.asp?product=65&more=65

[76] http://www.army.forces.gc.ca/lf/English/2_display.asp?product=65&more=65

1.3 Development of Engineering Processes at Oerlikon Aerospace

In the following paragraphs, the author describes the steps taken by Oerlikon Aerospace to assess and improve its engineering processes. This description will help the reader better understand the challenges associated to the development and deployment of engineering processes in an organization. The author wants to illustrate how the new international standard and associated guide, presented in the next chapter, should help VSEs without having to go through the expensive and lengthy tasks of an organization such as Oerlikon Aerospace. The author led all the process improvement activities at Oerlikon Aerospace (Laporte et al 1993).

1.3.1 Development of Software Engineering Processes[77]

A software engineering process improvement cycle was initiated, in 1992, as illustrated in figure A-2, and completed in 1998. At the beginning of the improvement cycle, the executives decided to establish a Software Engineering Process Group (SEPG) and approved the conduct of a formal Software Process Assessment.

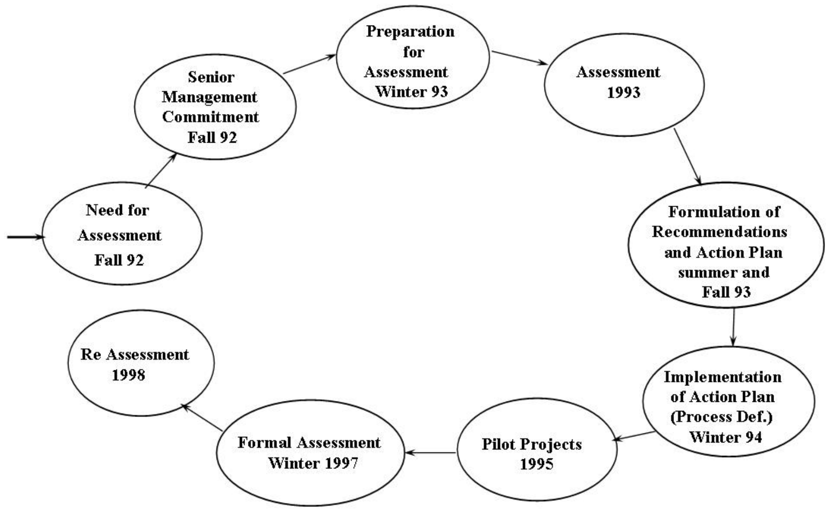
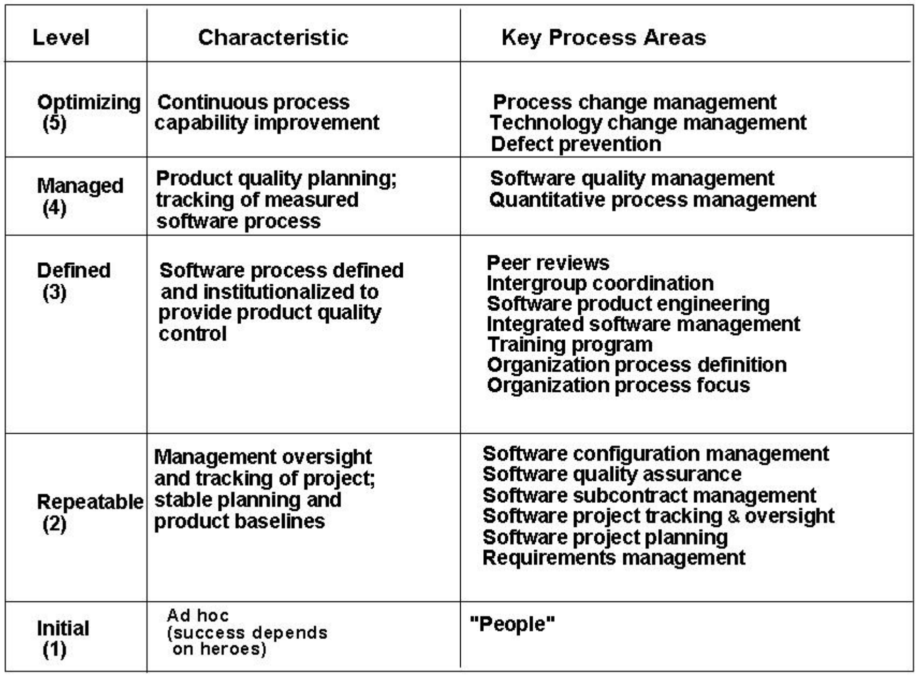

Need for Assessment Fall 92
Senior Management Commitment Fall 92
Preparation for Assessment Winter 93
Assessment 1993
Formulation of Recommendations and Action Plan summer and Fall 93
Implementation of Action Plan (Process Def.) Winter 94
Pilot Projects 1995
Formal Assessment Winter 1997
Re Assessment 1998

Figure A-2 Improvement Cycle at Oerlikon Aerospace

An action plan was developed by the SEPG using, as a framework, the Capability Maturity Model (CMM) for software (Paulk et al. 1993) as illustrated in Table A-4. Working groups were mandated to develop specific parts of the software process under the close coordination of the SEPG.

[77] This section is adapted from an article originally published in (Laporte et al. 1996)



Level	Characteristic	Key Process Areas
Optimizing (5)	Continuous process capability improvement	Process change management Technology change management Defect prevention
Managed (4)	Product quality planning; tracking of measured software process	Software quality management Quantitative process management
Defined (3)	Software process defined and institutionalized to provide product quality control	Peer reviews Intergroup coordination Software product engineering Integrated software management Training program Organization process definition Organization process focus
Repeatable (2)	Management oversight and tracking of project; stable planning and product baselines	Software configuration management Software quality assurance Software subcontract management Software project tracking & oversight Software project planning Requirements management
Initial (1)	Ad hoc (success depends on heroes)	"People"

Table A-4 Capability Maturity Model (Paulk et al. 1993)

The CMM has been developed in the late 1980's and early 1990's by the SEI with the participation of a large segment of the international software community. Version 1.1 was used in the assessment conducted at Acme Inc. In a CBA IPI assessment, the CMM is typically used to structure the data collection process and the data analysis that ensues. Throughout the assessment, the team used it to interpret the collected data and to identify the information still needed to determine if the assessed process satisfied the KPA goals. The CMM was being referred to in order to rate the maturity of the process.

The CMM, which is graphically depicted in Table A-5, is structured along five maturity levels. The CMM is made up of 316 practices grouped into 18 KPAs that represent aggregates of activities typically carried out in software engineering. Each KPA is mapped to a given maturity level and has from two to four Goals that allow an assessment team to determine if a KPA is mastered, based on a detailed examination of the practices associated with that KPA. This examination, which constitutes the core of the CBA IPI, brings out strengths and weaknesses of the assessed site with respect to those practices.

It is necessary to master all the goals of a KPA in order to conclude that that KPA is satisfied. It is then necessary to master all KPAs associated with a maturity level N in addition to those at all lower maturity levels to conclude that maturity level N is achieved.

1. Initial	The process is informal and largely improvised, and performance is unpredictable.
2. Repeatable	A software process is defined within the scope of each project and the focus is placed on stabilizing the approach used to carry out the work within each project taken individually.
3. Defined	An organizational software process is in place and consequently, the emphasis is directed at defining such a process from the best practices implemented in each project, at adapting the resulting process to the needs of each project, and at establishing and using a database in which the data gathered as part of performing the work is consolidated for use in connection with future projects.
4. Managed	The process defined at level 3 is gauged, quantified and characterized statistically, and the focus is placed on controlling the software process outputs with respect to statistical parameters.
5. Optimizing	The quantitative data is used to improve the process by reducing the value of the control parameters established at level 4. Continuous improvement becomes a way of life in order to satisfy the business objectives of the organization.

Table A-5 Description of the maturity levels of the CMM

After the CBA IPI onsite period had been completed, the CMM was used to prepare recommendations to be documented in the assessment report, to elaborate the improvement actions that were be part of the action plan, and to guide personnel assuming the responsibility of coordinating its implementation.

At Oerlikon Aerospace, the approach, to process engineering was fourfold: first define a process and bring it under management control; secondly, support the process with methods; thirdly, support the process and methods with appropriate tools; and fourth, train all personnel in the utilization of processes, methods and tools.

Essentially, the software process improvement initiative followed the five phases of the IDEAL model (McFeeley et al. 1996) as illustrated in Figure A-3. This model has been developed by the SEI based on their experience with large enterprises mainly of the defense sector. The five phases of the model are: Initiating the improvement program, Diagnosing the current state of practice, establishing the plans for the improvement, Acting on the plans and recommended improvements and, Leveraging the lessons learned and the business results of the improvement effort.

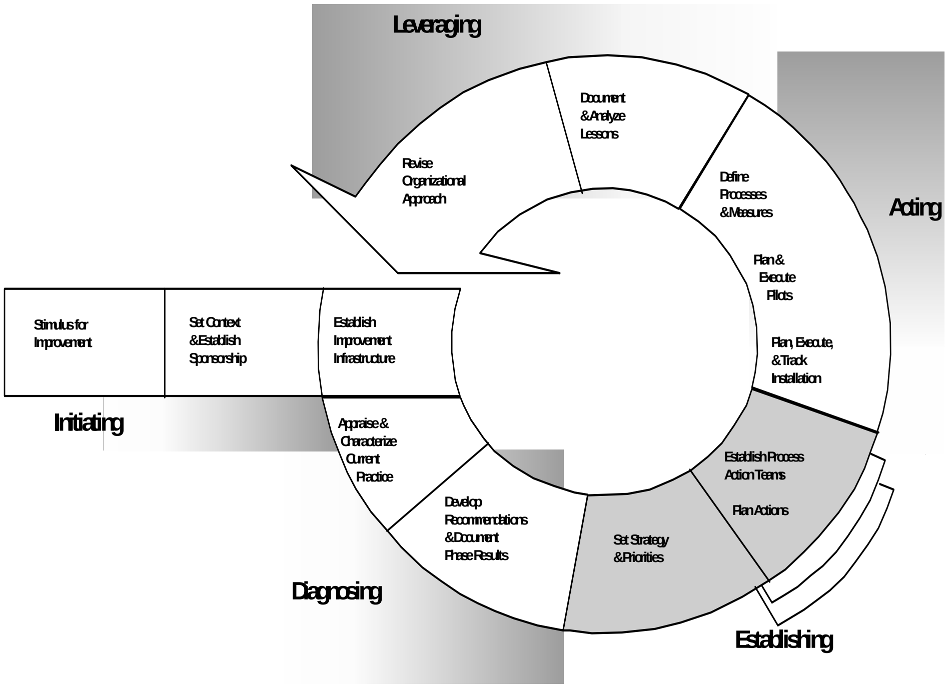

Leveraging
Document & Analyze Lessons
Revise Organizational Approach
Define Processes & Measures
Acting
Plan & Execute Pilots
Stimulus for Improvement
Set Context & Establish Sponsorship
Establish Improvement Infrastructure
Plan, Execute, & Track Installation
Initiating
Appraise & Characterize Current Practice
Establish Process Action Teams
Plan Actions
Develop Recommendations & Document Phase Results
Set Strategy & Priorities
Diagnosing
Establishing

Figure A-3 The IDEAL Improvement Cycle (McFeeley et al. 1996)

During the Initiating phase, a business case was prepared and presented to the president. Recognizing that software engineering was a core competence of Oerlikon Aerospace, the president approved the establishment of a Software Engineering Process Group (SEPG) (Fowler 1990). A budget was also approved for the conduct of a Software Process Assessment (SPA) and the development of an action plan. Briefing sessions were held to inform the organization about the software process improvement effort.

During the Diagnosing phase, a SPA was performed jointly by the SEPG and by independent assessors certified by the Software Engineering Institute (SEI). Strengths and weaknesses were identified and priorities for improvements were recommended. An action plan skeleton was presented, to the president, identifying the resources required for its implementation.

1.3.2 Summary Description of the CMM-Based Appraisal

The assessment method, as illustrated in figure A-4, is briefly described such that the reader has an appreciation of the high cost of the assessment process and the need to have expertise in the CMM model and assessment method required to perform such an elaborate assessment. As it will be demonstrated in a next chapter, an assessment method had been developed specifically for VSEs.

The CMM-Based Appraisal for Internal Process Improvement (CBA IPI) is a method for assessing an organization's software development or maintenance capability, using the CMM version 1.1 as a reference model of best software engineering practices, to enable a more effective process improvement program within the organization. The CBA IPI method is designed to support organizations in assessing their current software process. As such, it has two primary Goals:

- To support, enable, and encourage an organization's commitment to software process improvement;
- to provide an accurate picture of the organization's current software process strengths and weaknesses, using the CMM as a reference model, in order to identify key areas for improvement.

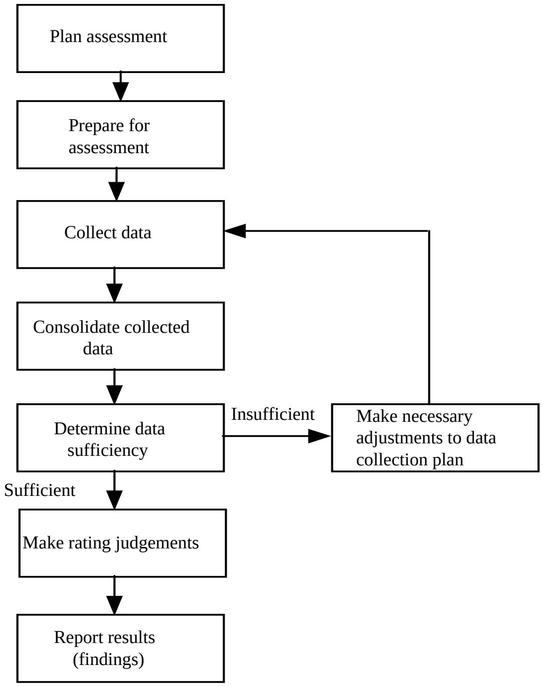

Plan assessment
Prepare for assessment
Collect data
Consolidate collected data
Determine data sufficiency
Insufficient
Make necessary adjustments to data collection plan
Sufficient
Make rating judgements
Report results (findings)

Figure A-4 CBA IPI approach

A CBA IPI involves three categories of personnel who actively participate in the assessment, namely, the sponsor, the assessment team and the assessment participants.

The sponsor is the person who acts as the recipient of the final findings resulting from the assessment and who has both the authority and the responsibility for providing resources for follow-on process improvement activities. The sponsor identifies to the team the business goals that bear on the organization's software development and maintenance activity and describes any current quality improvement initiatives in the assessed organization. He or she also negotiates the scope of the assessment with the assessment team leader and any existing constraints that may have an impact on the assessment. The sponsor gives the authorization to proceed and personally participates in the assessment opening meeting by urging participants to be forthcoming and supportive of the assessment effort.

The assessment team consists of a group of six to ten individuals who conduct the assessment. Typically, half or more of the assessment team is made up of individuals who are from the assessed site. Each assessment team member is responsible for reviewing documentation during the assessment, asking questions during interview sessions, reviewing notes, identifying and classifying significant information obtained during interviews and document reviews, and identifying additional information required. Team members are responsible for coming to a consensus on the assessment findings and ratings. The assessment team is led by an experienced individual authorized by the SEI as Lead Assessor. During the assessment planning phase, the team leader is responsible for explaining to the sponsor the impact of assessment scope and constraints on assessment Goals, providing cost/schedule estimates, setting expectations, and obtaining a commitment to proceed.

The assessment participants are managers, project leaders and practitioners from the assessed site who attend the interview sessions to which they are assigned and during which they provide information to the assessment team. Assessment participants also attend the final findings presentation. It is also recommended that as many people as possible in the organization attend the assessment opening meeting and final findings presentation in order to create the momentum for change necessary for improving the process.

1.3.3 Findings and Recommendations of a Process Assessment

Since process improvement information of Oerlikon Aerospace is confidential, the author cannot use it. But, for a better understanding of the size of the effort required to improve processes in a large organization, the author will describe a fictitious organization, called Acme Inc, which operates in a business sector similar to Oerlikon Aerospace.

The main finding and recommendation were of the assessment of Acme Inc was that it was at CMM level 1 (as illustrated in Table A-6), and that there was a foundation for achieving CMM level 2.

Maturity level	Key Process Area	Total number of goals	Number of satisfied goals	KPA rating
2	Requirements Management	2	0	Not Satisfied
2	Software Project Planning	3	1	Not Satisfied
2	Software Project Tracking and Oversight	3	1	Not Satisfied
2	Software Subcontract Management	4	2	Not Satisfied
2	Software Quality Assurance	4	2	Not Satisfied
2	Software Configuration Management	4	2	Not Satisfied
3	Organizational Process Focus	3	0	Not satisfied
3	Organizational Process Definition	2	0	Not satisfied
3	Training Program	3	0	Not satisfied
3	Integrated Software Management	2	0	Not satisfied
3	Software Product Engineering	2	0	Not Satisfied
3	Intergroup Coordination	3	0	Not satisfied
3	Peer Reviews	2	0	Not Satisfied

Table A-6 Results of the Acme Inc CMM Process Assessment

The team recommended that Acme Inc established an immediate goal to achieve software maturity level 2, the repeatable level (see Table A-7).

1. Requirements Management
   - Develop policies, procedures, standards and practices dealing with requirements definition and management.
   - Develop the appropriate engineering skills to manage requirements efficiently.
2. Planning Process
   - Implement formal project size, cost and schedule estimation procedures.
   - Formalize the policies and procedures that Acme Inc uses to spell out the way commitments are made for the organization.
3. Project Tracking and Oversight
   - Define and implement a project tracking policy and supporting procedures to collect and analyze actual project size, cost and schedule data.
4. Subcontract Management
   - Define and implement a process for selecting subcontractors.
   - Define and implement a process for involving, in a consistent manner, the software disciplines in the management of subcontractors.
5. Software Quality Assurance (SQA)
   - Define minimum corporate standards for involvement that are applied to the entire product-related software development process.
6. Software Configuration Management
   - Establish working groups to streamline the process for generating and maintaining documentation.
   - Define and implement the process for analyzing the impact of software changes.
7. Training on Software Engineering and Personal Development
   - Determine the training needs of the organization and develop a comprehensive training program.
8. Training on Company Resources
   - Develop user's manuals and provide training on tools and processes used at Acme Inc.
9. Software Engineering Manual
   - Implement a Software Engineering Manual in which existing procedures have been

rationalized. • Identify and develop policies, procedures, practices and standards to be incorporated in the Software Engineering Manual to define how software development and maintenance will be accomplished at Acme Inc.

Table A-7 Recommendations resulting from the process assessment of Acme Inc

The report described each finding and consequences in details. As an example, for the requirements management process area, the high level findings were:

- An inconsistent process is used to flow down requirements from system to software.
- Design and coding often occur before software requirements have been defined.
- Software requirements are inconsistently defined.

Then, an explanation of the finding was presented as follow: the requirements definition constitutes the first step in a development project. The entire project depends on the quality of the requirements and the manner in which they have been flowed down. If erroneous requirements are flowed down through the development cycle (system requirements, software requirements, system design, detailed design, code, unit test, system integration and maintenance), the cost of corrective actions increases as the time of discovering the errors approaches the end of the cycle.

The assessment team heard that Acme Inc engineers are not following a standard process to define the requirements and flow them down through the development cycle. The team also heard that the process should ensure that relevant documentation activities are performed along the entire development cycle in order to avoid over-reliance on software personnel. Such a process should also ensure the completeness, the clarity and the accuracy of the requirements by having a level of detail matching the level of the document in which a given requirement is defined. This means that a software requirement defined in the system performance specification should be verifiable at the system level, a software requirement defined in the system design specification should be verifiable at the system integration level, and a software requirement defined in the detailed design specification should be verifiable at the code or module level.

The design and coding activities are often carried out before requirements have been defined and the team heard of cases where requirements were developed after the code had been written. The manner in which the process is followed is very dependent on the project manager and the project engineers. They must be made aware of the importance of the process which must be taken into account during project planning.

Finally, a description of the consequences was presented as follow: The first step in developing a quality product is to start a project with consistently defined requirements. Consequences of an inconsistent or undefined process for flowing down or defining software requirements in a system are requirements that are incomplete, unclear, incorrect, unverifiable and/or missing the appropriate level of detail.

First, since project planning is based on the requirements definition, the plan may not be meaningful if the system is not well defined. This means that the cost may not be correctly evaluated and the delivery dates may not be realistic. In addition, the final product may not meet the customer needs and may be based on unverifiable requirements leading to uncertainties in its quality. The company is then faced with image problems resulting from a dissatisfied customer. Loss of control on the development budget may also ensue because of changing priorities to react to development problems caused by incomplete, unclear or incorrect requirements. The problem will be compounded if the project has been subcontracted to another company since the prime contractor does not have full and constant visibility in the activities carried out by the subcontractor.

Secondly, the software development may be performed by skipping some important steps in the development cycle (e.g. design and coding activities performed before requirements have been defined). There is a risk that the resulting product will contain patches, making it difficult and costly to maintain and forcing Acme Inc to over-rely on the engineering personnel who developed it.

As a result of an inconsistent process to flow down the requirements from system to software, it is difficult to prove that all requirements have been satisfied. Acme Inc may then lose time and money by wasting efforts to demonstrate requirements compliance to the customer or by arguing with the customer over unverifiable requirements.

Another consequence of this finding is the difficulty for SQA of verifying that the software development process is being followed.

Finally, because of the lack of consistent process in the requirements management, the development is reactive to events instead of being pro-active. Too often, developers must then work unplanned overtime, leading to frustration because there is no time to perform their job correctly, and Acme Inc is exposing itself to losing key personnel to the benefit of other organizations.

1.3.4 Action Plan

During the Establishment phase (summer-fall 1993) a detailed action plan was prepared by the SEPG following the steps listed in table A-8. During a three-day workshop, assessment findings and recommendations were reviewed and a strategy was developed. It was decided that working groups would be established to define individual processes under the close coordination of the SEPG. For each process, a process owner, i.e. a person responsible for the implementation and improvement of a process, was identified. Working groups of four to six members were staffed with representatives of software engineering, systems and sub-systems engineering, quality assurance and configuration management.

- Review the findings of the assessment
- Introduction to the Capability Maturity Model (CMM)
- Preparation of a plan by the working group
- Brainstorm on strengths and weaknesses of current process
- Understand the current process
- Compare the current process with the CMM
- Describe first level (i.e. overall view) process steps
- Describe second level of the process using the selected notation
- Describe/update, if necessary, third level components:
    - Procedures
    - Users' guides
    - Checklists
- Review process steps
- Select a pilot project
- Brief participants
- Monitor the pilot
- Institutionalize the process:
    - Modify, if necessary, policies and procedures
    - Develop the training material
    - Train all users (technical and non-technical) of the process
    - Monitor the utilization of the process
    - Measure the process and products
    - Improve the process

Table A-8 Process Definition Steps

Each member of the working groups spent up to 8 hours per week on process related activities. In each working group, a member of the SEPG acted as a facilitator. At regular intervals, SEPG members met to resolve issues raised within their groups and passed along lessons learned within their own working groups. For each working group, a mini action plan was prepared by the SEPG (see Table A-9). The action plan listed the following elements: goals of the working group, identification of the owner of the process, identification of the part-time participants, implementation steps, risk issues, timetable, level of effort planned and reference documents.

Acme Inc should implement a central guidance mechanism that defines how system requirements must be defined and flowed down to the software development cycle. In order to ensure the availability of consistent requirements, Acme Inc should proceed with the followings activities and track their accomplishment:

1. Implement engineering guidelines that will include all policies, procedures, standards, and practices required to:

   a. ensure that all necessary disciplines are involved in the requirements management process to review requirements before they are allocated
   b. ensure that the requirements allocated to software (technical and non technical) are documented
   c. ensure that all necessary disciplines are involved in the requirements management process to review changes to requirements allocated to software before the software is modified
   d. ensure that the software project plans, development activities and end product items are kept consistent with changes to the allocated requirements to software
   e. ensure that the requirements can be tracked from the highest level (system requirements) to their implementation (code)

2. Develop the appropriate requirements management expertise by:
   a. identifying the skill set and experience necessary for systematic requirements creation, modification, and implementation
   b. identifying the skill set and experience necessary for systematic requirements management
   c. Assessing the internal engineering skill abilities to prepare a training strategy for upgrading the current skill set to the required level

The following critical factors also need to be addressed:
- The relationship between the system engineering and the software engineering functions.
- Tools required to support the entire requirements management effort

Considering the company's business area, Acme Inc's project managers need general guidance to manage the requirements allocation efficiently and all that is affected by this activity. This is a critical and strategic issue. The company may find in this area an excellent opportunity for a good return on its investment. However, it requires that Acme Inc fully support this initiative by investing the resources required to address the preceding recommendations.

Table A-9 Example of an Action plan about requirements management

It was also necessary to estimate the resources associated to the implementation of the recommendations. As an example, Table A-10 lists the estimation of resources necessary to address the recommendations about requirements management.

Activities	Resources	
1. Review existing process: – Policies – Procedures – Standards – Practices	Individual in charge: Participants:	10 staff-days 10 staff-days
2. Define required process, skills and tools	Individual in charge: Participants:	30 staff-days 35 staff-days
3. Pilot implementation	Individual in charge:	15 staff-days
4. Review	Individual in charge: Participants:	5 staff-days 5 staff-days
5. Implementation: - Training Preparation - Training Delivery	Individual in charge: Participants:	5 staff-days 85 staff-days
Total	**200 staff-days**	

Table A-10 Resource Estimation for Requirements Management

Table A-11 lists the estimation of resources necessary to develop all CMM Level 2 Processes as well as to address to remaining recommendations of the action plan. Over 1,500 staff-days were estimated for this improvement project.

Process to be developed	Estimated Resources (staff-days)
Requirements Management	200 staff-days
Software Project Planning	180 staff-days
Software Project Tracking and Oversight	115 staff-days
Software Subcontract Management	100 staff-days
Software Quality Assurance	90 staff-days
Software Configuration Management	205 staff-days
Software Training Program	50 staff-days +
Training on Company Resources	435 staff-days
Software Engineering Manual (i.e. Consolidate all policies, procedures and standards in a Software Engineering Manual)	30 staff-days
Change Management Plan (i.e. in order to successfully manage the transition from SEI Level 1 to Level 2, a change management plan was developed to support the software improvement action plan)	115 staff-days
Total:	**1,520 staff-days +**

Table A-11 Estimation of resources to develop CMM Level 2 Processes

It is evident, from these numbers that most VSEs do not have the expertise and cannot afford the resources to improve its processes in such a way. As will be presented in a chapter below, the author has proposed to the working group mechanisms to help VSEs adopting and utilizing the future ISO standard.

1.3.5 Selection of a Process Notation

When deciding how to initiate a program of process representation, the selection of a process notation was important since it was the language used to communicate to all engineers their role and responsibilities, tasks, deliverables, etc. Examples of notation techniques were: flowchart, Statecharts, Structured Analysis and Design Technique (SADT), Petri Nets, cross functional diagram (e.g. swimlane), Integration Definition (IDEF), Entry-Task-Validation-eXit (ETVX), etc. Some of the questions we had to answer were (SPC 1992):

- Can the notation represent details and also abstractions? (Scalability)
- Can the notation be used to represent your processes? (Applicability)
- Can the notation be adapted for use in representing different processes? (Flexibility)
- Are process descriptions resulting from this notation easily interpreted? (Readability)
- Can process depictions be easily updated to reflect changes? (Maintainability)
- Can average people become quickly competent without extensive training? (Learnability)
- Is the notation capable of representing a large variety of processes? (Robustness)
- Does the notation yield machine-interpretable process models? (Formality)

Because of its simplicity, the Entry-Task-Validation-eXit (ETVX) (Radice 1985) notation was selected for the description of the processes[78]. Table A-12 presents the details of the ETVX convention used to document the process.

• Step Title	An active verb is used to define the process step title • e.g. Prepare project plan
• Step Description	This section describes the overall purpose of this step
• Inputs	This section identifies what artefacts will be used during the performance of the activity. It is not necessary for all the inputs to exist in order to begin the activity. It is possible for only a subset of inputs to exist during the first iteration through the activity; as more inputs become available, subsequent

[78] This section is adapted from an article originally published in (Laporte et al. 1996)

	passes are made through the activity.
● Entry Criteria	This section identifies prerequisites that must satisfy in order to begin the activity.
● Task Description	This section describes the tasks to be performed during the execution of the activity. Active verb-noun are used to describe activities.
● Exit Criteria	This section states what conditions must meet before the step is considered complete. It is assumed that each activity satisfies the following exit criteria: ● Each artefact generated by the activity has been baselined verified, and approved. ● Each artefact generated by the activity conforms to the organisational policies, standards, and procedures. ● Each artefact generated by the activity has collected process, risk, and quality information.
● Outputs	This section identifies what artefacts will be generated as a direct result of performing the activity.
● Measures	This section describes the measures to be collected during the execution of the step. The measures are used to monitor progress, to aid in estimating the effort required to complete the step, and to improve the process.
● Risks	This section lists the risks elements of this activity
● References	This section lists the documents that can be used to support this step

Table A-12 ETVX Process Notation

Figure A-5 illustrates the graphical ETVX notation. To help define the processes, the working groups also used extensively a document produced by the SEI (Olson 1994), that describes each Key Process Area (KPA) of the Capability Maturity Model for Software (CMM) (Paulk et al. 1993) using the ETVX notation.

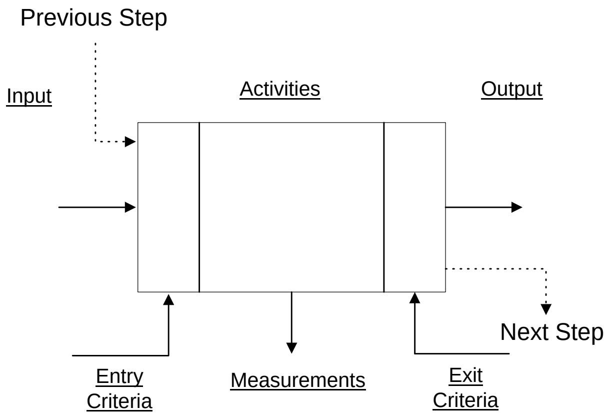


Figure A-5 Graphical representation of the ETVX Notation

During the Acting phase, initiated in winter 1994, working groups started their activities. Working groups were kicked of in one to two months intervals. This way, problems inherent to the dynamics of teams were solved, and lessons learned were captured before starting another group. Once the processes were defined, pilot projects were identified for a trial period. Each process is described at three levels of details: the top-level view is a black box approach describing the major steps required to satisfy the goals of the KPAs. A second level of details describes each black-box with the following information: the objective of the activities to be performed; inputs required to perform the activities; a list of activities; outputs produced; entry and exit criteria controlling the initiation and completion of each process step, measurements (e.g. size, effort, quality), and persons responsible for performing and supporting each process step. At the third level of details, methods are described in process guides (e.g. size estimation, risk assessment). Each person who has to use the processes receives his own copy of the software engineering guidebook which contains: processes, methods and guides. Each person is also trained on the utilization of the processes, methods and guides.

The following processes were developed, tested in pilot projects and implemented: software development, software maintenance, software project planning and tracking, software quality assurance, software configuration management, software subcontractor management, documentation management and document inspection (Gilb et al. 1993).

To illustrate the work performed, the planning and tracking process is described. At the higher level of details, there are three phases (see Figure A-6): the planning activities during a proposal phase, the project planning phase after contract award, and the project tracking phase. The proposal phase either takes the original vision of a potential product or transforms it into a business case or, for a contractual development; the requirements of the request for proposal are analyzed: size, cost and schedule estimates are performed, and a risk analysis is done. For both cases the main outcome of this phase is a go no-go decision. Since, during the contract negotiation phase, it is possible that some requirements (i.e. schedule, software requirements) have been modified, the planning phase after contract award is a required to finalize the plans prepared during the proposal phase. During the third phase, project data are collected, analyzed and adjustments to the initial plans are made.

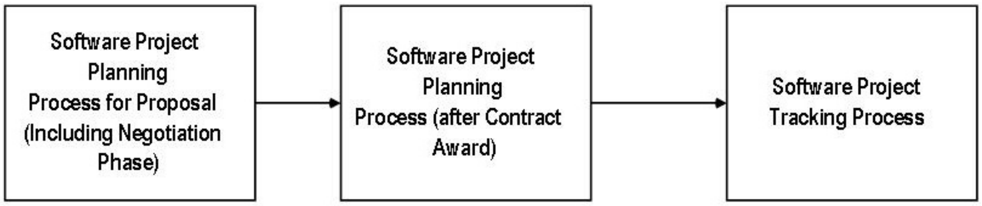
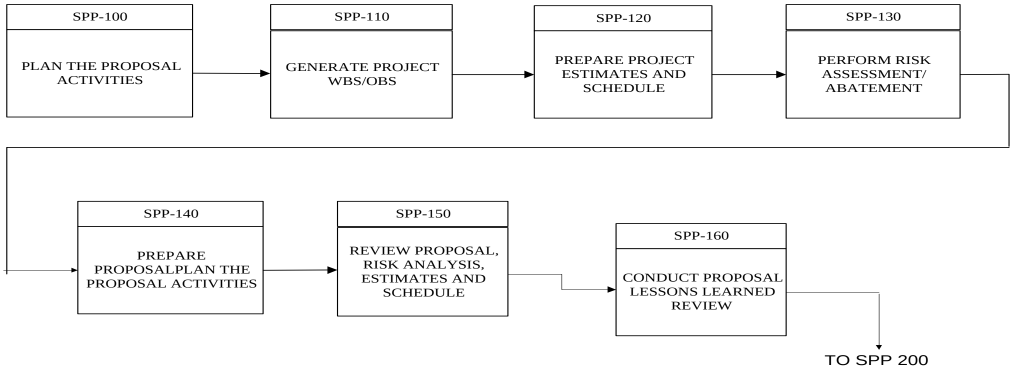

Software Project Planning Process for Proposal (Including Negotiation Phase)
Software Project Planning Process (after Contract Award)
Software Project Tracking Process

Figure A-6 Three Phases of the Project Planning and Tracking Process

The second level of details of the planning and tracking activities during the proposal phase is illustrated in Figure A-7**.** As shown, each step of the process is numbered; also, each step is defined with a verb and a noun. The steps could be used as building blocks and could be linked together according to the needs of the project. It is the responsibility of the project manager to tailor the building blocks. Even though the steps are illustrated as a linear set of steps, feedback to previous steps are allowed. Feedback loops have not been illustrated in order not to clutter the diagrams.


SPP-100
PLAN THE PROPOSAL ACTIVITIES
SPP-110
GENERATE PROJECT WBS/OBS
SPP-120
PREPARE PROJECT ESTIMATES AND SCHEDULE
SPP-130
PERFORM RISK ASSESSMENT/ ABATEMENT
SPP-140
PREPARE PROPOSALPLAN THE PROPOSAL ACTIVITIES
SPP-150
REVIEW PROPOSAL, RISK ANALYSIS, ESTIMATES AND SCHEDULE
SPP-160
CONDUCT PROPOSAL LESSONS LEARNED REVIEW
TO SPP 200

WBS = Work Breakdown Structure

OBS = Organization Breakdown Structure

Figure A-7 Software Planning Process for Proposal

Figure A-8 illustrates the third level of details. This figure shows the ETVX diagram of step SPP-120 as illustrated in the previous figure. Since the diagram cannot contain all the information for a particular step, diagrams are complemented by a textual representation where all elements of the steps are listed. In the process engineering guidebook, each step is illustrated using two notations: the ETVX diagram and the textual description. In the guidebook binder, the diagrams are on the left side and the textual information is on the right side, i.e. facing the ETVX diagrams.

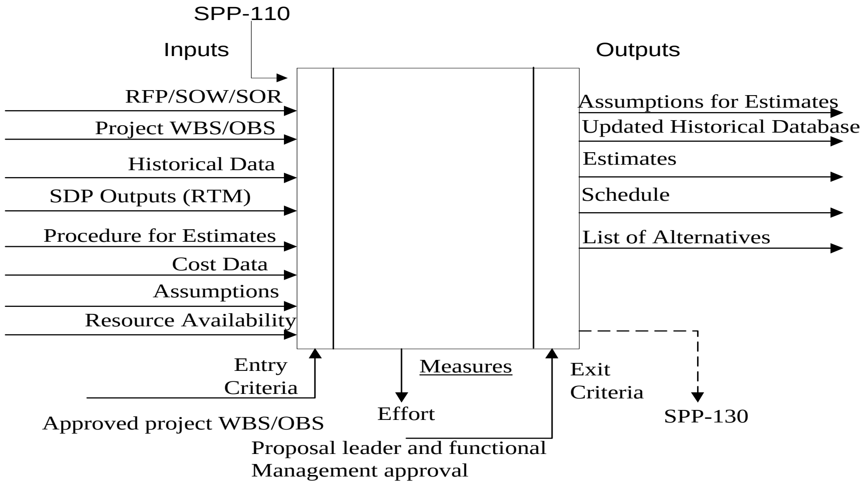


Figure A-8 ETVX Diagram of Step SPP-120

A reverse engineering process had also being defined (Laporte et al. 1998a). It drawn on the experiences based on the process developed under the STARS program (Software Technology for Adaptable, Reliable Systems) (STARS 1995). The reverse engineering process has three major steps: first, a define project step which will include 1) define objectives, 2) identify baseline, 3) define re-engineering project plan; a second major step to reverse engineer the software system, and a third major step to "forward" engineer the software.

In order to constantly improve the process, all users have been invited to propose corrections, modifications or improvements to the process. A process improvement form has been distributed to all users of the process. The SEPG collects, analyzes and proposes improvements to appropriate process owners. Once the modification to the process is completed, a new version is distributed to all users.

Audits were also performed on all projects. The objective of the audits was not to "fix the people" but to bring to the surface barriers to the institutionalization of the processes. The focus on the process rather than on the people is critical for company-wide acceptance of the new process. Each project team was interviewed separately and composite results of the audits were presented to management and project teams. A questionnaire was used to probe projects. The questionnaire used scoring guidelines developed by Motorola (Daskalantonakis 1994). Motorola has developed a ten-level scoring scale which allows a finer evaluation of the institutionalization of each key process area. The scoring guidelines measure the attainment of the following three elements: first the approach, i.e. criteria that show the organization's commitment to and management's support for a practice; second the deployment, i.e. the breadth and consistency of practice implementation; and third the results, i.e. the breadth and consistency of positive results over time. With such a scale, it is easier to measure the progress made by each team from one audit to another. After each audit, a mini action plan was developed to address the findings and implement corrective actions.

Another feature was built in the process in order to capture the lessons learned. The software planning and tracking process had been defined such that it is the first process to be initiated in any project and also the last process to be called at the completion of a project. During the planning phase, the project has to estimate the effort required to conduct lessons learned reviews. During the tracking phase, lessons learned reviews are performed in each project. In order to make sure that the lessons are learned by the organization, each lesson is analyzed in order to identify if a process step could be improved (Basili 1994). If this is the case, modifications to the process, methods or guides are made before the project is allowed to exit from the last step of the tracking process.

As the processes are being used in current projects, artifacts are collected and stored in a process asset library (PAL). A PAL is an organized, indexed, searchable, repository of process assets that is easily accessible by everyone who needs processes and process-related assets such as deployment packages, forms, checklists, templates, examples or other process support materials. Initially, the PAL contains mostly paper documents. As the organization moved toward an environment where each practitioner had access electronically to documents, the PAL contained electronic copies of documents produced. The PAL librarian had read-write privileges while practitioners had only read privileges. The librarian also performed configuration management functions on the artefacts of the PAL. Table A-13 lists the artefacts that were stored in the PAL as projects were producing documents.

• Software Engineering Policy • Process Descriptions • Forms and Templates • Examples of Documents Produced • Business Case Examples • Proposal examples • Software Development Plans (SDP) • Tailored Processes • Tailoring Guidelines • Process Definition Process • Lessons Learned	Software Version History List of Process Owners Process Improvement Suggestions Training Material Quality Assurance Reports (e.g. reports from audits) Quality Data (e.g. results from inspections) List of Software Tools under configuration Historical Data (e.g. project estimates) Software Methods Documentation Charter of Software Engineering Process Group

Table A-13 Content of the Process Asset Library

Finally, during the Leveraging phase, lessons learned from projects and processes are collected, analyzed and implemented. These lessons will be used to prepare the next improvement cycle which is planned to start in the fall of 1996 by a re-assessment of the software engineering process (i.e. a CMM-Based Appraisal for Internal Process Improvement, CBA IPI) by certified SEI assessor from the Applied Software Engineering Centre (ASEC).

Table 2.K list the software processes developed as well as the size, in number of pages, of each process. Acme Inc had been ISO 9001 certified since 1993. In 1997, the organization had been assessed as CMM ({Paulk et al. 1993) level 2 by independent assessors certified by the Software Engineering Institute. In addition to satisfying level 2 goals, the organization also met 8 of the 17 level 3 goals.

Table A-14 lists the main components of the 300-page Software Engineering Process of Acme Inc. As mentioned before, most VSEs cannot afford the resources to develop such a process.

Title of Section of the Software Engineering Process	Size of Process (Number of pages)
Policy	13
Project Planning, Tracking and Oversight	98
Development	48
Documentation Management	34 + Style guide (25pages)
Quality Assurance	14
Configuration Management	28 + CM Procedures
Maintenance	52
Subcontract Management	38
Document Inspection	70

Table A-14 Size of the Software Engineering Process of Acme Inc

Development of a Systems Engineering Process[79]

Although the organization had in use ISO-9001 compliant procedures describing the work that systems engineers have to perform, it was decided that a systems engineering process had to be defined in order to integrate, seamlessly, disciplines associated with systems engineering. In 1995, Acme Inc conducted an internal assessment of its systems engineering practices using the Systems Engineering Capability Maturity Model (SE-CMM) (Bate et al. 1995) and the SE-CMM Appraisal Method (SAM). The objective was to help identify priorities for improvement within the 18 process areas of the SE-CMM as illustrated in Figure A-9.

[79] This section is adapted from an article originally published in (Laporte et al. 1998a)

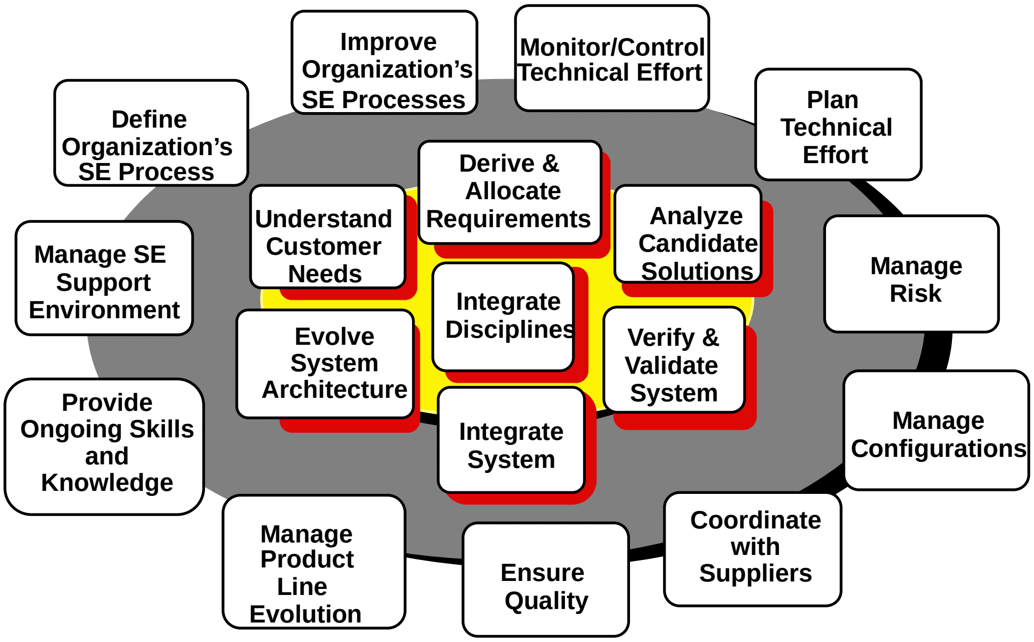

Improve Organization's SE Processes
Monitor/Control Technical Effort
Define Organization's SE Process
Plan Technical Effort
Derive & Allocate Requirements
Understand Customer Needs
Analyze Candidate Solutions
Manage SE Support Environment
Integrate Disciplines
Manage Risk
Evolve System Architecture
Verify & Validate System
Provide Ongoing Skills and Knowledge
Integrate System
Manage Configurations
Coordinate with Suppliers
Manage Product Line Evolution
Ensure Quality

Figure A-9 SE-CMM Process Areas (Bate et al. 1995)

Three systems engineers and two management staffs answered the SAM questionnaire. Results from the questionnaire were compiled and a maturity level for each process area was computed. After analysis of the results management decided to put a higher priority on the engineering process areas as defined in the SE-CMM. Managers reviewed the current literature and a decision was made to use, as frameworks, the SE-CMM and the Generic Systems Engineering Process (GSEP) developed by the Software Productivity Consortium (SPC 1995). The GSEP has been developed to incorporate most of the practices of the SE-CMM. A working group, composed of 11 systems engineers, software engineers and a representative from quality assurance, was established to define and facilitate the implementation of a systems engineering process. Another objective of the working group is to integrate the current software engineering processes to the systems engineering process. This objective is part of the progress that has to be made to work at SEI level 3 of the CMM for software.

The GSEP document describes, using the IDEF notation (USAF 1981), management and technical activities and also the artifacts produced by each activity. The major management activities (i.e. major steps and major sub-steps), as illustrated in table A-15 and figure A-10, are: understand context, analyze risk, plan increment development, track increment development and develop system.

Major Steps	Sub-Steps
110 Understand Context	111 Define Approach 112 Estimate of Situation 113 Review Context
120 Analyze Risk	121 Perform Risk Analysis 122 Review Risk Analysis 123 Plan Risk Aversion 124 Commit to Strategy
130 Plan Increment Development	131 Execute Risk Aversion 132 Review Development Alternatives 133 Plan Increment Development 134 Commit to Plan
140 Track Increment Development	141 Monitor and Review Increment Development 142 Update Increment Plan 143 Review Technical Product
150 Perform Increment Closure	151 Baseline System Definition 152 Assess Increment Closure 153 Update External System Plan 154 Commit to Proceed

Table A-1-15 Management Activities of the Systems Engineering Process

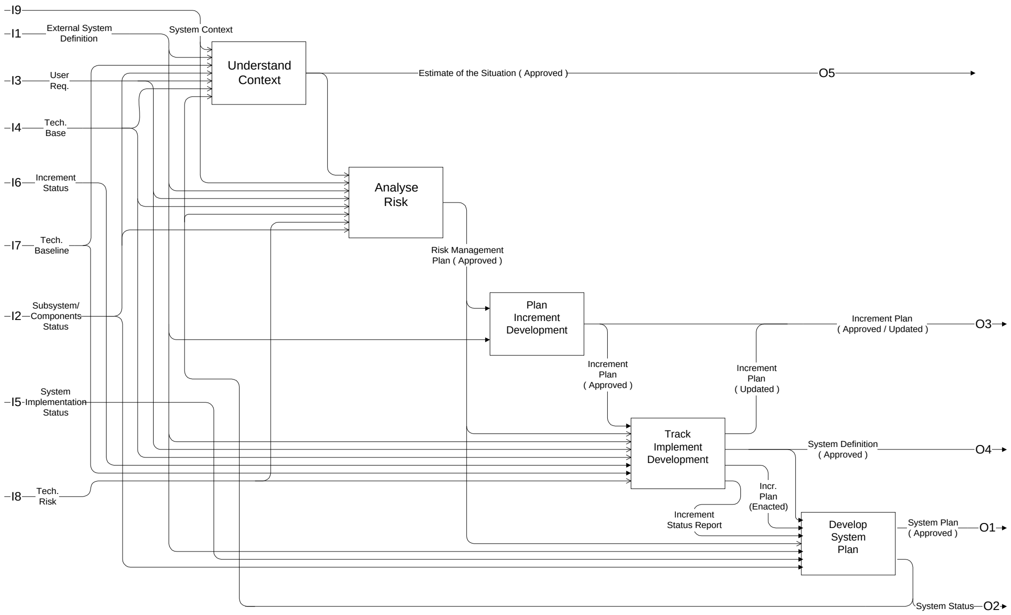

I9
I1
External System Definition
System Context
I3
User Req.
I4
Tech. Base
I6
Increment Status
I7
Tech. Baseline
I2
Subsystem/ Components Status
I5
System Implementation Status
I8
Tech. Risk
Understand Context
Estimate of the Situation (Approved)
O5
Analyse Risk
Risk Management Plan (Approved)
Plan Increment Development
Increment Plan (Approved / Updated)
O3
Increment Plan (Approved)
Increment Plan (Updated)
Track Implement Development
System Definition (Approved)
O4
Incr. Plan (Enacted)
Increment Status Report
Develop System Plan
System Plan (Approved)
O1
System Status
O2

Figure A-10 Management Activities of the Systems Engineering Process

The major technical activities, as illustrated in table A-16 and figure A-11, are: analyze needs, define requirements, define functional architecture, synthesize allocated architecture, evaluate alternatives, validate and verify solution and control technical baseline. Each major activity is broken down in a certain number of smaller activities which are described, individually using the ETVX notation. The strategy was to define a beta version of the technical activities, then of the management activities, use the beta version on pilot projects and make corrections to both management and technical activities of the process before full deployment.

Major Steps	Sub-Steps
210 Analyze Needs	211 Determine Stakeholders 212 Define Problem Domain 213 Develop Informal Functionality
220 Define Requirements	221 Determine Behavioral Requirements 222 Determine Performance Requirements 223 Map Behavior to Performance 224 Refine Requirements
230 Define Functional Architecture	231 Partition Requirements into Functions 232 Define Lower Level Functions 233 Define Functional Interfaces
240 Synthesize Allocated Architecture	241 Allocate Functions to Alternative Solutions 242 Define Physical Parameters 243 Define Physical Interfaces 244 Integrate Design 245 Refine Physical Architecture
250 Evaluate Alternatives	251 Assess System 252 Perform Sensitivity Analysis 253 Allocate Performance to Technical Parameters 254 Assess Technical Risks and Problems 255 Identify and Perform Trade-off 256 Select best System Solution
260 Verify and Validate Work Products	261 Define V&V Procedures 262 Verify System 263 Validate System
270 Release System Definition	271 Control Technical Decision Data 272 Control System Configuration

Table A-16 Technical Activities of the Systems Engineering Process

In addition to defining the process, each member of the working group had a secondary duty, as each step of the beta version of the process is defined, members of the working group were tasked to collect the following information: updates to process descriptions, monitor compliance with the SE-CMM, monitor the interfaces with the software engineering processes, identify process and product measurements, identify roles and responsibilities, define glossary, identify methods, best practices, artefacts, CASE tools, lifecycle representations, project templates, estimation guidelines, course material, training resources, lessons-learned, and establish the systems engineering process asset library.

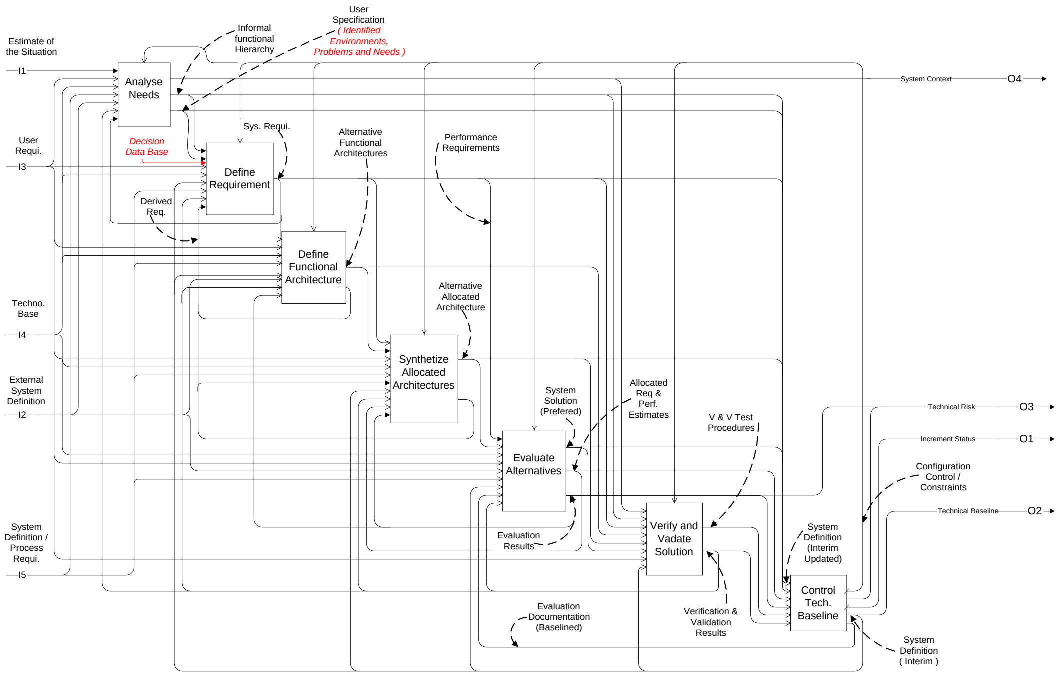

Estimate of the Situation
I1
User Requi.
I3
Techno. Base
I4
External System Definition
I2
System Definition / Process Requi.
I5
Informal functional Hierarchy
User Specification
(Identified Environments, Problems and Needs)
Analyse Needs
Decision Data Base
Sys. Requi.
Define Requirement
Derived Req.
Alternative Functional Architectures
Performance Requirements
Define Functional Architecture
Alternative Allocated Architecture
Synthetize Allocated Architectures
System Solution (Prefered)
Allocated Req & Perf. Estimates
V & V Test Procedures
Evaluate Alternatives
Evaluation Results
Verify and Vadate Solution
Evaluation Documentation (Baselined)
Verification & Validation Results
System Definition (Interim Updated)
Control Tech. Baseline
System Definition (Interim)
Configuration Control / Constraints
System Context
O4
Technical Risk
O3
Increment Status
O1
Technical Baseline
O2

Figure A-11 Technical Activities of the Systems Engineering Process

Table A-17 lists the main components of the 200-page System Engineering Process. As mentioned before, most VSEs cannot afford the resources to develop such a process. The author will present in a next chapter a project, under the sponsorship of the International Council on Systems Engineering (INCOSE), to develop for VSEs a process using an approach similar to the approach used by the ISO working group 24.

Title of Section of the System Engineering Process	Size of Section (Number of pages)
Introduction	26
Roles and Responsibilities	16
Manage Development Effort	60
Define System Increment	75
Glossary and abbreviations	15

Table A-17 Size of the Systems Engineering Process

1.3.6 The Management of Change

Since the management of change is a key element of a successful process improvement program, a series of actions were planned in order to facilitate the development, the implementation and the adoption of the processes, methods and tools (Laporte et al. 1993). As an example, to build the sponsorship level, the president attended a one-day executive seminar on process improvement at the SEI, two directors attended a three-day seminar discussing the CMM, process, process assessment and improvement. Also, one member of the SEPG attended two courses at the SEI: managing technological change and consulting skills. Briefing sessions were held and articles were written in each company's newsletter to explain the why, what and how of process assessment and improvement and describing the progress made. Finally, surveys were conducted in order to assess the organization's readiness to such a change in practices. The surveys identified strengths of the organization and potential barriers to the planned improvement program.

Also, in order to get support and commitment for the future implementation of processes, working groups were staffed with representatives from many departments: software engineering, systems engineering, sub-systems engineering, quality assurance, contract management, and configuration management. Each working group was managed like a project. It had a charter, a budget and a schedule. A process owner, i.e. a manager responsible for the definition, implementation and improvement of each process was part of a working group. A member of the SEPG acted as a facilitator in each working group. Therefore, the process owner would focus on the content of a specific software process while the facilitator would focus on the process of developing a specific software engineering process.

In order to facilitate the conduct of working group activities, a certain number of meeting guidelines (Siddall 1996) were proposed, by the facilitators, to the members of working groups during the kick-of meeting of their group. The proposed guidelines are listed in Figure A-12. It was decided that consensus decision making was the preferred decision-making option. We defined consensus, according with the definition found in the Team Handbook (Scholtes 1988): consensus is not unanimity, consensus is based on the assumption that solutions are more likely to succeed if all of the key participants are "comfortable enough" with the outcome to move forward. During meetings we use "thumb voting" procedure (Popick 1996) to make decision by consensus. Thumb voting allows the following three alternatives: first, if the proposition is favoured, the thumb is up; second, if someone can live with the decision, the thumb is to the side; third, if someone cannot live with the decision the thumb is down. In the later case, the members of the working group have to take time to understand the issues at stake and propose an alternative that everyone can live with.

- One conversation at a time.
- In the meeting or out, but not both (i.e. participants should make a commitment to participate to the meeting for the full duration).
- 100-kilometer rule (i.e. no interruptions, e.g. telephone messages, allowed unless urgent)
- How decision will be made (e.g. by consensus, majority or minority rule, autocracy, unanimity).
- Once a decision is made, participants support it inside and outside the meeting.
- Be as open as possible.
- We listen, with respect, to others and do not interrupt them.
- Silence is consent.
- Few recreational stories.
- Differences are respected.
- Avoid blaming individuals.
- We come prepared to meetings.
- We publish minutes and action items at each meeting.

Figure A-12 Proposed meeting guidelines

EVALUATION OF THE PEOPLE DIMENSION

Since the management of change is a key element of a successful process improvement program, a series of actions was planned to facilitate the development, implementation and adoption of the processes, methods and tools (Laporte et al. 1993, Laporte et al. 1998, Laporte et al. 1999). As illustrated in Figure A-13, this structured approach was used to evaluate and manage the human and cultural elements of an organization in order to manage the changes necessary to meet its business objectives.

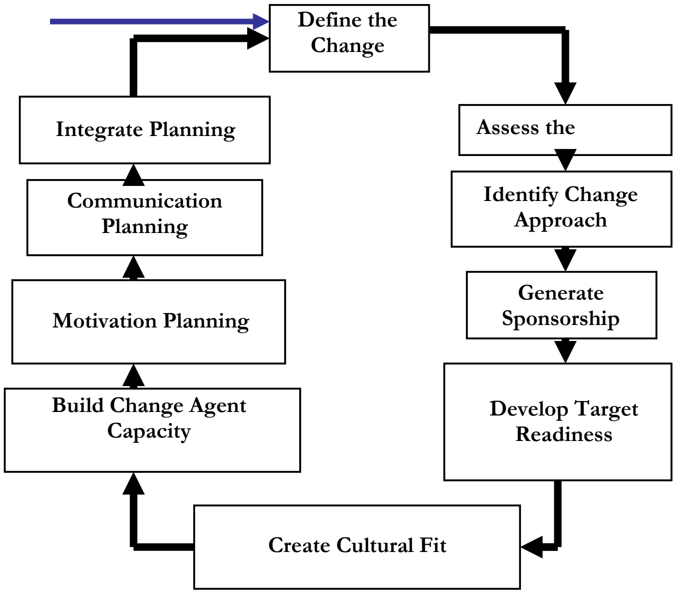


Figure A-13 Change management approach (adapted from IMA Inc.)

The assessments performed allowed the organization to better identify potential barriers and perform mitigating actions to increase the likelihood of the success of a change project.

Finally, on a periodic basis, members of the working groups had to evaluate the effectiveness of their group. A survey (Alexander 1991) was distributed at the end of a meeting, members were asked to complete the survey and to send it to the facilitator of their group. The survey addressed the following issues: goals and objectives, utilization of resources, trust and conflict resolution, leadership, control and procedures, interpersonal communications, problem solving, experimentation and creativity. At the following meeting, issues that were surfaced by members were discussed in order to generate suggestions for improvement.

1.3.7 Supporting Processes

To support the software and systems engineering processes, the following set of supporting processes have also been developed: Document Inspection Process, Documentation Management Process, Project Management Process and a Lessons Learned Process. Each process is briefly described in the following paragraphs.

DOCUMENT INSPECTION PROCESS

A Document Inspection Process (DIP) was developed using the method described by Gilb (Gilb et al. 1993). After conducting a few inspections, it became evident that software engineers had a higher level of confidence of software documents that were inspected. Although, the organization had a document management process imposing a structured review by peers, software engineers requested that documents that were used as an input to the software process had to go through the inspection process. Since the inspection process had been, on purpose, documented to be a generic document inspection process, other engineering disciplines, such as systems engineers, will be able to use the inspection process as is. Dedicated checklists will be developed for other engineering work products. As a mean to foster a smooth deployment of the inspection process in systems engineering, representatives of software engineering, systems engineering and quality assurance were trained as inspection leaders.

DOCUMENTATION MANAGEMENT PROCESS

Initially, the Document Management Process (DMP) had been developed to support software engineering activities. At that time the process ownership belonged to the software engineering manager. A few years after its initial deployment, the scope of the process had been enlarged such that it would cover all engineering documentation activities. The process ownership was transferred to the director of engineering services. Recently, with the introduction of new management processes, it was decided that the

DMP would be applied to all documents produced by the organization. The process ownership had been transferred to the manager of data and configuration management.

Also, with the successful deployment of the document inspection process (DIP), the DMP was modified in order to accommodate two types of reviews: a peer review and the inspection process. The first type of review is where an originator circulates a document to his colleagues. They individually review the document and forward their comments to the originator. No metrics are collected as part of this process while metrics, such as the number of errors, are collected as part of the document inspection process.

PROJECT MANAGEMENT PROCESS

It was felt that it would also benefit from a standardized project management process. A mandate was given to a working group, in 1996, to develop and implement a Project Management Process (PMP). The author of this thesis was nominated to coordinate and facilitate the development of the process. The working group selected the Guide to the Project Management Body of Knowledge (PMBOK), developed by the Project Management Institute (PMI 2008), as the framework for the organizational process. Because it oversees all technical processes that are required to generate plans and products, the PMP was not designed as a stand-alone process. To make sure that all stakeholders of a typical project were represented, the working group was composed of project managers and representatives from engineering disciplines, and representatives from quality assurance, manufacturing, configuration management and logistic support. The process owner of the PMP was the vice-president - Project Management.

The four phases of the project management process, illustrated in Figure A-14, are:

- **Initiate Project** - As the project begins, a Project Development Folder (PDF) is created and maintained as a central repository that contains all the received, generated, and sent products for the project. It is assumed that when a product is needed to perform an activity, it is found in the PDF. Project begins with a statement of needs, in any form (e.g. statement of work, request for proposal, request for information). The outcome of this phase is a decision as to whether or not the organization will propose competitive solutions to the expressed needs. In the affirmative, a proposal plan is generated describing how the proposal team will accomplish the required planning effort. Depending on the proposal estimated duration, one or more iterations of the Execute and Control phase must be performed.
- **Plan Project** - The Plan Project phase establishes reasonable plans for performing

and managing the project. Project planning involves developing estimates for the work to be performed, establishing the necessary commitments, and defining the plan to perform the work. When the work is authorised to proceed, this phase may have to be re-iterated if a long period of time has elapsed before approval or due to contract negotiations.

- **Execute and Control Project** - The Execute and Control Project phase consists of a series of recurring activities that begin as soon as the project work is authorized. Its main purpose is to provide adequate visibility into actual project progress so that the PM can take effective actions when the project's performance deviates significantly from the plans and adaptation to changes is required.
- **Close Project** - Throughout the PMP, many reasons may cause the project to terminate. In such case, the Close Project phase is always performed. That phase consists mainly in closing all open issues with the customer as well as suppliers, performing a lessons learned session, closing work orders, and archiving the Project Development Folder.

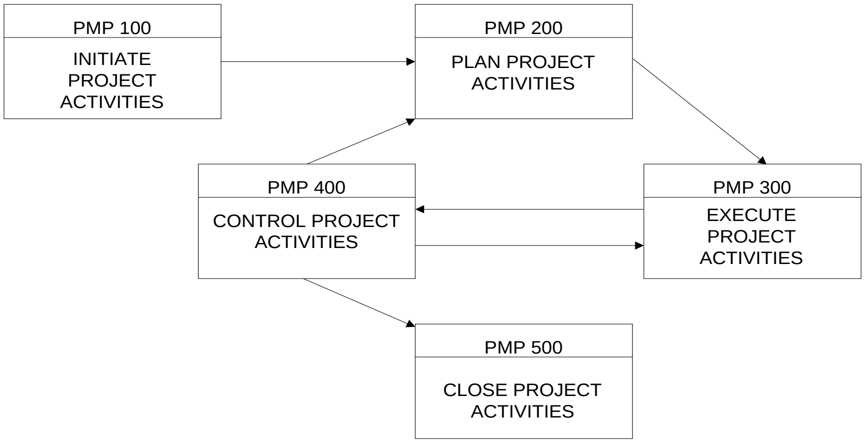


Figure A-14 Project Management Process

Table A-18 lists the main components of the 150-page Project Management Process. As mentioned before, most VSEs cannot afford the resources to develop such a process.

Title of Section of the Project Management Process	Size of Process (Number of pages)
Introduction, Policy, Code of Ethics	21
Initiate Project	4
Plan Project	40
Execute and Control Project	26
Close Project	2
Guides	70

Table A-18 Size of the Project Management Process

LESSONS LEARNED PROCESS

A lessons' learned process had been originally developed for the software engineering process. In order to make sure that lessons learned from a project would be captured at the organizational level, a feature borrowed from the NASA (Basili 1994) was added to the process. Once a lesson learned session is completed, the process owner is mandated to make appropriate modifications to the process, procedures or methods. This process is now used by other organizational processes such as the systems engineering process and the project management process.

INTEGRATION OF PROCESSES

Defining engineering and supporting processes was quite a challenge. These processes also had to be integrated for the conduct of engineering projects.

INTEGRATION OF SOFTWARE PROCESS TO SYSTEMS PROCESS

Throughout the development of the systems engineering process, the working group kept on the agenda the integration between the systems process and the software process. It was decided to adopt, as a framework for the integration the Integrated Systems and Software Process (ISSEP) from the SPC (SPC 1996). Since many problems, when developing complex computer-based systems, are discovered at integration time, the solution is to use a process that will decompose the systems in parts that can be developed independently and easily integrated together at the system level. It was also noted that, because of digitization of electro-mechanical systems, the apparent space of software was increasing on projects from nominally 30% in the mid-70's. Software has now reached 60% to 70% of the non-recurring activities in system development.

Furthermore, the expansion of Integrated CASE technologies, which crossed departmental barriers, through common process framework, reinforced the desire of the organization to integrate both software and systems engineering process and to focus the organization into an integrated project team approach. In other words, software and systems engineering are beginning together at the inception of a project. Therefore, the working group selected ISSEP as the reference model.

The ISSEP model defines a decomposition strategy for system development as well as a set of management and technical activities and interfaces between processes. ISSEP describes activities at three levels: the system level, the configuration item (CI) level and the component level. It is at the component level that software and hardware are developed. Figure A-15 illustrates the integration between processes. The manage development effort and define system increment boxes are described in detail in the systems engineering process, the develop software configuration item box is essentially the software engineering process, while the develop hardware configuration item box, i.e. the design engineering process, represents a process presently being documented.

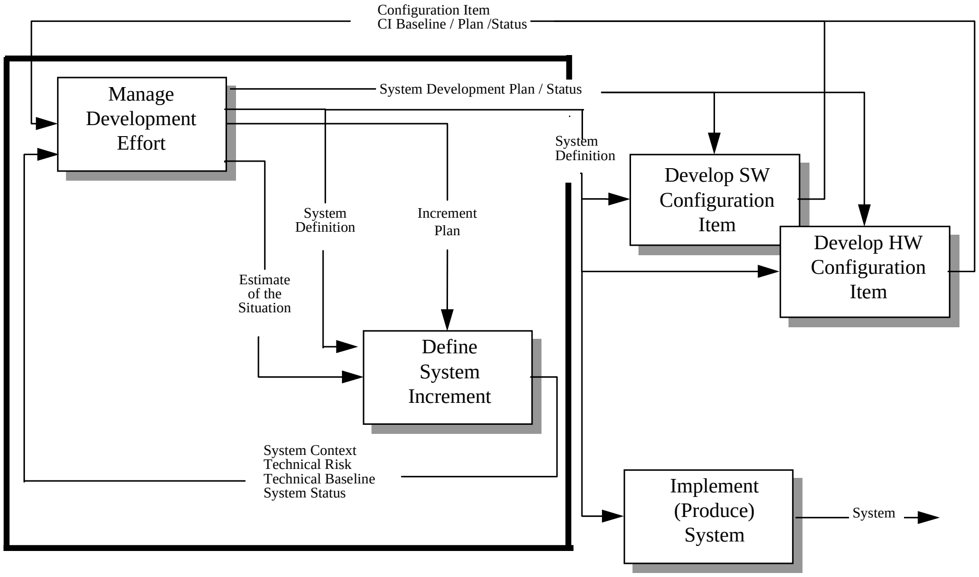

Configuration Item
CI Baseline / Plan /Status
Manage Development Effort
System Development Plan / Status
System Definition
Develop SW Configuration Item
Develop HW Configuration Item
System Definition
Increment Plan
Estimate of the Situation
Define System Increment
System Context
Technical Risk
Technical Baseline
System Status
Implement (Produce) System
System

Figure A-15 Integration of Engineering Processes

QUALITY ASSURANCE AND CONFIGURATION MANAGEMENT PROCESSES

Other supporting processes that were developed during the software initiative are: configuration management and quality assurance. The scope of both processes was enlarged to be used by other organizational processes. Also, ownership of these processes had been transferred from software engineering to configuration management and quality assurance managers.

1.3.8 Integration of Activities Mandated by more than One Process

As we integrated the processes, two types of issues surfaced. First, some activities had been documented in the three processes because they were developed, sequentially, in a bottom up approach. As an example, risk management activities have been defined, back in 1994, in the software engineering process because it was felt that risk management was important. Then, risk management activities were defined in the systems engineering process and the project management process. In order to prevent duplicating these activities, the issue was resolved by assigning the primary responsibility for risk management to the PMP process, and inserting in the engineering processes dedicated risk activities.

Another type of issue surfaced because some activities were mandated by all frameworks used. As an example, the management of subcontractors is mandated by the SW-CMM, the SE-CMM and the PMP from the Project Management Institute. Since the subcontractor management process had already been defined during the software initiative, the scope of this process had been broadened to include hardware acquisitions. Even though, this process is part of the PMP process, it was decided that the process ownership would remain with the department responsible for acquisitions. The relationship between the PMP process and this process is viewed as a client-server relationship where a project manager issues his requirements for a particular acquisition, the requirements are then transferred to the subcontractor process, and once the goods are delivered to the satisfaction of the project, the subcontractor process is stopped.

1.3.9 Integration of Plans

Since engineering and management processes are mandating the development of plans (e.g. software development plan, systems engineering master plan, project plan), a tailoring of the processes was needed in order to develop plans that would use the same vocabulary and would not contradict other plans for the same project. As an example, the word prototype had a different meaning for software engineers, systems engineers and project managers. In order to resolve similar issues, a common vocabulary was developed in collaboration of stakeholders.

1.3.10 Integration of Design Activities between Engineering Processes

Another integration issue in the design process is to determine when to stop the design using the systems engineering process in order to hand over the design information to the software engineering process. As the processes are integrated, other issues are also considered. As an example, the selection of engineering methods and tools is important in order to facilitate the transfer of information between processes.

Integration of People

One of the biggest issues in integrating process is the ¨people¨ issue. Great pain must be taken to avoid inter-departmental conflicts as the organization is transitioning from a matrix organization based on technical disciplines to a matrix organization based on processes. If not managed properly, this transition can create stress and resistance to change.

It was decided, as a mean to foster collaboration and understanding, to train together all users of a process, i.e. users from different functional departments. As an example, two-day training sessions on the Systems Engineering Process were held with representatives from system engineering, software engineering, sub-system engineering, quality assurance and configuration management. Students, assembled in mini groups of six and originating from different departments, were guided through the process, by a facilitator.

As another mean to foster integration of processes, Integrated Project Teams were established. In such a team, people have to work together from the beginning to the end of a project. They have to develop project plans and therefore they have to tailor and integrate different processes.

1.3.11 Team Start-up Process

A team, as defined by (Katzenbach 1994), is a small number of people with complementary skills who are committed to a common purpose, performance goals, and approach for which they hold themselves mutually accountable. Becoming committed to a purpose and performance goals and a common approach and hold themselves mutually accountable does not happen by itself. Neither is a team "created" by a mandate from senior management. In our organization, teams are developed using a start-up process [9]. The process describes a series of practices, grouped in stages that guide a leader in developing a team. The typical stages of team building are: member orientation, trust building, goal and role clarification, commitment, implementation, high performance and renewal.

1.3.12 Performance Management Process and Reward System

Establishing new processes, practices, tools and teamwork require additional changes to the organization. As an example, when individuals are requested to work in teams, the old performance evaluation system based solely on individual contributions has to be transformed to take into account the new team structure. It becomes almost impossible, in a team structure, to evaluate a member of a team since his performance is also greatly influenced by the performance of his teammates. Similarly, the reward system of the organization needs to be adapted to the new team environment since, in a team everybody win or everybody loose. The reward system should not reward the lone ranger or the lone hero anymore. Reward system must evaluate the performance as a team and reward individuals as team members.

1.3.13 Organizational Process Coordination

In early 1997, it was felt that the implementation of these processes would need organizational coordination and direction to manage the transition to a product line organization and to manage processes across product line. It was decided to establish a steering committee called the Process Action and Coordination Team (PACT). The PACT was composed of three vice-presidents and the coordinator for process performance improvement. The functions of the PACT were:

- Establish time-to-market, quality, costs and product performance objectives to be supported by organizational processes
- Set priority in accordance with company vision and yearly objectives
- Liaise with executive committee
- Establish consensus among different groups
- Provide support for process performance improvement:
- Review results of assessments and audits
- Charter technical area working groups
- Budget for resources for process groups
- Monitor process performance

1.3.14 Lessons Learned from the Process Improvement Activities[80]

These years of process improvement had enabled Oerlikon Aerospace to learn certain lessons likely to be used by other organizations in the future. The author used many of these lessons when working at the development of the international standards, technical reports and deployment packages that will be presented in the next chapters.

LESSON 1: SET REALISTIC EXPECTATIONS FOR SENIOR MANAGEMENT

Appropriate expectations must be set prior to embarking on a process improvement journey. The trap, especially for CMM level 1 organizations, consists in communicating to

[80] This section is adapted from an article originally published in (Laporte et al. 1998b)

management the idea that the initiative will be easy, fast and inexpensive, has to be avoided at all costs.

A typical scenario looks like this: senior management realises the benefit that attaining a maturity level can represent for his organization's competitiveness. As second step, a project manager or an external consultant states, in order not to upset the top management, that this objective is easily attainable. As a third step, top management gives managers the mandate to attain this objective in a very short lapse of time. During the assessment, the managers face a string of countless findings. Findings that had been known by developers for a long time, but remained ignored due to the mode of management that consists in dealing continuously with the problems created (i.e. fighting fires), in a clumsy way at times, by managers. Top management, that had already announced its objective to its peers from other organizations, realises suddenly that this objective will take a lot more time and resources than what had been estimated. At that time, three reactions are possible. Top management may accept the findings and confirm that it will continue to support the objectives announced. It may announce discreetly that it will be lowering its objectives. Finally, it can deny everything and renounce to implement an action plan to correct the deficiencies highlighted by the assessment. This decision could have a destructive effect on developers, since they know for a fact that the deficiencies they had been deploring for a long time are now known by everybody and will remain ignored for a long time.

The lesson to be remembered is to prepare a first action plan -- some sort of a brief appraisal of the situation status -- preferably by someone who is not involved in the sector targeted and to assess the time and resources necessary to assessing, writing and implementing the action plan. One has to remember top management does not like bad surprises. Moreover, it is better not to proceed to an assessment if it is not intended to deal with the findings. As a matter of fact, once the problems are identified and publicised within the organization, if the management decides not to act, it then sends a very bad message to practitioners.

LESSON 2: SECURE MANAGEMENT SUPPORT

A second lesson for CMM level 1 organizations consists in realizing that most of the assessment findings target the deficiencies of project management processes. It is necessary to create an environment where the management is ready to invest in the implementation of processes rather than blame its managers; in other words "where the management is ready to fix the process, not the people". This is one of the reasons why it is necessary to also keep informed senior management representatives so that they can show understanding and full commitment when these findings are publicized within the organization.

Beside senior management buy-in, it is essential that middle management and first line managers become strong supporters of the process improvement program. The strongest signal sent by managers is their day-to-day activities, because "what a manager does talks louder than what a manager says". The developers must receive very clear signals announcing that the changes advertised will be implemented and that they themselves will have to adopt new practices.

LESSON 3: IDENTIFY MANAGEMENT NEEDS, EXPECTATIONS AND UNDERSTANDING OF THE PROBLEM

The involvement of process owners or managers is largely related to their understanding of the current situation (i.e. strengths and weaknesses). Once convinced that the current situation is undesirable, they will provide the leadership (e.g. direction and momentum) to implement solutions. They can also keep a working group focused on solving the right problems. Since, it is very easy, after a few meetings, for members of a working group to start solving what they perceived to be the problems.

LESSON 4: ESTABLISH A SOFTWARE PROCESS ENGINEERING GROUP

The Software Capability Maturity Model suggests the formation of a formal Software Engineering Process Group (SEPG) for any organization heading toward level 3. Even for a level 1 organization, it would be better that a small number of persons became active in process activities a few months before the on-site assessment. The SEPG should take this time to familiarize itself with the Capability Maturity Model and associated process improvement methods and tools. Ideally, in a large organization, there should be one full-time person on the SEPG while the other members could be assigned on a part-time basis. Beside their technical competencies, the members of the SEPG should be selected based on their enthusiasm for improvement and the respect they have within the organization.

LESSON 5: START IMPROVEMENT ACTIVITIES SOON AFTER AN ASSESSMENT

With regards to the development of the action plan, the organization should capitalize on the momentum gained during the assessment period. The organization does not have to wait for a completed action plan to start process improvement activities. Some improvement activities can begin soon after the completion of the on-site assessment. The implementation of certain improvements is an important motivation factor for all members of the organization.

During the assessment, it is recommended to collect both quantitative and qualitative data (i.e. indicators) which will be used later to measure the progression realized. One could obtain data on slipped budgets and schedules, or measure the degree of satisfaction of the customers regarding product quality level. Since senior management will have made investments, it is very appropriate to be able to demonstrate that these investments have been profitable.

LESSON 6: TRAIN ALL USERS OF THE PROCESSES METHODS AND TOOLS

Once the processes are defined, it is essential to train all users. Otherwise, all related documents will end up getting dusty on shelves. It is illusory to think that developers will study, by themselves, new processes in addition to their workload. Training sessions also serve as a message that the organization is going ahead and will require that its developers use these practices. During the training sessions, it is necessary to indicate that, however everybody's good will, errors are bound to happen while using new practices. This will help reducing developers' level of anxiety in their using these new practices. It would be a good thing that a resource-person be available to help developers (i.e. hotline) when the latter face obstacles while implementing new practices.

LESSON 7: MANAGE THE HUMAN DIMENSION OF THE PROCESS IMPROVEMENT EFFORT

The author also wishes to make the reader aware of the importance of the human dimension in a process improvement program. The people responsible for these changes are often extremely talented software engineering practitioners, however not too well equipped in change management skills. The reason for this is simple: during their training, they focused on the technical dimension and not on the human aspect. However, the major difficulty in the whole improvement program is precisely the human dimension. Also while preparing the technical part of the action plan, the change management elements have to be planned (Laporte 1994). This implies, among other things, a knowledge of (1) the organization's history with regards to any similar efforts, successful or not, made formerly; (2) the company's culture; (3) the motivation factors; (4) the degree of emergency perceived and communicated by (a) the management, (b) the organization's vision, and (c) the management's real support. The author is convinced that the success or the failure of an improvement program has more to do with managing the human aspect than managing the technical aspect.

LESSON 8: PROCESS IMPROVEMENT REQUIRES ADDITIONAL "PEOPLE SKILLS"

In an organization that truly wants to make substantial gain in productivity and quality, a major cultural shift will have to be managed. Such a cultural shift requires a special set of "people" skills. The profile of the ideal software process facilitator is someone with a major in social work and a minor in software engineering. The implementation of processes implies that both management and employees will have to change their behaviours. With the implementation of processes, management will need to change from a "command and control" mode to a more "hands-off" or participative mode. As an example, if the organization truly wants to improve its processes, a prime source of ideas should come from those who are working, on a daily basis, with the processes, i.e. the employees. This implies that management will need to encourage and listen to new ideas. This also implies that the decision making process may have to change from the autocratic style, e.g. "do what you are told" to a participative style, e.g. "let us talk about this idea". Such a change requires support and coaching from someone outside the functional authority of the manager who has to change its behaviour. Similarly, employees' behaviour should change from being the technical "heroes" that can solve any bug, from being passive and unheard in management issues to work in teams and generate and listen to others' ideas to make improvement.

Also, the first few months of the introduction of a new process, a new practice or a new tool, both management and employees must acknowledge that mistakes will be made. Unless a clear signal has been sent by management and a "safety net" has been deployed to recognize this situation, employees will "hide" their mistakes. The result is that not only the organization will not learn from them but other employees will make the same mistakes again. As an example, the main objective of the inspection process is to detect and correct errors as soon as possible in the software process. Management has to accept that in order to increase the errors detection rate, results from individual inspections will not be made public, only composite results from many inspections (e.g. at least ten inspections) will be made public. When this rule is accepted by management, employees will feel safe to identify mistakes in front of their peers instead of hiding them. The added benefit to correcting errors early in the process is that those who participated to an inspection will learn how to avoid these errors in their own work.

Facilitating such a change in behaviours requires skills that are not taught in technical courses. It is highly recommended that the people responsible for facilitating change be given appropriate training. The author recommends a course given by the SEI, the title of which is "Managing Technological Change". For lack of such a course, the authors recommend to read two books that may facilitate the management of change: the first one (Block 1981) gives advises to anybody acting as internal consultant; the second one

(Bridges 1991) gives the steps to be followed for writing and implementing a change management plan.

LESSON 9: SELECT CAREFULLY PILOT PROJECTS

It is also very important to carefully select pilot projects and participants to the pilots since these projects will foster adoption of new practices throughout the organization. Also, first time users of a new process will make mistakes. It is therefore mandatory to properly coach the participants and provide them with a "safety net". If participants sense that mistakes will be used to learn and make improvements to the process instead of "pointing fingers", the level of anxiety will be reduced and they will bring forward suggestions instead of "hiding" mistakes.

Managing the human dimension of the process engineering initiative is the component, which not only fosters the adoption of change but also creates an environment where changes could be introduced at an increasingly greater rate. Members of the engineering organization now realize that managing the "soft stuff" is as important as managing the "hard stuff".

The utilization of models such as the CMM for software and systems engineering is slowly changing the culture of the organization from the "Not Invented Here" to the "Not Reinvented Here" mindset. Although a few practitioners still believed that they are different most see the benefits of reusing someone else's work. They also see that the organization encourage them to look for solutions instead of constantly reinventing the wheel. Engineers are now intensively using the Internet to look for practices developed by other organizations and adapting these practices to the environment of the organization. Practitioners attend conferences sponsored by organizations such as the SEI and INCOSE to identify best practices for their utilization in day-to-day activities.

LESSON 10: CONDUCT PROCESS AUDITS

Process audits should be conducted on a regular basis for two main reasons: first, to ensure that practitioners are using the process, and second, to discover errors, omissions, or misunderstandings in the application of the process. Process audits help to assess the degree of utilization and understanding of the practitioners. As an example, a documentation management process was released and practitioners were asked to produce and update documents using this new process. It is widely known that engineers are not very prone at documenting their work. Oerlikon Aerospace launched an audit and measured process compliance. As expected, see Table A-19, results were not exhilarating. The engineering manager kindly reminded engineers, in writing, to use the process. He also informed them that a second audit would be performed in the future. As shown in Table 4, the results of the second audits are substantially better than the first audit. Also, the auditor gathered feedback from engineers; such information will be used by the process owner to improve the process.

Activity	**Results from First Audit**	**Results from Second Audit**
Comments made by reviewers	38 %	78%
Approval matrix completed	24%	67%
Effort log completed	18%	33%
Review checklist completed	5%	44%
Configuration management checklist completed	5%	27%
Distribution list completed	38%	39%
Document formally approved	100%	100%

Table A-19 Results of Audits performed on the Documentation Management Process

LESSON 11: CONDUCT TEAM EFFECTIVENESS SURVEYS

Oerlikon Aerospace has found that such a tool promotes open discussion with members of a group besides improving the performance. Usually, people are not very prone to raise “soft” issues. Oerlikon Aerospace has found that this tool provides the facilitators with information that help him probe delicate issues. As an example, if the majority of a working group reports that interpersonal communications are weak, the facilitator can probe the members and invite them to propose solutions. After a few meetings, the results of a new survey will show if the solutions really helped the team improve their communications.

LESSON 12: START A PROCESS INITIATIVE FROM THE TOP LEVEL PROCESS

Oerlikon Aerospace process improvement initiative was a bottom up exercise, i.e. first software process was developed, then systems engineering, then project management where each additional process “sits” on top of the other. Historically, this was the selected strategy because, in 1992, only the software CMM was available, then came the systems CMM and after, the Body of Knowledge in project management. If an organization had to start, today, a process initiative, it would be easier and more efficient to start from the top by developing the project management process, then the systems and finally the software process. It would also be possible to develop these processes in parallel once the requirements for the top level process are well stabilized.

LESSON 13: GET SUPPORT FROM ORGANIZATIONAL CHANGE EXPERTS

As mentioned above, surveys were conducted in order to “measure” issues such as culture, implementation history, and team effectiveness. Once the surveys were compiled, we had some indications of organizational strengths and weaknesses. The difficult part was to decide what to do next. As an example, one issue on the survey is risk taking. If the survey showed that people resent taking risks, one possible cause for such behaviour was that people did not want to be blamed for an error. Having found this cause was not too helpful since we would have to find the cause for this behaviour, and so on. It would have been very helpful to have access to someone with expertise in organizational change. This would have saved a lot of long discussion and wrong answers.

LESSON 14: TIE PROCESS IMPROVEMENT ACTIVITIES TO BUSINESS OBJECTIVES

It was observed that software and systems engineering process improvement really picked-up momentum when a common focal point was created between management, engineers and customers. Understanding that the real benefit of process improvement lies in improving product quality, reducing time-to-market and cost. Consequently, improving the ability of the organization to better compete. Additionally, a multi-year Process Improvement Plan (PIP) is a very important tool to illustrate the links between business objectives, project requirements and process development or improvement. Essentially the PIP illustrates that the engineering of processes is not a paper exercise but an important infrastructure for the successful accomplishment of projects. Being a multi-year plan, the PIP also shows to practitioners the long-term commitment of management to process improvement activities.

LESSON 15: ADOPT A COMMON VOCABULARY

To succeed in any project endeavour, a common vocabulary is a basic requirement. As Oerlikon Aerospace developed the processes, Oerlikon Aerospace realized that different players had different meaning for the same word, or the same word had different meanings, and some words were not well known to some individuals. Oerlikon Aerospace therefore mandated one team member as the “glossary keeper”. His role was to collect a vocabulary, propose some “clean-up” in the terminology, and to gradually build a common glossary for all processes that have to be disseminated across the process users.

1.4 Applications of the Engineering and Management Processes

In the next paragraphs, the author describes the application of the software and systems engineering processes developed at Oerlikon Aerospace. A first project used the systems engineering process to re-engineer two sub-systems of the ADATS system; a second project used the systems engineering process to evaluate the risks in the redesign of the ADATS operator console; a third project used most of the engineering, management and support processes to develop a capability to support the Canadian government to verify the compliance to Year 2000 also known as the ‘Y2K bug’ project.

1.4.1 The Application of a Systems Engineering Process to the Re-Engineering of a Defense System

The systems engineering process had been applied to the re-engineering of two major components of the ADATS air defense missile system (Laporte et al. 1998)

The re-engineering initiative targeted the two subsystems: the launcher control electronics and the operator consoles. The launcher control subsystem is composed of a main data processor which coordinates the operation of the sensors and the launch and guidance of the missiles, a missile tracker processor, a target tracker processor, and a servo control processor. The operator consoles consist in a radar console, which allows controlling the radar and communication subsystems, and an electro-optical console, which allows controlling optical sensors and missile launcher.

1.4.2 Implementing Risk Management as part of a Systems Engineering Process[81]

A new systems engineering process involving the management of risks had been deployed and used in the redesign of the ADATS missile system operator console. The risk management activities, listed in Table A-20, were found very useful to plan activities and collect technical and managerial information more formally in the course of the projects. It also helped to manage and improve the dynamic human dimension of the development project. (Laporte et al. 1999)

120 Analyze Risk	121 Perform Risk Analysis	Identify potential risks Identify potential loss and consequences Analyze risks dependencies Identify risks probability of occurrence Prioritize risks Identify risk aversion strategies for each risk
	122 Review Risk Analysis	Review risk analysis Identify risks to be part of the Risk Management Plan (RMP)
	123 Plan Risk Aversion	Define a risk monitoring approach Estimate risk aversion strategy cost and schedule Recommend risk aversion strategies
	124 Commit to Strategy	Obtain stakeholders commitment

Table A-20 Risk Activities of the Systems Engineering Process

During step 120-Analyze Risks, risks are analyzed, risk mitigation strategies are developed, and stakeholders commitment is made on mitigation strategy (see steps 121 and 122). In both the launcher control and console projects, a Risk Management Plan (RMP) was developed. The RMP had two sections: "Risk descriptions and Impacts" and "Mitigation Strategies and Associated Risks". The first section identified and categorized risks: project

[81] This section is adapted from an article originally published in (Laporte et al. 1999)

risks such as budget overrun, schedule delays mostly due to lack of dedicated resources, and technical risks such as the lack of experienced personnel in using a new SEP and a new CASE tool. Also, since the two projects were performed concurrently, it was necessary to closely monitor integration, validation and verification activities, and interfaces definition with the rest of the missile system. Finally specific risks like availability of COTS (Commercial Of the Shelf Software) hardware, mastering of new technologies, development of new custom circuit card assembly (CCA), and development of a new communication bus. The risks impacts were represented by a weighted probability of occurrence and consequence index. This risk matrix was stored in a database and was continuously updated during the two projects.

The second section of the RMP associated and developed a mitigation strategy for each risk (see step 123). The strategy included description, monitoring approach, schedule and cost impact, and required resources of the mitigation activity. In some cases, the same mitigation strategy addressed several risks. Mitigation strategies included activities such as pilot projects, engineering models and mock-ups, additional analyses, and components and subsystem modeling. Specific participant training was also planned in some areas. Finally, a formal review with stakeholders helped to identify other risks, gather mitigation suggestions, and obtain final commitment (see step 124).

1.4.3 A Software Factory for the Canadian Government Year 2000 Conversion Program[82]

A software Factory has been established, in Canada, to re-engineer major systems of the Canadian Government mainly for real-time embedded systems to verify the compliance to Year 2000 also called the Y2K bug. (Laporte et al. 1998)

To solve the Y2K problem for any level (system of systems, system, sub-system, or components of a system), a global system engineering approach was necessary, supported by other processes when applicable. Rigorous planning must be done at all levels and all work products must be submitted through disciplined formal peer reviews (inspection) in order to detect and address the maximum number of issues.

The Factory adopted an integrated discipline approach for management and execution of all software-systems engineering activities of the company. To that effect, various inter-related processes had been developed and documented to serve as a basis for all projects requiring engineering participation. These processes were compliant to: ISO 9001, Project Management Institute, Software Engineering Institute (SEI) Software Engineering

[82] This section is adapted from an article originally published in (Laporte et al. 1998c)

Capability Maturity Model, and Electronic Industry Association Standard 731 on Systems Engineering Capability Model. The set of processes consisted of the following:

- Systems Engineering Process;
- Software Maintenance Process;
- Software Development Process;
- Software Re-engineering Process;
- Software Quality Assurance Process;
- Software Configuration Management Process;
- Documentation Management Process;
- Document Inspection Process;
- Project Management Process;
- Procurement Management Process; and
- Staffing Management Process.

The Year 2000 Process was divided into five major phases: Phase 0: Awareness; Phase 1: Assessment; Phase 2 Conversion; Phase 3 Test & Validation; and Phase 4: Deployment. In addition to the activities performed in those phases, there were activities related to Program/Project Management and Engineering that were performed continuously as background activities. Figure A-16 gives an overview of the Y2K conversion process.

The Awareness phase main objective was to distribute information about the Y2K problem and its potential solutions and insure awareness/commitment from stakeholders. The Assessment phase, phase 1, evaluated the scope of the Y2K problem for a specific system and recommends solutions through a business case. The Conversion Phase, phase 2, executed the solution proposed as the result of the Assessment phase up to the point where the solution were tested and validated, which was done in the Test & Validation phase, i.e. phase 3. At last, phase 4 ensured that the tested and validated system was correctly deployed in its final environment, including all the required training to the users, when applicable.

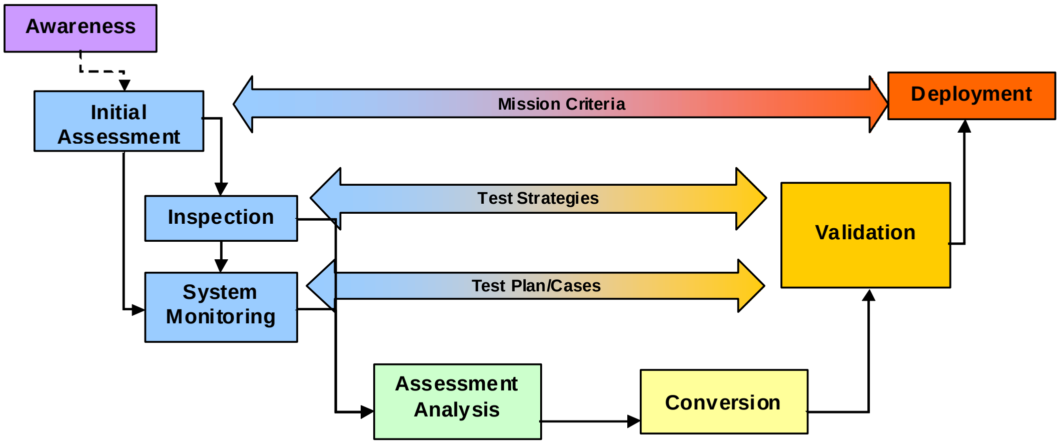


Figure A-16 Overview of Year 2000 Conversion Process

Conclusion from the Process Development and Improvement Activities at Oerlikon Aerospace:

We have shown that the development and deployment of engineering and management processes entailed technical and management competencies. Five elements were necessary for a successful implementation of process changes. First, management had to set a direction and process objectives linked to business objectives. Without a clear direction, confusion would have misled people from reaching the desired change. Second, people were trained to perform new tasks. Without the proper training, anxiety among the organization's staff is likely to slow down the occurrence of change. Third, incentives were provided to facilitate the adoption of changes. Forth, resources were estimated and provided. Otherwise, frustration could have put an end to the organization's willingness to change. Fifth, an action plan was developed and implemented to avoid false starts. Also, these years of process improvement activities had demonstrated that the constant attention to the "people issues" was critical to the success of technological changes.

1.5 Other Significant Professional Activities in this Period

After leaving Oerlikon Aerospace, the author kept busy in software process activities but in another domain: the railway industry at Bombardier Transport. The author will not describe the work performed since the activities were quite similar to the process improvement activities performed at Oerlikon Aerospace. He was also invited to teach a graduate course, about software quality assurance (SQA), at the Computer Science Department of Université du Québec à Montréal (UQAM).

1.6 Accomplishments from 2000 to Present Time

In 2000, the author was invited to join the department of electrical engineering of École de technologie supérieure (ÉTS). At that time, ÉTS was already offering a graduate program, in collaboration with UQAM and INRS (Institut National de Recherche Scientifique) télécommunication (another branch of the Université du Québec), in software engineering and was preparing the launch an undergraduate program in software engineering.

In this section, the courses developed and taught by the author are presented. The author will show the changes brought to these courses as he learned more about the challenges and needs of VSEs. Finally, the projects of graduate software engineering students, related to VSEs, are briefly described.

1.6.1 Graduate Courses in Software Engineering

The author was asked to develop and teach a graduate course in software engineering titled 'Case Study'. The course was designed for students to learn about software process improvement. Initially, students studied a real industrial case study. The case study described the findings and recommendations resulting from a formal software process assessment, using the Software CMM (Paulk et al 1993), of the set of software engineering processes of a fictitious organization.

Students were asked to develop, in teams of 3 to 4, based on the findings and recommendations, a software process improvement plan, a communication plan, an installation plan and a description of processes. Each team had to present to the class their documents for discussion.

After the course had been presented a few years, the author introduced a more dynamic teaching strategy by requesting students to perform, in teams of 3 to 4, a software process improvement project in a real organization. The students had to; obtain agreement and commitment from an organization, document a business case, perform a mini-assessment of the organizational software processes, develop an action plan and a communication plan, assess the cultural climate of the organization using special tools, document and deploy a process, as a pilot project, and document a lessons-learned report of their industrial project. The course objectives are listed in Table A-21.

Students had to perform a complete improvement cycle in less than 12 weeks. From the initial 'static' case study course to the more 'dynamic' intervention course, the author had moved from a case study 'in-the-large' to an 'intervention in the small'. The author had gradually switched its teaching strategy from 'studying process-improvement-in-the-large' to 'performing process-improvement-in-the-small. Most case studies were performed in

VSEs, i.e. either in stand alone enterprises, in a small department of a large organization or in a small project within a large organization. In the future, students will be encouraged to use the documents developed by the ISO working group, especially the set of deployment packages (the deployment packages will be explained in a next chapter).

General objectives Objective and concepts of software quality. Quality Factors. Quality assurance and verification and validation standards. Quality assurance process and activities. Quality assurance and verification and validation plan. Methods and tests. Software tools facilitating the implementation of quality assurance, of verification and validation and tests.
Specific Objectives To understand the principles of quality assurance and control. To understand the quality factors. To understand the costs and benefit of quality. To use quality assurance and verification and validation software engineering standards. To understand quality assurance process and activities. To understand the techniques of verification and validation. To prepare and use checklists. To understand the main functions software configuration management. To understand types of software reviews. To participate to the inspection. To understand tests. To take part, as a team, in a project conducted in industry.
Strategy The objectives will be achieved by weekly three-hour lectures. The student will have to supplement his formation by personal work and team project conducted industry.

Table A-21 Case Study Course description

He also taught a graduate course about software quality assurance (SQA) (Table A-22). Similarly to the process improvement course, initially, originally this course was more ‘static’, and later students were asked to perform an intervention in an organization.

General Objectives

The case study course promotes the acquisition of practical software engineering knowledge and the preparation of the software engineering (9-credit) project. The case study use lectures, where the student receives information, and project activities where the student develops a solution of a given problem. The case study develops the skills of the student to solve a problem, to develop its aptitudes for teamwork and to deepen acquired concepts. The case study covers all the facets of a problem.

Specific Objectives

To provide students with concepts, methods and tools to improve the performance (e.g. productivity, development cycle and quality) of software engineering processes.
To use various process improvement models and approaches.
To develop a business case, a communication plan and an improvement plan.
To assess a process.
To measure and analyze organizational issues related to the management of change.
To deploy an improvement plan.
To define and document a software process.
To identify and manage the risks associated with improvement.

Strategy

The objectives will be achieved by weekly three-hour lectures. The student will also complete personal work and a team project conducted in industry.

Table A-22 Graduate Quality Assurance Course description

This course used the draft standards and deployment packages in the winter 2009 academic session. One team used the following three deployment packages (DP) to improve the process in a VSE: requirements analysis, project management and version control. One member of the team was an employee of this VSE. For their project, the students analysed the content of the DPs, they analysed the processes currently used in the VSE, they made adjustments to the DPs and they proposed to the management of the VSE a series of recommendations. This course ended in mid April. In June, I requested the team to provide me with a progress report. Essentially, in June, the set of recommendations had been approved and some of them had already been implemented. The detailed report of this project is attached as annex A to this appendix.

1.6.2 Undergraduate Program in Software Engineering

ÉTS began offering its software engineering undergraduate program in 2001. The aim of the SQA course, which is mandatory in this software engineering curriculum, is to ensure that software engineering students are aware of the importance of SQA, and that they understand and are able to manage its theoretical and practical aspects. This includes knowledge of the key ISO and IEEE standards, as well as how to use SQA tools in practice. The course allows students to apply SQA practices across the whole software life cycle.

OVERVIEW OF THE SOFTWARE ENGINEERING CURRICULUM

The software engineering curriculum is a ten-term program, and includes three four-month mandatory paid internships in industry. Courses are offered during all three four-month terms of the year. Students may opt to complete their internships during the fall, winter or summer terms. Every course includes a weekly three-hour lecture and a weekly two- or three-hour laboratory session where students must complete practical assignments. Laboratories are equipped with modern equipment as well as software. Table A-23 lists the software engineering courses in the curriculum excluding courses such as mathematics, physics, management and social sciences, which are common to all undergraduate engineering programs at the school. The author developed the software quality assurance course (LOG 330) and directed students in the conduct of their capstone projects (LOG 790). The following section describes the software quality assurance course and illustrates, to the reader, the evolution of the course from 'software quality assurance for large organizations' to 'software quality assurance for very small, medium and large organizations'.

Course Label	Course Title
LOG120	Software Design
LOG220	Advanced Object-Oriented Programming
LOG230	Management of Software Development Process
LOG310	Formal and Semi-Formal Languages (optional)
LOG320	Data Structures and Algorithms
LOG330	**Software quality assurance**
LOG340	User Interface Analysis and Design
LOG410	Requirements Analysis and Specification
LOG420	Software Architecture and Design
LOG510	Quality Control and Measurement
LOG520	Systems Security
LOG540	Analysis and Design of Telecommunications Software
LOG550	Design of Real-Time Computer Systems
LOG610	Telecommunication Networks
LOG620	Algorithm Analysis
LOG630	Introduction to Databases
LOG640	Introduction to Parallel Processing
LOG650	Compilation Techniques
LOG660	High-Performance Databases
LOG710	Principles of Operating Systems and Systems Programming
LOG720	Distributed Object-Oriented Architecture
LOG730	Introduction to Distributed Systems
LOG740	Interactive Multimodal Systems
LOG790	**Capstone Project**

Table A-23 List of Software Engineering Courses

UNDERGRADUATE COURSE IN SOFTWARE QUALITY ASSURANCE[83]

Quality is increasingly seen as critical to business success, customer satisfaction and acceptance. Its absence may result in financial loss, dissatisfied users and damage to the environment, and may even result in deaths (Leveson et al. 1993). Software quality assurance (SQA) becomes even more important when we consider all the software development projects that have failed, and the financial losses generated by those failures. As reported by Charette, software specialists spend about 40 to 50 percent of their time on avoidable rework (Charette 2005).

THE GUIDE TO THE SOFTWARE ENGINEERING BODY OF KNOWLEDGE

The objectives of the Guide to the Software Engineering Body of Knowledge (SWEBOK) (Abran et al. 2004) was to characterize the content of the software engineering discipline, to promote a consistent view of software engineering worldwide, to clarify the place, and set the boundary, of software engineering with respect to other disciplines, and to provide a

[83] This section is adapted from an article originally published in (Laporte et al. 2007d)

foundation for curriculum development and individual licensing material. The SWEBOK Guide was a project of the IEEE Computer Society. It is a consensually validated document available free of charge from IEEE[84]. The SWEBOK has also been published as ISO/IEC Technical Report 19759 (ISO 2005a), it is available free of charge from ISO[85]. The author was the Canadian delegate during the last step of the ISO approval process. He also acted as the ISO Working Group 20 co-project editor and prepared proposed dispositions of comments for the working group meeting discussion and approval.

The SWEBOK Guide is subdivided into ten knowledge areas (KAs), the descriptions of which are designed to discriminate among the various important concepts, permitting readers to find their way quickly to subjects of interest. Upon finding such a subject, readers are referred to key papers or book chapters selected because they present the knowledge succinctly. The ten KAs are: Software Requirements, Software Design, Software Construction, Software Testing, Software Maintenance, Software Configuration Management, Software Engineering Management, Software Engineering Process, Software Engineering Tools and Methods, and Software Quality. Each of them is treated in a chapter of the SWEBOK Guide. In the SQA course, we cover the Software Quality KA in depth, and also some elements of Software Configuration Management.

Figure A-17 illustrates the breakdown of the SQA KA into three topics. Each topic is covered in the SQA course. As an example, the ethics topic is covered by a class presentation of the IEEE/ACM Code of Ethics (Gotterbarn et al. 1999), followed by a two-hour practical session where students have to identify clauses of the Code of Ethics that were violated in a Case Study entitled, "The Case of the Killing Robot" (Epstein 1994). The author contributed to this topic by translating the IEEE/ACM Code of Ethics in French and in Thai for his SQA students at the Chiang Mai University. The French translation has been published on the internet site of Professor Gotterbarn[86].

The Code describes eight top-level technical and professional obligations against which peers, the public, and legal bodies can measure a software engineer's ethical behaviour. Each top-level obligation, called a principle, is described in one sentence and supported by a number of clauses which gives examples and details to help in interpretation and implementation of that obligation. Software engineers adopting the Code commit to eight principles of quality and morality. The following are examples: Principle 3 (Product) states that software engineers shall ensure that their products and related modifications meet the highest professional standards possible. This principle is supported by 15 clauses, and

[84] www.swebok.org
[85] http://standards.iso.org/ittf/PubliclyAvailableStandards/c033897_ISO_IEC_TR_19759_2005(E).zip
[86] http://seeri.etsu.edu/Codes/default.shtm

clause 3.10 reads that software engineers shall ensure adequate testing, debugging, and review of software and related documents on which they work.

In class, the author challenged the students by providing them with behaviours of typical students and confronting them with clauses of the Code. As an example, many students have acquired software from Internet without paying the license. They were asked to read and discuss this clause (clause 2.02): not knowingly use software that is obtained or retained either illegally or unethically. Students were also asked if the Code could be applied in different sizes of organization and how they could deploy the code when they do an internship or when they obtain their first job as a software engineer. They systematically answer that the code can be applied in all sizes of organizations.

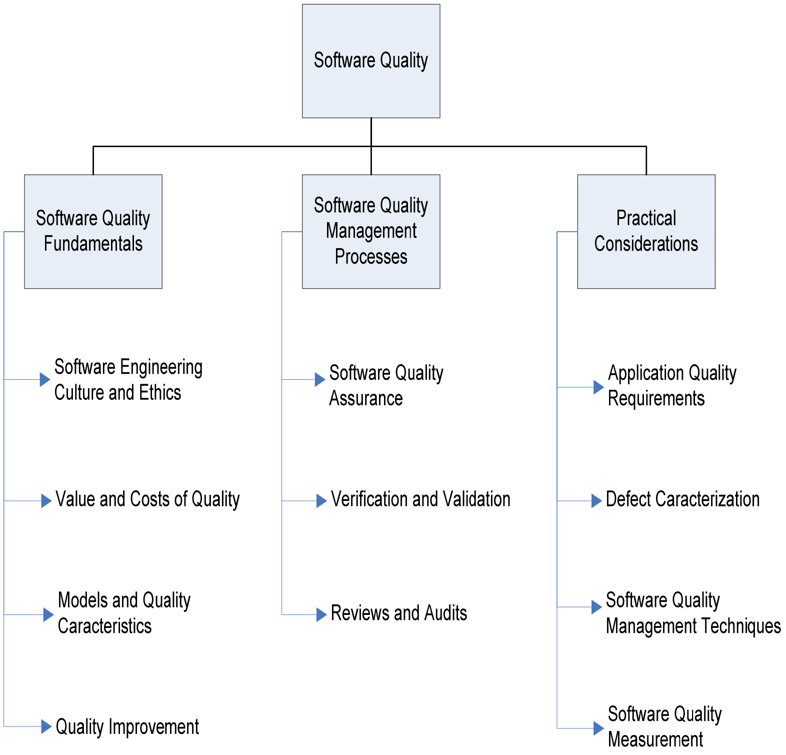

Software Quality
Software Quality Fundamentals
Software Quality Management Processes
Practical Considerations
Software Engineering Culture and Ethics
Value and Costs of Quality
Models and Quality Caracteristics
Quality Improvement
Software Quality Assurance
Verification and Validation
Reviews and Audits
Application Quality Requirements
Defect Caracterization
Software Quality Management Techniques
Software Quality Measurement

Figure A-17 Breakdown of software quality topics (ISO 2005a)

Each lecture topic is supported by industrial examples, international standards clauses and process improvement model practices. To ensure that students grasp the importance of SQA activities, the concept of the cost of quality is stressed throughout the course. When performing SQA activities as part of their term projects, students must make tradeoffs between prevention, appraisal, and conformance and rework costs. They must experience, first-hand that an investment in prevention and appraisal will result in a significant reduction in failure costs (e.g. rework effort).

The laboratory sessions have been designed in such a way that teams of students will apply the SQA theory presented in the lectures to their SQA term projects. Also, to simulate an industrial context where an employee does not usually select his teammates, we create teams by randomly assigning 3 or 4 students to a team.

Since 80% of our graduates will work in very small enterprises (VSEs) and small and medium-sized enterprises (SMEs), it became imperative to expose them to low-cost tools to help in the deployment of SQA practices in organizations with scarce resources. Also, many students were members of student clubs. These clubs have problems similar to those of VSEs: limited budget, scarce resources and high turnover. The author was pleased to learn that a few clubs, like the remote-controlled submarine and the unmanned piloted helicopter clubs, had implemented SQA practices and open-source tools presented in the SQA course.

Students attend twelve 2-hour laboratory sessions during a semester. They also undertake a 10-week project, where they have to apply the SQA concepts presented in the lectures. Table A-24 briefly describes the laboratories that are part of the SQA course. In the future, students of the SQA course will be exposed to the standards, technical reports and deployment packages developed by ISO working group 24. These documents, being written for VSEs, could be used for the term-project conducted in teams of 3 to 4 students.

Topic	Description
1	**Code of Ethics**: 1) Study the IEEE/ACM Code of Ethics and the robot killer case study; 2) Find the violated clauses of the Code; and 3) Determine the responsibilities.
2	**Team project – Draft a project plan and a software requirement specifications document (SRS).** 1) Software quality plan (IEEE standard 730); 2) Take into consideration the customer's local Java programming rules; 3) Develop project plan and estimates by Cost of Quality items; 4) Use the IBM Rational SRS template (functional and nonfunctional requirements (use ISO/IEC 9126 for the nonfunctional requirements); 5) Carry out a walkthrough of the documents produced; and 6) Carry out a traceability analysis with the IBM Rational RequisitePro tool or Excel.
3	**Team project** - **Software Configuration Management (SCM) and Traceability**. 1) Implement the SCM plan using IEEE 730 and IEEE 828; 2) Document the SCM procedure using the Entry-Task-Verification-Exit notation (ETVX) (Rad85); 3) Update the project effort estimation; 4) Read documentation, and configure and test CVS tool for the following roles: system administrator, the individual responsible for configuration management and users; and 5) Carry out a walkthrough of the document produced.
4	**Team project** - **Programming and test**. 1) Program additional features into existing software; 2) Test the software produced; 3) Update information on the IBM RequisitePro/Excel, Bugzilla and CVS tools; and 4) Update the project effort.
5	**Team project** - **Problem/change and defect management and inspection.** 1) Complete section 4.8 of the IEEE 730 standard; 2) Document the change/problem and defect management procedure using the ETVX notation; 3) Read the documentation, configure and test the Bugzilla tool; 4) Print the statistics and management reports from Bugzilla; 5) Carry out a walkthrough; and 6) Update the project effort.
6	**Team project** - **Product Quality Assessment.** 1) Assess source code conformance to customer standards using CheckStyle and software complexity/quality using Logiscope.

7	**Team project - Finalize/update quality assurance plan.** 1) Finalize the plan according to IEEE standard 730; 2) Inspect the plan; 3) Carry out an evaluation of team members (peer evaluation); 4) Carry out a project postmortem; and 5) Use the effort estimation and project tracking data to: Analyze the variations and the costs of quality, explain the variations, explain how, in similar projects, the tasks could be carried out to minimize the variations and the costs of an absence of quality (rework).

Table A-24 Topics of the SQA laboratories

1.6.3 Graduate Projects, at the Master Level about Very Small Entities

This section describes the projects, conducted by graduate students, about Very Small Entities that were supervised or co-supervised by the author. Table A-25 lists the projects performed by graduate students under the supervision of the author or under the co-supervision of the author.

- Bégnoche, Luc, Amélioration des processus de développement logiciel dans les très petites organisations, 2008.
- Brahimi, Ahmed, Implantation de la Norme ISO/CEI 20000 dans la commission scolaire Marguerites Bourgeoys, 2008.
- Berrhouma, Nabil, Development and Deployment of a Cost of Software Quality Measurement System at Bombardier Transportation, 2007.
- Foisy, Marco, Development and Deployment of a Software Test Process for the Wind Mill Research Laboratory of École de technologie supérieure. 2007.
- Corrales, Yaqueline, Evaluation of a Software Architecture Migration Process and its Impact at Technologies Interactives Mediagrif, 2007.
- Sédion, Yacouba, Development and Deployment of a Project Management Office at Vircom Inc, 2007.
- Bluteau, Karine, Development and Analysis of a Survey to Investigate the Utilization of ISO (International Organization for Standardization) Software Engineering Standards by Very Small Software Organizations, 2007.
- Desfossés, Yves, Development of an ISO 9001 Certification Kit for Small Software Organizations of Québec, Project co-directed with Professor Alain April, ÉTS, 2006.
- Abou El Fattah, Mohamed Mounir, Project to Improve the Competitiveness of Small Software Development and Maintenance Companies in Québec. Project co-directed with Professor Jean-Marc Desharnais, ÉTS. 2006.

- Benchérif, Khaled, Development and Deployment of a Software Safety Development Process for an Unmanned Vehicle System, 2006.
- Stambollian, Anabel, Improvement of Performances of Software Engineering Processes for Small and Medium Sized Enterprises. Project co-directed with Professor J-M Desharnais, ÉTS, 2006.
- Rheault, Frédéric, Software Process Performance Improvement Project for a Government Organization in Quebec. Co-directed with Professor Michel Plaisant of UQAM, 2005.
- Deniger, Frédéric, Development and Deployment of a Project Management Framework for a Multi Media Company, 2005.
- Piedboeuf, Dominique, Evaluation and Improvement of the performance of the Estimation and Requirement Management Processes at the Land Software Engineering Center of the Department of National Defense, 2004.
- Mougja, Rachid, Technology Change Management Project on Cyclomatic Complexity Measurement Tool Introduction at Motorola Canada Software Center, 2003.
- Belkebir, Youssef, Analysis And Improvement of the Role Definitions of the Software Engineering Process of the Competency Center of Bombardier Transportation, 2003. Co-directed with Professor Pierre Bourque of ETS.
- Yagoub, Saïd, Adaptation of the Capability Maturity Model to an Academic Environment, 2003.
- Quenum, William, Implementation of a Metrics Program at Bombardier Transport, 2003.
- Bouet, Mickaël, Development and Deployment of a Engineering Project Support Group at Oerlikon Aerospace, 2002.
- Guessous, Youssef, Development and Deployment of Software Engineering Processes at Astec APS, 2001.
- Amar, Fouzia, Requirements Engineering of a Digital Jukebox, 2001.

Table A-25 List of Completed Projects of Graduate Students

Table A-26 lists the projects that are not completed yet.

- Guillemot, F., Développement et implantation d'un processus de génie logiciel pour un groupe de recherche en neuroscience et d'une trousse de déploiement pour la conception logicielle inspirée de la future norme ISO pour les très petites structures.
- Socarras, R., Développement et mise en œuvre d'un plan d'amélioration des processus de génie logiciel pour le groupe de développent de systèmes embarqués chez Gemalto Canada Inc.
- Goyette, Martin, Recherche, Développement et Implantation d'une gestion de projet améliorée chez AXON Intégration & Développement.
- Kabli, Samia, Conception, réalisation et mise à l'essai de trousses de déploiement pour faciliter et accélérer l'implémentation de la norme ISO/CEI 20000 par les très petites structures.

Table A-26 List of Ongoing Projects of Graduate Students

THE CONDUCT OF AN INTERNATIONAL SURVEY OF VERY SMALL ENTITIES[87]

This project has been realized by a graduate student under the supervision of the author of this thesis (Bluteau 2007).

At one the ISO working group meeting (BK1-014 2005), the author proposed to participants to conduct a survey of VSEs to better understand the problems and needs of VSEs about the utilization of standards. A set of specifications for the survey was developed by the members of the working group and a project was given to a graduate student in software engineering (Bluteau 2007) with the following high level objectives:

- Develop a preliminary set of questions to address the specifications of the WG. The working group will review and finalize the set of questions.
- Develop a web site to host the survey. The site will have to be able to host the survey in about 7 to 10 languages.
- Collect the data on monthly basis
- Perform an analysis of the data

The development, conduct and results of the survey are presented in details in another chapter.

[87] This section is adapted from an article originally published in (Laporte et al. 2008a)

IMPROVING SOFTWARE PRACTICES AND PROCESS PERFORMANCES IN SMALL SETTINGS[88]

This master degree project has been realized by Anabel Stambollian, a graduate student under the supervision of Jean-Marc Desharnais, then professor at ÉTS, and the author of this thesis.

The main goal of this project was to find a way to assess and then improve software processes and practices those very small settings, while keeping all resource consumption at its lowest. The assessment method chosen for this project is the "Micro-Evaluation[89]", which took into account the particular context of small settings.

Several assessments have been done in France, in order to propose improvements to their processes. Table A-27 lists the organizations that have participated to this project.

Location and date of establishment	Number of IT employees
Gémenos, 2000	9
Avignon, 2001	2
Aix-en-Provence, 1999	4
Aix-en-Provence, 1998	15
Avignon, 1985	30
Montbonnot, 2003	3 permanent 8 on call
Avignon, 1996	45
Saint-Rémy-de-Provence, 1998	10

Table A-1-27 Characteristics of small and very small organisations assessed (Stambollian et al. 2006)

Strengths and weaknesses of existing software practices were documented and compared to the results of previous studies performed at the Belgium technology transfer center (CETIC) in collaboration with Professor Habra of Université de Namur (Habra et al. 1999). The assessments also provided the opportunity to design customized, light and low cost improvement plans. In turn, the approach allowed making recommendations to improve the Micro-Evaluation.

[88] This section is adapted from an article originally published in (Laporte et al. 2005b) and (Stambollian et al. 2006)).

[89] Assessment method developed and improved in collaboration with the *Observatoire Wallon des pratiques logicielles* (OWPL) from the *Centre d'excellence en technologies de l'information et de la communication* (CETIC), University of Namur, Belgium.

This project provided advances to the research in progress at ÉTS that will have, in time, a positive impact on the performances of Québec's small settings. It is also possible that the Micro-Evaluation will sustain improvements. This project also provided inputs to the first draft of a future software process lifecycle standard adapted to very small settings.

DEVELOPMENT OF SOFTWARE TESTING PROCESS FOR THE RESEARCH LABORATORY ON WIND TURBINES

This project has been realized by Marco Foisy, a graduate student under the supervision of the author of this thesis (Foisy 2007).

The project involved the research group on wind turbines of ÉTS. The main objective was to develop processes, plans and methodologies to facilitate the development and maintenance of the software programs more easily. Since the goal of such a group is to perform research, software development and maintenance was not of a high priority. As an example, one software program, develop by a student, consisted of about 11,000 line of not documented code. The software analysed the performances of wind turbines. To better understand the context, the graduate student reverse-engineered one software program and document the specifications, design, code, etc. The student also developed a traceability matrix from the specifications to tests. A configuration management process had been defined and implemented using an Open source software. The student also developed templates, for a software project plan and a quality assurance plan, tailored to the needs of the laboratory.

PROJECT MANAGEMENT OFFICE DESIGN AND SOFTWARE PROCESS IMPLEMENTATION AT VIRCOM INC.

This project has been realized by Yacouba Sédion, a graduate student under the supervision of the author of this thesis (Sedion 2007).

The goal of the project was to develop and establish a project management office (PMO) and software processes for a company called Vircom. Founded in 1994, the company offered protection systems that allow clients to manage outbound and inbound email communications while protecting against spam, viruses, fraud, phishing, and non-authorized communication. Vircom had about 35 employees having 5 teams working on 5 products.

During the project, processes, procedures, practices and data storage framework were defined. The management and development's practices, based on Project Management Body of Knowledge of Project Management Institute (PMI 2008), were added to the processes of the company and tested through a 5-month project.

SOFTWARE PROCESS IMPROVEMENT IN VERY SMALL ORGANIZATIONS

This project has been done by a graduate student under the supervision of the author (Bégnoche 2008). This process improvement project was performed, using the draft version of the ISO/IEC 29110 standard for VSEs, in a very small team within a larger organisation. This organisation distributes and supports three types of software products: Computer Aided Design, Computer Aided Manufacturing and Computer Aided Engineering. The products serve mainly the aerospace and the automobile industries.

The process improvement project lasted about 4 months. The student was responsible, within the VSE, to define the tasks of 4 developers and undertook to improve the following processes with the approval of management:

- Project management
- Software configuration management
- Issue tracking
- Requirements management

A software process improvement approach, adapted to the needs of this VSE, was developed in order to implement, adapt and improve the processes developed by the student.

The student developed a set of step-by-step guides for the implementation of version control practices using an Open source tool (SVN), Project management practices using GForge, Requirements management practices using XMLbasedsrs and configuration management practices using SVN. Volume III contains a few guides developed by this student.

DEVELOPMENT OF PROFILE AND DEPLOYMENT PACKAGE FOR THE IT GOVERNANCE STANDARD FOR VERY SMALL ENTITIES

This project has been realized by Samia Kabli, a graduate student under the supervision of the author (Kabli 2009). The graduate student used the ISO/IEC 20000 standard to develop a profile and a set of deployment packages. This standard promotes the adoption of an integrated process approach when establishing, implementing, operating, monitoring, measuring, reviewing and improving a Service Management System (SMS) to design and deliver services which meet business needs and customer requirements (ISO 2005e). The concepts of profile and deployment package are presented in a next chapter.

Improvement of Software Engineering Process at Bombardier Transportation

In this section, the author describes the mandates performed at Bombardier Transportation. The work performed at Bombardier is interesting because this organization, as described below, is an 'assemblage' of many small and very small software development centers.

OVERVIEW OF BOMBARDIER TRANSPORTATION

Bombardier Transportation, which was created in 1974 to provide subway wagons for the Montreal Transit Authority, grew through many acquisitions to become a manufacturer of rail material for moving people. The company had 16,000 employees before acquiring ADtranZ in 2001, an acquisition that brought the company 20,000 additional employees with an engineering presence in 25 countries. It is interesting to note that ADtranZ had also been the result of a merger, in that case of sections of ABB and Daimler Chrysler. Bombardier Transportation now has more than 36 software engineering sites located in 12 countries in Europe, North America and Asia, bringing the number of people employed in software engineering-related jobs to a total of around 1,000. The number of software developers, in each site, varies between 5 and 150 with an average of 27 practitioners. Over 23 sites have less than 26 software practitioners of which 7 have less than 8 practitioners. In other words, 23 sites met the definition of a VSE.

SOFTWARE ENGINEERING CENTRE OF COMPETENCE

Bombardier Transportation established a Software engineering Centre of Competence (CoC) to support the various divisions in reducing technical risks and quality deficiency costs, and in continuously improving the reliability of BT products. The major role of the CoC is to bridge the gap between increasing demands on new functionalities and cost reduction, while at the same time increasing the maturity level of Bombardier's software engineering capacity.

The following constitute the mandate of the CoC:

- Support world-wide utilization of engineering experts and tools
- Support projects to help meet the schedule
- Ensure product performance through Technical Risk Analysis and Risk Mitigation
- Support standardization and modularization
- Support total cost reduction
- Ensure technical competitiveness both today and in the future

- Generate a culture of openness and willingness to share and support

In order to provide technologies to all divisions, and to do so at a rapid pace, it has been decided that a common vocabulary, common processes and common roles will be used. The strategy developed to achieve this is as follows:

- Adopt internationally recognized reference documents
    - Models
    - Standards
    - Body of Knowledge
- Develop common processes, work instructions and role definitions
    - Independent of the organizational structure

So far, the CoC has led the development of integrated software engineering processes (BES), a set of software engineering Roles and Responsibilities (Bourque et al. 2004, Laporte et al. 2005d), a set of Peer Reviews ranging from informal desk check reviews, to walkthroughs and inspections (IEEE 2008a), and the measurement of the cost of quality of software development (Berrhouma et al. 2009).

1.6.4 Development of a set of Roles for Bombardier Transportation[90]

As stated by Humphrey (Humphrey 2000): "Without clearly identified responsibilities, it could take some time for a team to understand everything that it must do, to decide who should do each task…That is not so much because the engineers don't want to take responsibility but rather because they don't know what all the actions are or they are not sure whether anyone else is already doing them. They may also be reluctant to take on tasks that the team or team leader might plan to give to someone else." This why we have conducted a project to improve the software engineering role definitions within the software engineering process definition of a large multinational organization. The project was also conducted because of the many conflicting and sometimes absent role definitions across the many Bombardier Transportation[91] sites and projects. Moreover, many of the roles that had already been defined were defined only very briefly.

In order to facilitate the roll-out of the role definitions to all Bombardier Transportation software engineering sites, it was decided that the coverage analysis and subsequent improvements to the role definitions would be founded on internationally recognized

[90] This section is adapted from an article originally published in (Laporte et al. 2007b)

[91] www.bombardier.com

reference documents. This was viewed as a way of adding credibility to the improved role definitions without giving the impression that one software engineering site was imposing its role definitions on the other sites. The selected reference documents were IEEE/EIA Standard 12207, Standard for Information Technology–Software Life Cycle Processes, the IBM Rational Unified Process (RUP)[92] and the Guide to the Software Engineering Body of Knowledge (SWEBOK Guide). Figure A-18 lists the set of software engineering roles and Figure A-19 presents a description of the Software Architect Role.

[92] Version 2001A.04.00 of IBM RUP was used in this project.

Name of Roles
Change Control Board
Customer
Product Manager
Project Manager
Proposal Coordinator
Safety Representative
Senior Manager
Software Architect
Software Engineering Manager
Software Implementer
Software Infrastructure Administrator
Software Integrator
Software Metrics Coordinator
Software Process Engineer
Software Project Coordinator
Software Project Manager
Software Quality Assurance Engineer
Software Quality Assurance Manager
Software Requirements Coordinator
Software Team Leader

Figure A-18 List of Software Engineering Roles

Purpose:

The Software Architect establishes the overall software architecture. Thus, in contrast with the other Roles (e.g. Software Implementer), the Software Architect's view is one of breadth, as opposed to one of depth.

The Software Architect is responsible for articulating the architectural vision, conceptualizing and experimenting with alternative architectural approaches, creating models and components, interface specification documents, and validating the architecture against requirements and assumptions. Activities in this area include the creation of technology roadmaps, making assertions about technology directions and determining their consequences for the technical strategy, and hence architectural approach. This role involves not just these technical activities, but others that are more political and strategic.

Core Responsibilities:

- Derive the requirements for the system and software architecture.
- Identify the key design issues that must be resolved to support successful development of the software.
- Generate one or more alternatives and constraints for the architecture and select a solution.
- Allocate the software and derived requirements to the chosen architecture components and interfaces.
- Maintain requirement traceability for the software architecture's requirements.
- Describe the software architecture by capturing the design results and rationale.
- Identify appropriate derived requirements that address the effectiveness and cost of life-cycle phases following development, such as production and operation.

Hard Skills:

- Ability to identify technical project risks based on the software architecture model.
- Ability to perform software modeling and architecture conception/definition.
- Ability to perform conceptual product design, and specify a software architecture and implement a software system embodying it.
- Ability to apply modeling techniques such as use case, and other techniques, using the UML notations.
- Ability to apply Architectural Styles, Reference models and Reference Architectures.
- Ability to specify structural descriptions with techniques and notations such as: Architecture Description Language, Class Responsibility Card, Entity Relation Diagram, Interface Description Language, Jackson structure Diagrams and Structure Charts.
- Ability to specify behavioral descriptions with techniques and notations such as: Program Design Language PDL, Data Flow Diagram DFD and Flowcharts.
- Ability to use computer-aided software engineering (CASE) tools in an architecture-driven design process.
- Knowledge in concepts of structural patterns such as layers and client/server, mechanisms such as brokers and bridges, and middleware such as CORBA, DCOM, RMI.
- Knowledge in operating system architectures, compiler and interpreter design, and Real-time and Embedded Systems.
- Ability to perform software engineering activities across the full development cycle, including analysis, design, implementation, testing and documenting.
- Understand the business context of Bombardier Transportation and its competitors, their products, strategies and product generation processes.

Figure A-19 Definition of the Software Architect Role

Some of the work performed in this project, as it will be presented in a next chapter, will be used in the development of the set of Deployment packages. The definition of roles and associated responsibilities and skills is rarely done in VSEs. This situation often creates confusion among the employees of VSEs such as: who does what?, how do we select a potential employee?, which training is required for an employee?

1.6.5 Improvement of Software Engineering Performances at Bombardier Transport[93]

The performance of one of the Bombardier Transportation sites, the TTS/Pittsburgh Signaling group, has been evaluated twice: first in November 2003 and again in January 2006. The 2003 evaluation established a baseline for the evaluation of progress made in 2006. During those visits, the same evaluation method was used to evaluate project performance and organizational change management, i.e. the people issues. This section summarizes the multi-dimensional methodology used to perform the evaluations.

The Total Transit Systems (TTS) Division offers transportation solutions for urban and airport applications. The TTS portfolio of products includes fully automated advanced rapid transit, people mover and monorail systems, as well as guided light transit, light rapid transit and metro systems. More specifically, the Pittsburgh (Pennsylvania, USA) site of this division is mandated to develop and manufacture people mover systems. Around 100 software engineers work at this site and at the Signaling area, with some 30 additional people at the Bombardier Transport Engineering Center in India site in Hyderabad.

[93] This section is adapted from an article originally published in (Laporte et al. 2007e)

An evaluation method has been developed to assess real organizational progress on three dimensions: Process, Technology and Engineering (People) (see Figure A-20). As illustrated in this figure, the three dimensions are used to support business objectives (e.g. Business Context). Such a methodology sends the message, to management and employees, that progress should not be made in one dimension, e.g. process maturity, to the detriment of the other dimensions. Doing so ensures real commitment from the organization at every level, since visible return on investment is expected from any improvement initiative.

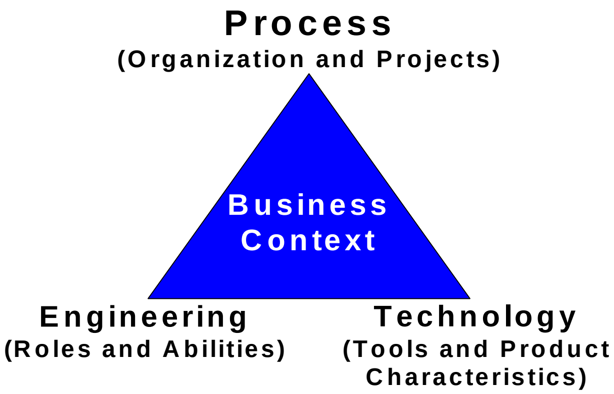


Figure A-20 The three dimensions of the Evaluation Methodology (Laporte et al. 2007)

EVALUATION OF THE PROCESS DIMENSION

The Process dimension was a tailored version of the industry-proven Capability Maturity Model (CMM) [94] evaluation methods. Depending on business needs (organizational and project list) and the scope of the evaluation, the Process Areas, or Key Process Areas (KPA) for the Software Capability Maturity Model (SW-CMM), were prioritized (high/medium/low).

EVALUATION OF THE PERFORMANCE MEASURES DIMENSION

Performance measures are mandatory if the contribution of the process to the achievement of business goals is to be correctly assessed. The first step was to identify the performance measures in use in the organization. Then, the methods and values were validated for applicability, validity and correctness. Finally, the results were used to evaluate the performance dimension. The elements considered during the evaluation are:

- Quality: Measure the quality (i.e. number of defects) of the software developed
- Productivity Index: Measure both the productivity and the productivity

[94] CMM and Capability Maturity Model are registered with the U.S. Patent and Trademark Office. CMM Integration and CMMI are service marks of Carnegie Mellon University.

improvement over time, and to use the results as a basis for future estimation.

EVALUATION OF THE PEOPLE DIMENSION

Since the management of change is a key element of a successful process improvement program, a series of actions was planned to facilitate the development, implementation and adoption of the processes, methods and tools (Laporte 1993, Laporte 1998, Laporte 1999). This structured approach, similar to the approach used in Oerlikon Aerospace described above, was used to evaluate and manage the human and cultural elements of an organization in order to manage the changes necessary to meet its business objectives. The organization's change readiness was evaluated, using Implementation Management Associates (IMA, www.imaworldwide.com) tools, by measuring the elements listed in Table A-27.

Organization's stress level	Evaluation of the priorities for resources in the organization
Sponsor assessment	Evaluation of the resources, reinforcement (e.g. motivation) and communications commitments made and demonstrated by the sponsor(s) of a change project
Change agent skills	Evaluation of the skills and motivation of those responsible for facilitating the implementation of organizational changes
Individual readiness	Evaluation of the reasons why people may resist an organizational change
Culture assessment	Assessment of the fit between the desired change and the actual organizational culture in order to identify potential barriers and to leverage actual cultural strengths
Implementation history	Assessment of barriers and lessons learned from previous change projects (since past problems are likely to recur, this tool allows identification of the issues that need to be managed for the change project to be successful)

Table A-28 Elements measured with the IMA Tools

The assessments performed allowed the organization to better identify potential barriers and perform mitigating actions to increase the likelihood of the success of a change project.

1.6.6 Measuring the Cost of Quality of a Large Software Project at Bombardier Transportation[95]

A measure of the cost of software quality has been made at the software development group of Bombardier Transportation Company located in Québec. A team of 15 software engineers has developed the software to control the subway of a large American city. A project to measure of the cost of software quality was carried out in four stages: preparation of a list of typical tasks related to the costs of software quality, categorization of these tasks (prevention, assessment and correction of anomalies), development and

[95] This section is adapted from an article originally published in (Berrhouma et al. 2009)

application of weight factors, and finally the measure of the cost of software quality. A set of 27 rules have been developed and weights were assigned to each task of the project. Over 1,100 software tasks were analyzed on a project totalling 88,000 hours.

The measurements show that the cost of software quality represents 33% of the overall project cost. The cost of rework, or cost or anomalies is 10%, the cost of prevention is 2% and the cost of evaluation is 21% of the total development cost. The cost of software quality is comparable to the values published by organizations with a similar level of maturity with the exception of a higher cost of evaluation. This may be explained by the fact that the software developed by this team is more demanding since it is real-time embedded software. Since this type of software is critical to the operation of the subway, it required more evaluation activities such as tests.

The results of this project should be applied in other development centers of Bombardier. As presented above, many development centers are typically VSEs. We should also be able to reuse the results of this project within the work of WG24.

1.7 Software Quality Assurance Course at Chiang Mai University

The author taught, during the summer of 2008, a course about Software Quality Assurance (SQA) to undergraduate students in software engineering. Essentially, the content of the course is similar to the course taught at ÉTS. It was the first time that the author spent about one month in an Asiatic culture. The lessons learned during this course will be presented in the chapter that discusses the development of international standards and the impacts of culture in the development and application of standards.

1.8 Conclusion

In this chapter, the author presented the work performed in large projects or large organizations such as the Department of National Defense, Oerlikon Aerospace and Bombardier Transport.

After joining ETS, in the summer of 2000, the author had to adjust his framework to the students in his classes. It took a little while to realize that these students came in class with a hands-on experience of very small organizations. They had difficulties understanding the concepts presented in class since they were presented by someone having worked only in large and very large organizations. They were 'complaining' that the material presented in the initial software quality assurance (SQA) course were 'too theoretical'. At that time, the author did not quite grasp the meaning of that message. After many discussions with many students, the author adjusted the course material to reflect a wider spectrum of organizations: i.e. varying from 'SQA-in-the-large' to SQA-in-the-very-small'. Similar adjustments were also made to graduate courses by adding a ‘hands-on’ intervention in a real organization or project. This chapter was also presented for the reader to better understand the involvement and motivation of the author in the development of international standards for VSEs.

Annex A

Application de la future norme ISO 29110 à la Commission Scolaire de la Seigneurie-des-Mille-Îles

Rédigé par : Jean-Claude Viau, Yann Bourdeau, Matthieu Riopel

Travail réalisé dans le cadre du programme de maîtrise en génie logiciel

Vérification et assurance qualité de logiciels (MGL 805), session d'hiver 2009.

Objectifs

Un des objectifs de ce document est de présenter les différents aspects de la qualité logicielle en fonction des 3 trousses de déploiement utilisées dans le cadre d'une petite entreprise. Nous avons également identifié certaines lacunes et nous avons quelques recommandations pour tenter d'améliorer les processus. Pour ce faire, nous avons utilisé le projet Portfolio qui est présentement en cours de développement dans l'entreprise ainsi que les trousses de déploiement du groupe de travail 24 de l'ISO/IEC JTC1/SC7.

Présentation de l'entreprise

La Commission Scolaire de la Seigneurie-des-Mille-Îles (CSSMI) représente le plus important employeur dans les Basses-Laurentides, avec plus de 8000 employés, dont 6 300 rémunérés sur une base régulière. Elle représente 54 écoles primaires, 14 écoles secondaires, 2 centres de formation générale et 4 centres de formation professionnelle. La mission première de la CSSMI est d'offrir un milieu stimulant pour l'apprentissage des élèves et la formation académique de ceux-ci. Pour y arriver, elle a régulièrement recours à des développements informatiques pour aider son personnel à réaliser leur mandat. Elle avait donc recours à certaines firmes externes pour ces logiciels et c'est pourquoi depuis environ 2 ans, elle s'est dotée d'une équipe de développement. Pour l'instant, il n'y a aucune méthodologie de qualité dans l'entreprise. Une des intentions de ce projet est d'implanter un minimum de qualité afin de mieux gérer les projets et tout le cycle de développement. L'équipe de développement est composée de quatre personnes, 1 analyste et 3 développeurs. Un fait important de mentionner est que la majorité des projets pour ne pas dire la totalité sont d'une durée allant de 2 semaines à un maximum de 6 mois. Il est très rare de dépasser ce temps puisque ce sont des besoins internes et assez ciblés.

Méthodologie

Pour la réalisation de ce travail, nous avons utilisé la méthodologie suivante :

- Récupération et lecture des 3 trousses du groupe de travail 24 de ISO/IEC JTC1/SC7.

  - Exigences logicielles
  - Gestion des versions
  - Gestion de projet

- Analyse des trousses dans le cadre de l’entreprise et du projet Portfolio.
- Comparaison des trousses avec les documents et les façons de faire de l’entreprise.
- Nous avons ajouté et adapté dans les trousses des éléments en rapport avec OpenUp.
- Reprendre les 3 trousses et ajuster leurs contenus en fonction de l’entreprise.
- Identification des points positifs et négatifs toujours en fonction des trousses.
- Identifier les ajustements à faire et les recommandations pour rendre la CSSMI conforme aux trousses et ainsi améliorer la qualité des projets et des logiciels.
- Prioriser les changements à apporter autant aux documents qu’aux façons de faire dans cette organisation.

RECOMMANDATIONS

Gestion de projets

Afin de corriger plusieurs points faibles au niveau de la gestion de projets à la CSSMI, voici quelques recommandations pour améliorer ce processus. Dans un premier temps, il sera essentiel de mettre en place une procédure pour avoir l’acceptation du client pour le plan de projet. On devra donc prévoir une rencontre avec le client lorsque le plan de projet sera terminé afin de lui expliquer. Si tout est accepté tel quel, une signature du client s’impose, dans le cas où il manque certaines informations ou détails, on devra apporter les modifications et faire de nouveau approuver le plan de projet par le client. Il y aura également dans cette tâche un changement de pratique à apporter afin de nous assurer que le client spécifie le maximum de détails et d’éléments afin d’avoir un plan de projet réaliste et qui contiendra l’ensemble des besoins du client. De plus, avec cette étape, le client devra être plus précis et clair pour définir ses besoins puisqu’il y aura un certain contrat de signé avec lui et l’équipe de projets. Présentement, il n’est pas rare de voir des clients dire simplement le minimum et au fur et à mesure que le projet avance, font des changements et des ajouts sans tenir compte des impacts de ces demandes.

Au niveau de la description des projets et de leur étendue, ces étapes sont réalisées dans l’ensemble des projets à la CSSMI. Par contre, certains projets n’ont pas un niveau suffisant de détails selon nous pour bien comprendre leur étendue et leurs spécifications. On devra donc améliorer ces étapes afin de bien comprendre toutes les facettes du projet et ainsi être en mesure de mieux estimer et gérer le projet.

Dans le même ordre d'idées, la gestion des demandes de changements est une tâche clé dans la gestion de projets. On devra instaurer un système de demandes de changements documenté avec des études et analyses des impacts des demandes sur le projet. Une bonne pratique serait de prendre le gabarit de demande de changements dans la trousse de gestion de projets et l'appliquer à toutes les demandes des clients. Cette procédure nous permettra d'avoir un suivi des demandes ainsi que des arguments pour défendre les écarts entre les estimations et le temps réel des projets. De plus, tous les membres de l'équipe de développement pourront être au fait des changements et ainsi mieux se préparer et réagir en cas de problèmes.

Pour ce qui est de la gestion des risques, il serait bon d'utiliser le gabarit dans la trousse de gestion de projets afin d'identifier les risques les plus probables en fonction du projet et des clients. Cette démarche permettra de réduire au maximum certaines situations qui ont tendances à trop se répéter pendant les projets.

Une dernière recommandation serait de tenir un historique de tous les projets avec leurs estimations respectives, les demandes de changements, les ajustements et finalement le temps réellement passé sur le projet et sur les différentes tâches. Cette démarche permettra de pouvoir faire des statistiques et ainsi ajuster les estimations de départ et être plus juste dans la gestion des projets futurs. Cette démarche permettra également de se bâtir une base de connaissances en gestion de projets et faire des comparaisons entre les différents projets. De plus, on pourra avoir des statistiques sur le temps alloué à certaines tâches qui se répéteront dans d'autres projets et ainsi mieux gérer nos projets.

Exigences logicielles

- Le SRS devrait toujours être fait au complet. Un SRS non complet peut causer la confusion auprès des développeurs.

- Les rôles d'analyste, de client, de développeur et de gestionnaire de projets sont clairement expliqués. Cela va permettre à chaque membre de l'équipe de comprendre ce qu'il a à accomplir plus clairement.

- Processus de validations et vérifications. Le document décrit les étapes nécessaires pour faire la validation et la vérification des exigences. L'annexe B de la trousse sur les exigences logicielles fournit une grille de vérifications pour faciliter cette tâche.

- Processus de gestion des changements des exigences. Le document décrit les étapes à suivre pour faire la gestion des changements et en garder la trace. Un changement d'une exigence doit être fait avec le document de l'Annexe D. Ce document est une demande de changement (DDC). Ces documents devraient être sauvegardés dans un répertoire disque central. La solution préférable serait de les sauvegarder dans le gestionnaire de versions.

Gestion des versions

Pour corriger plusieurs points faibles, l'utilisation du serveur de gestion de versions Subversion est recommandée. Il permet de garder un historique des changements, d'automatiser les sauvegardes et de notifier les employés pour chaque changement. Il doit utiliser une structure comme celle utilisée présentement (par projet) avec quelques variantes pour suivre les bonnes pratiques recommandées par la communauté de Subversion. L'ensemble des détails de l'installation, la configuration et la structure des répertoires sont définis dans la trousse de gestion de versions. Il existe plusieurs outils facilitant l'utilisation d'un serveur SVN et la trousse recommande l'utilisation de TortoiseSVN puisque les utilisateurs travaillent sous Windows. Elle conseille également l'utilisation de TortoiseMerge pour comparer les changements entre deux versions d'un CI. Il existe plusieurs autres outils, mais ceux-ci devraient être amplement suffisants.

La présence d'un serveur de gestion de versions ne corrige pas tous les problèmes. Il faut que la version initiale approuvée soit bien définie et qu'elle soit continuellement à jour. Donc, il est nécessaire de favoriser les petits commis journaliers. Ces changements doivent cependant suivre une procédure pour s'assurer de l'intégrité des changements. Ils doivent entre autres utiliser un gabarit pour les logs messages tel que défini dans la trousse. Puis, les changements doivent être révisés par les pairs. Idéalement, un serveur d'intégration continue serait installé pour s'assurer de l'intégrité et la qualité de tous les projets après chaque changement. Toutefois, cet aspect peut être reporté ultérieurement pour minimiser les changements aux procédures et aux standards de l'entreprise et faciliter leur adoption. L'emphase doit être mise sur le serveur de gestions de versions et aux autres procédures reliées aux demandes de changement et aux livrables.

Il est nécessaire d'introduire une procédure pour les demandes de changements. Cette procédure est décrite dans la trousse pour la gestion de projets et dans la trousse de gestion de versions. Elle permet d'uniformiser les demandes de changements. Toutes les demandes doivent passer par une DDC au lieu d'être décrites verbalement aux développeurs.

Chacun des livrables doit maintenant suivre le standard qui était déjà utilisé pour la moitié des livrables (Majeur. Mineur. Révision donc 1.0.0). Ce standard doit être défini et suivi pour chaque projet. Puis, pour chacun des livrables il est nécessaire de créer une étiquette (Tag) pour qu'il soit possible de retrouver les CI à l'état d'une relâche antérieure.

Finalement, pour s'assurer du succès de tous ces changements aux procédures et aux standards de l'entreprise, il est nécessaire de motiver et former les employés sur les bienfaits de ces changements. Il est nécessaire de fournir une formation sur Subversion et sur les nouvelles procédures/standards de l'entreprise.

Rapport de progrès – Juin 2009[96]

L'ensemble des recommandations sont acceptées mais ne sont pas toutes complétées. Dans certain cas, on va diminuer les exigences compte tenu de la taille de l'équipe et des projets.

[96] Extrait du courriel de l'étudiant de maîtrise Matthieu Riopel du 9 juin 2009.

Appendix B

Centre de support en génie logiciel aux Très petites organisations

Appendix B - Centre de support en génie logiciel aux Très petites organisations

Plan d'affaires

Document confidentiel

Déposé le 19 juin 2009

Centre de support en génie logiciel aux Très petites organisations

Plan d’affaires

Objectif
Création d’un centre de support aux TPOs de l’ÉTS

Promoteurs du projet

École de technologie supérieure

Directeur des relations avec l’industrie de l’École de technologie supérieure
M. Jean Belzile

Éditeur du groupe de travail 24 de l’ISO sur les futures normes pour les TPOs

M. Claude Y. Laporte
Professeur en génie logiciel
Département de génie logiciel et des TI

Noms des centres et des leaders membres du réseau des centres de support aux TPOs[1]

Belgique - Centre d'Excellence en Technologies de l'Information et de la Communication (CETIC) (M. Simon Alexandre, Directeur)

Colombie - Parquesoft Foundation (Mr. Orlando Rincon Bonilla, Président)

Finlande - Université de technologie de Tampere, Pori (Mr. Timo Varkoi)

France - Université de Bretagne Occidentale (Professeur Philippe Saliou)

Hong Kong - Université Polytechnique (Professeur Hareton Leung,)

Irlande - Lero, The Irish Software Engineering Research Centre (Professeur Rory O’Connor)

Luxembourg - Centre de Recherche Public Henri Tudor (M. Jean-Pol Michel, Director of the Centre for IT Innovation)

Thaïlande – Federation of Thai Industries (Mr. Anukul Tamprasirt, Deputy Secretary General)

[1] Les centres sont brièvement décrits à l’annexe A.

Table des matières

Liste des figures

Liste des tableaux

Analyse du contexte industriel visé par le projet

Croissance de la demande en logiciels

Aujourd'hui, la capacité des organisations à concurrencer, à s'adapter et à survivre dépend de plus en plus des logiciels. En 2010, un téléphone cellulaire contiendra plus de 20 millions de lignes de code. Un constructeur automobile a estimé que ses voitures auront bientôt plus de 100 millions de lignes de code (Charette 2005). La figure 1 illustre la croissance de demande en logiciels depuis les années 80. De toute évidence, ce taux de croissance se maintiendra pour plusieurs années à venir.

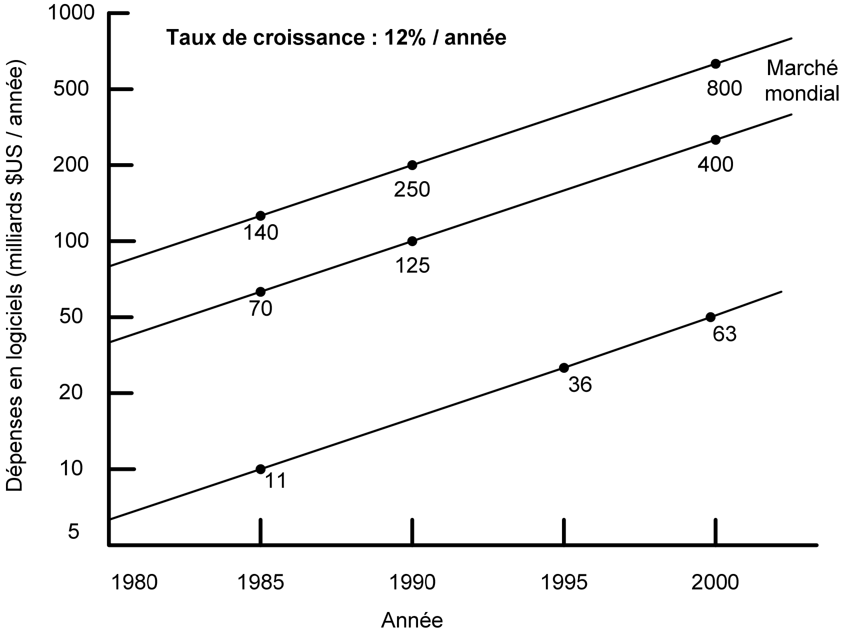


Figure 1 Croissance de la demande en logiciels (Boehm 1987)

Une dépendance accrue envers les fournisseurs logiciels

De plus en plus les fabricants dépendront de leurs fournisseurs pour développer des produits rapidement afin de répondre au marché et d'arriver le premier sur ce marché. Une chaîne de fabrication des grands produits de masse, a souvent une structure pyramidale. Par exemple, comme l'illustre la figure 2, un grand fabricant de produit de masse japonais a intégré dans l'un de ses produits une composante logicielle qui comportait une erreur non détectée lors des nombreux tests. Cette composante avait été produite par l'un de ses 6000 fournisseurs. La pièce défectueuse a entraîné une perte de 200 millions de dollars au fabricant (Shintani 2006).

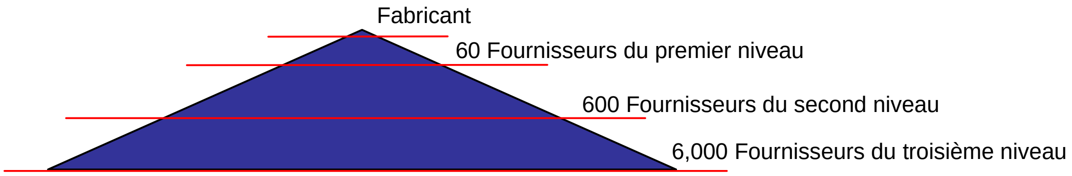

Fabricant
60 Fournisseurs du premier niveau
600 Fournisseurs du second niveau
6,000 Fournisseurs du troisième niveau

Figure 2 Exemple d'une chaîne d'approvisionnement d'un fabricant japonais

Les systèmes avioniques, tel qu'illustré à la figure 3, sont de plus en plus complexes. À titre d'exemple, la société Boeing, pour son modèle 777, a dépensé environ $800 millions pour le développement des 4 millions de lignes de code des 1280 processeurs à bord. Boeing pourrait dépenser cinq fois ce montant pour le logiciel du nouveau Boeing 787 (Long 2008). Un rapide calcul montre le nombre de personnes-années nécessaires pour développer les logiciels du 787:

- Coûts du développement logiciel du 787 = $ 800 millions (pour le Boeing 777) X 5 (pour le Boeing 787) = $ 4 milliards
- Nombre de lignes de code = $ 4 milliards / $ 200 par ligne de code = 20, 000,000 lignes de code.
- Nombre de personnes-mois = 20, 000,000 lignes/100 lignes par personne-mois = 200,000 personne-mois
- Nombre de personnes-années (sur une base de 10 mois de travail par année) = 200,000/10 = 20, 000 personnes-années

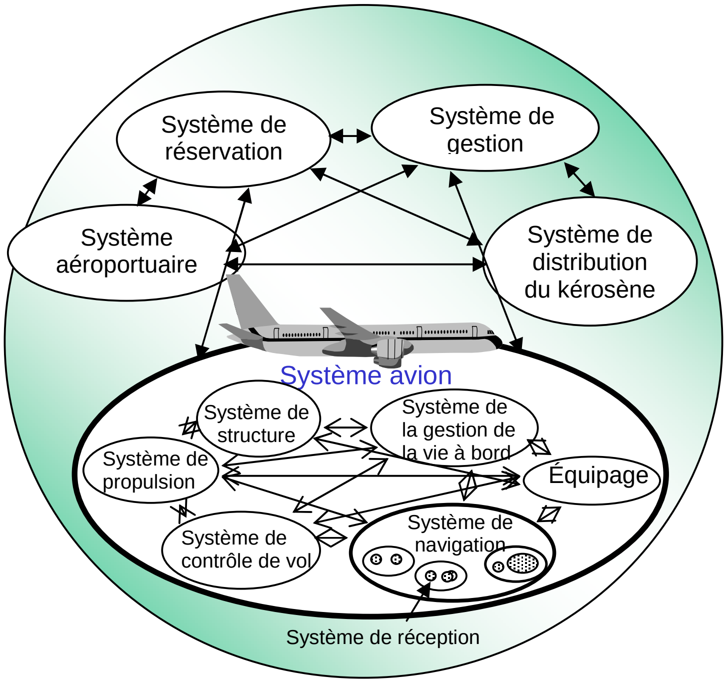


Figure 3 Vue systèmes d'un avion et de son environnement (AFNOR 2003)

Il est fort douteux que Boeing puisse développer, sans l'aide de nombreux fournisseurs, ces logiciels dans un délai raisonnable. Il est très probable que de nombreuses très petites organisations (TPOs) fourniront des composantes à Boeing ou à ses principaux fournisseurs, qui, à leurs tours, intégreront les éléments développés par les TPOs, à des sous-systèmes du 787.

Des fournisseurs logiciels de petites tailles confrontés à des exigences complexes

L'industrie reconnaît l'intérêt des petites et très petites organisations (TPOs) en logiciel en raison des produits et services précieux qu'elles offrent. En Europe, par exemple, 85 % des sociétés du secteur des technologies de l'information (TI) ont entre 1 et 10 employés. Au Canada, la région de Montréal, tel qu'illustré dans le tableau 1, comporte près de 80 % de sociétés de moins de 25 personnes, environ 50% des sociétés ont moins de 10 personnes (Gauthier 2004). Une autre étude réalisée en Wallonie (CITA 1997) fournit des données semblables où environ 60 % des organisations ont moins de 5 employés. Au Brésil, les petites entreprises représentent environ 70 % du nombre total des sociétés (Anacleto et al. 2004). Finalement, en Irlande du Nord (McFall et al. 2003), une enquête montre que 66 % des organisations emploient moins de 20 employés.

Taille (emploi)	**Entreprise**		**Emploi**	
	Nombre	%	Nombre	%
1 à 25	540	78 %	5 105	29 %
26 à 100	127	18 %	6 221	36 %
+ de 100	26	4 %	6 056	35 %
TOTAL	693	**100 %**	**17 382**	**100 %**

Tableau 1 Taille des sociétés du secteur logiciel de la région de Montréal (Gauthier 2004).

Les normes de génie logiciel de l'Organisation de normalisation internationale (ISO) n'ont pas été rédigées à l'intention des petits projets, ni des petites organisations de développement comprenant moins de 25 personnes. Elles sont souvent difficiles à appliquer dans de tels contextes. Les TPOs ont des moyens faibles, sinon très limités, pour être reconnues en tant qu'organisations produisant des systèmes logiciels de qualité et, par suite, n'ont pas accès à certains marchés. Actuellement, la conformité aux normes de génie logiciel exige une masse critique, en termes de nombre d'employés, de coût et d'efforts, que les TPOs ont des difficultés à atteindre.

Compréhension du contexte d'affaires des Très petites organisations

Les caractéristiques des très petites entreprises ont été identifiées et un sondage a été effectué auprès des TPOs pour bien identifier leurs problèmes et leurs besoins quant à l'utilisation des normes en génie logiciel.

Les caractéristiques des petites entreprises[2]

Les caractéristiques des petites entreprises, énumérées au tableau 2, ainsi que la spécificité de leurs besoins en font des entreprises très différentes des grandes entreprises (Mtigwe 2005). Ces caractéristiques sont classifiées en quatre catégories: les contraintes financières, le profil typique de la clientèle, l'orientation des processus opérationnels internes et les contraintes sur l'apprentissage et la croissance.

Caractéristiques	Petites entreprises	Grandes entreprises
Planification	Non structuré/opérationnel	Structuré/stratégique
Flexibilité	Élevé	Structuré/stratégique
Prise de risques	Élevé	Moyen
Processus de gestion	Informel	Faible à élevé
Capacité à apprendre	Limité	Moyen à élevé
Impact de changements négatifs du marché	Élevé	Plus facile à gérer
Avantage compétitif	Centré sur le capital humain	Centré sur le capital organisationnel

Tableau 2 Caractéristiques des petites et grandes entreprises (adapté de Mtigwe 2005).

Les TPOs sont économiquement vulnérables, car elles sont dépendantes de leur mouvement de trésorerie et des bénéfices de leurs projets de sorte qu'elles doivent absolument exécuter les projets dans le cadre du budget alloué. Elles ont tendance à avoir de petits budgets qui ont de nombreux impacts: elles ont peu de ressources allouées à la formation, peu ou pas de budget pour exécuter les activités d'assurance de la qualité, pas de budget pour la documentation des processus logiciels, un petit budget pour répondre aux risques d'un projet et un budget limité pour effectuer l'amélioration de leurs processus ou obtenir une certification ou une évaluation de leurs processus.

En général, les TPOs ont peu de clients et parfois qu'un seul client. Il revient souvent au client de s'occuper de l'intégration des logiciels, de leur installation et opération. Il n'est pas d'une pratique courante, pour le client, de définir des exigences de qualité quantitatives. Le développement de logiciels dans les petites et très petites entreprises dépend beaucoup de la communication verbale entre les développeurs.

Les processus d'affaires internes des TPOs sont généralement axés sur le développement de logiciels personnalisés (*custom software*). Ces logiciels sont développés progressivement pour répondre aux besoins du client. Habituellement, la plupart des processus de gestion, tels que les ressources humaines et la gestion des infrastructures, sont effectués de manière informelle. La

[2] Extrait et traduit de (Laporte et al. 2008b)

majorité des communications, la prise des décisions et la résolution des problèmes étant effectuée face à face et généralement non documentée.

La croissance des TPOs est caractérisée par un manque d'expertise en amélioration des processus et par un manque de ressources humaines à s'engager dans la normalisation de leurs processus. Les TPOs ont souvent une perception négative des normes. Elles sont perçues comme étant lourdes et complexes et faites par les grandes entreprises, pour servir les grandes entreprises.

En conclusion, une TPOs n'est pas une miniature d'une grande entreprise tant par sa structure, son style de gestion, ses capacités financières et la disponibilité d'expertise.

Les TPOs et les coûts associés au développement de logiciels

Il est important de savoir qu'une partie importante du budget d'un projet est la perte causée par l'absence de pratiques logicielles éprouvées dans les processus de développement de logiciels.

Le coût d'un projet est divisé en 2 parties: le coût de réalisation et le coût de la qualité. Le coût de la qualité peut être décomposé en trois catégories: les coûts de prévention, les coûts d'évaluation et les coûts des anomalies internes et externes:

- Les coûts de prévention sont les coûts encourus par une organisation pour prévenir l'occurrence des erreurs dans les diverses étapes pendant le processus de développement (ex. : conception, développement, production et expédition) d'un produit.
- Les coûts d'évaluation sont les coûts de vérification ou d'évaluation d'un produit aux diverses étapes pendant le processus de développement.
- Les coûts des anomalies, aussi appelés coûts de non-conformité, se divisent en deux types :
    - o Les coûts des anomalies internes lesquels sont tous les coûts résultants, des anomalies avant que le produit soit livré au client;
    - o Les coûts des anomalies externes lesquels sont tous les coûts encourus par l'organisation quand le client découvre des défauts.

Des études ont montré que 40% à 50% du temps de travail des spécialistes logiciels est attribué à la correction des anomalies, c.à.d. d'erreurs, qui auraient pu être évitées (Charrette 2005). Une étude faite par la société américaine Raytheon (Dion 1993; Haley 1996), montre à la figure 4 que le coût des reprises de travail (*rework*), lorsque la société était au niveau 1 du modèle de maturité (*Capability Maturity Model*, CMM) du Software Engineering Institute (SEI), était en 1987, d'environ 41% du coût total des projets, à 18% au niveau 2, à 11% au niveau 3 et à 6% au niveau 4.

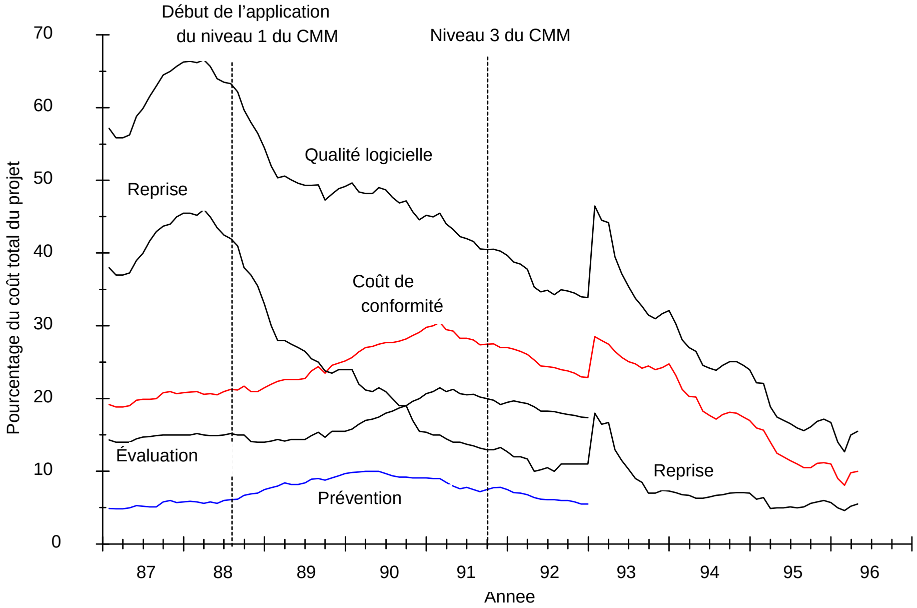


Figure 4 Les données de l'amélioration des processus logiciels (Dion 1993; Haley 1996)

Le SEI a montré qu'une augmentation du niveau de maturité comportait de nombreux avantages pour les organisations qui pouvaient investir dans l'amélioration de leurs processus: augmentation de la productivité, augmentation de la qualité, rencontre des budgets et calendriers, augmentation de la satisfaction des clients ainsi que de bons retours sur investissements (Gibson et al. 2006).

Les organisations visées par le Centre de support aux TPOs se situent typiquement au niveau 1 du modèle de maturité du SEI. Il y aurait donc place à de substantielles améliorations à leurs façons de faire pour réduire le coût des reprises et aider ces TPOs à être plus performantes et compétitives.

Des données, rassemblées dans le tableau 3, ont été fournies par des ingénieurs et des managers de 2 grandes sociétés et des étudiants de la maîtrise en génie logiciel de l'ÉTS. Ces données proviennent des coûts de la qualité observés dans différents environnements de développement de logiciels. Ces données, sauf pour la colonne 'Cours B', ont été recueillies dans deux multinationales: l'une œuvrant dans le secteur des transports et l'autre dans le secteur de l'aérospatiale. Comme l'illustre le tableau 3, le coût des reprises est d'environ 30%. Le nombre entre parenthèses indique le nombre de personnes de chaque échantillon.

	Site A É.U. Ingénieur (19)	**Site A É.U. Manager (5)**	**Site B Europe Ingénieur (13)**	**Site C Europe Ingénieur (14)**	**Site D Europe Ingénieur (9)**	**Cours A Canada 2008 (8)**	**Cours B ÉTS 2008 (14)**
Coût de réalisation	41%	44%	34%	31%	34%	29%	43%
Coût des reprises	30%	26%	23%	41%	34%	28%	29%
Coût d'évaluation	18%	14%	32%	21%	26%	24%	18%
Coût de prévention	11%	16%	11%	8%	7%	14%	10%
Qualité estimée	71	8	23	35	17	403	19

Tableau 3 Coût de la qualité des logiciels

La dernière ligne du tableau donne la qualité, estimée en nombre de défauts par 1000 lignes de code, des logiciels développés dans leurs organisations avant la phase de tests. À titre d'exemple, les logiciels commerciaux comportent de 1 à 10 défauts, et malheureusement parfois plus, par 1000 lignes de code. Un logiciel en opération ne devrait pas comporter plus de 0.1 à 0.5 erreur par 1000 lignes de code (Humphrey 2002).

Il n'a pas été possible d'obtenir des données semblables des TPOs car, presque toujours, ces organisations ne mesurent pas la performance de leurs processus. Mais, il est fort probable que les coûts de reprise soient au moins aussi élevés que ce qui est illustré dans le tableau 3.

Un sondage international sur les TPOs

En 2007, un sondage international a été piloté par le Professeur Laporte afin de valider les problèmes reliés à l'utilisation des normes en génie logiciel et de recueillir des données permettant d'identifier les besoins des TPOs afin de les aider à appliquer des normes et à devenir plus compétitives (Laporte et al. 2008a).

Afin de rejoindre le plus de TPOs dans le monde, le questionnaire du sondage a été conçu pour être répondu par internet et il a été traduit en 9 langues: allemand, anglais, coréen, espagnol, français, portugais, russe, thaï et turc. Tel qu'illustré au tableau 4, plus de 435 personnes, de 32 pays, ont répondu au sondage. Ce sondage a été réalisé dans le cadre d'un projet de maîtrise en génie logiciel d'une étudiante de l'ÉTS (Bluteau 2007).

Pays	Nombre de réponses	Pays	Nombre de réponses
Argentine	2	Italie	2
Australie	8	Japon	3
Belgique	10	Corée du sud	4
Brésil	68	Mexique	20
Bulgarie	3	Nouvelle Zélande	1
Canada	8	Pérou	4
Chili	1	Russie	4
Colombie	88	Afrique du sud	10
Tchécoslovaquie	3	Espagne	2
Équateur	9	Taiwan	1
Finlande	13	Thaïlande	52
France	3	Turquie	1
Inde	57	Angleterre	2
Irlande	10	États-Unis	3

Tableau 4 Réponses au sondage sur les TPOs (Laporte et al. 2008a)

Il est intéressant de noter que nous avons recueilli un nombre élevé de réponses en provenance de l'Amérique latine et de l'Asie, mais peu de réponses de l'Europe et des États-Unis.

En ce qui concerne la certification, un constat intéressant du sondage est la différence dans le pourcentage des organisations certifiées: moins de 18% des TPOs sont certifiés, tandis que 53% des plus grandes organisations sont certifiées.

Pourquoi les TPOs n'utilisent pas les normes

Les normes sont très peu utilisées par les TPOs qui ont répondu au sondage international. Comme le montre la figure 4, les principales raisons sont :

- Un manque de ressources (28%);
- Les normes ne sont pas exigées (24%);
- Les normes sont difficiles et bureaucratiques et ne fournissent pas de directives adéquates pour une utilisation dans un environnement de petite entreprise (15%).
- Les normes prennent trop de temps (*Time consuming*) (14%)

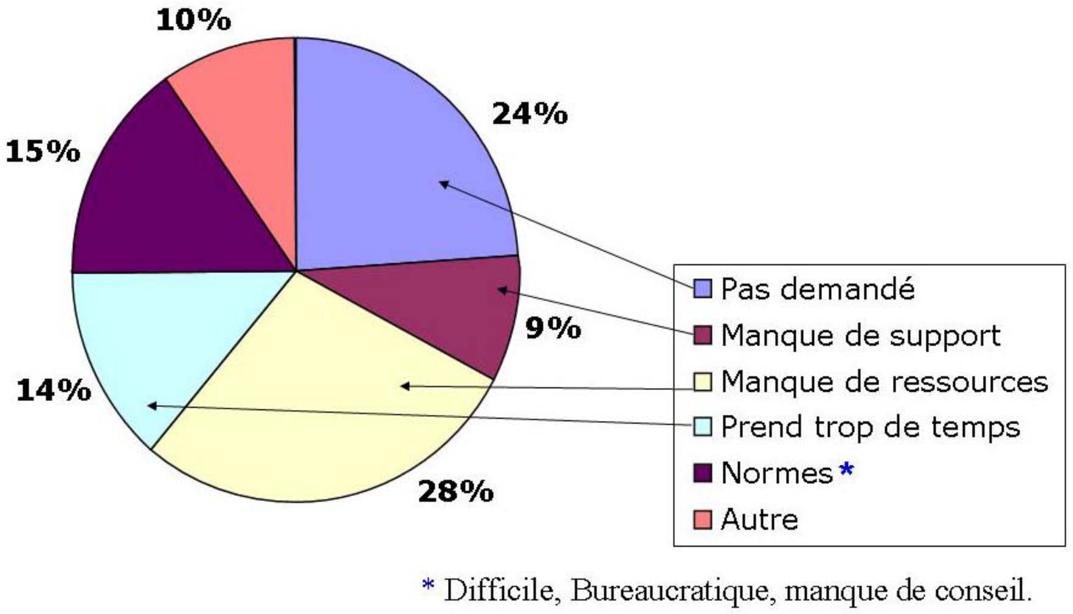


Figure 4 Pourquoi les TPOS n'utilisent pas les normes

Pour une grande majorité (74%) des TPOs, il est très important d'être évaluée et certifiée par rapport à une norme. Une certification ISO est demandée par 40% d'entre elles. Peu de TPOs sont intéressées par une certification nationale. De leur point de vue, les avantages offerts par la certification sont :

- L'accroissement de la compétitivité;
- Une plus grande confiance de la clientèle;
- Une plus grande qualité des produits logiciels;
- Une augmentation de l'engagement de la gestion de la TPE pour l'amélioration des processus;
- Une diminution des risques de développement;
- Une mise sur le marché facilité par une meilleure image;
- Un meilleur potentiel à l'exportation.

Les TPOS ont exprimé le besoin d'une assistance pour les aider à mettre en œuvre les normes. Plus de 62% souhaiteraient des guides, des gabarits et des exemples. Les TPOS veulent des normes légères et faciles à comprendre. Enfin, elles ont indiqué qu'il doit être possible de mettre en œuvre les normes à un coût minimum, dans un court laps de temps et avec peu de ressources.

Contexte d'affaires des TIC : Canada et région de Montréal

Contexte dans lequel évoluent les TPOs et le secteur industriel visé

Les industries des technologies de l'information et des communications du Canada – principalement le matériel informatique et les logiciels, les télécommunications et les services professionnels connexes – comptent pour 140 milliards de dollars du PIB du pays. Voici quelques chiffres qui résument bien la situation. De 2002 à 2007, la croissance du PIB du secteur des TIC était de 5,1 % annuellement, contre 2,7 % pour l'économie globale. L'industrie est d'ailleurs la moins touchée par la crise économique étant donné sa très grande capacité d'innovation. Cette industrie est de plus un moteur de croissance. Il faut donc tirer profit de la crise économique actuelle en réaffirmant le positionnement du Grand Montréal sur le plan mondial.

Les prévisions entourant le secteur des TIC estiment que ce secteur va se développer encore très rapidement puisqu'il s'imbrique dans une variété de domaines (énergie, sciences de la vie, santé, sécurité, affaires, médias, communications interpersonnelles, etc.). Cette croissance ne peut se baser que sur les connaissances, les habiletés et le talent; des enjeux centraux d'une économie du savoir florissante. Il est primordial de repenser les priorités de Montréal afin d'assurer que les TIC puissent y prendre la place stratégique qui leur revient.

À cet égard, la création d'une grappe des TIC et l'apport de TechnoMontréal ne peuvent que contribuer au succès économique du Grand Montréal. Le secteur des TIC dépense plus de 5,7 milliards de dollars par an en recherche et développement, soit deux fois plus que n'importe quel autre secteur, et 40 % de toutes les dépenses en R & D du secteur privé. C'est tout un exploit pour une industrie qui ne représente que 5,5 % du PIB. Il est encore une fois évident qu'une grappe dynamique des TIC à Montréal est essentielle pour la santé économique de la ville. À cet égard, TechnoMontréal soutient les entrepreneurs en facilitant l'accès au capital et en soutenant les initiatives de financement pour la R & D et pour l'innovation, par exemple. L'accès au capital et les investissements dans le domaine des TIC constituent les principaux moteurs de croissance de la productivité, tant au niveau des organisations que de l'économie dans son ensemble. C'est pourquoi, TechnoMontréal est un joueur important de la grappe des TIC puisque l'organisme vise à réunir les intervenants montréalais impliqués dans le soutien à l'innovation en plus d'établir le leadership mondial de la métropole en positionnant Montréal comme un centre de créativité et d'innovation.

Au Canada, le secteur des technologies de l'information et des communications (TIC) emploie annuellement près de 1,1 million de Canadiens, soit davantage de personnes que ceux de l'agriculture, de la foresterie, de la pêche, de l'exploitation minière, du pétrole et du gaz, des services, et du secteur manufacturier de l'industrie des transports (incluant l'automobile), combinés. Il s'agit d'une main-d'œuvre fortement scolarisée où plus des deux tiers ont une formation postsecondaire.

À Montréal, la grappe des TIC regroupe 5 000 entreprises dans des secteurs aussi variés que ceux de la fabrication, des logiciels, des services informatiques, des services de télécommunications, des services et médias numériques interactifs, de l'audiovisuel et du son numérique et des arts numériques, totalisant 120 000 emplois. Les TIC sont essentielles à la prospérité et à la productivité de toutes les industries : médias et culture, édition, publicité, ressources naturelles, services financiers, soins de santé, construction, détail ou éducation. Ces industries « utilisatrices » emploient plus de 225 000 professionnels en TIC, dont la contribution est essentielle à la créativité, l'innovation, la satisfaction de la clientèle, la productivité, la sécurité et l'avantage concurrentiel. Les emplois dans ces secteurs évoluent et le Canada doit intensifier ses efforts pour rester concurrentiel et gagner dans la conjoncture économique d'aujourd'hui. À Montréal, TechnoMontréal est un organisme essentiel à la préservation d'un réseau solide pour affronter ces défis.

Extrait des Recommandations du conseil d'agglomération sur le développement économique

Ci-dessous, un extrait des recommandations de la séance du 11 mai 2009 de la Commission du Conseil d'agglomération sur le développement économique de la Grappe des TIC, des recommandations majeures pour la valorisation du secteur des TIC dans la région de Montréal.

«…LA COMMISSION PERMANENTE DU CONSEIL D'AGGLOMÉRATION SUR LE DÉVELOPPEMENT ÉCONOMIQUE
Remercie la direction et le personnel de TechnoMontréal pour leur collaboration active et fructueuse aux travaux de la commission,
Et remercie les fonctionnaires qui ont participé au processus pour la qualité de leurs interventions lors des séances de travail de la commission.

CONSIDÉRANT l'importance de la Grappe des technologies de l'information et des communications (TIC) qui génère des revenus de l'ordre de 25 G$ annuellement et totalise quelque 116 000 emplois dans la région de Montréal;
CONSIDÉRANT que 12 500 de ces emplois se retrouvent dans des activités de recherche et de développement essentielles à la croissance de la Grappe;
CONSIDÉRANT la concurrence accrue d'autres villes dans le domaine des TIC;
CONSIDÉRANT la nécessité de promouvoir les investissements dans les TIC dans l'agglomération de Montréal;

La commission recommande :

Reconnaissance et croissance de la Grappe

R-1 Que le Conseil d'agglomération reconnaisse l'importance de la Grappe des technologies de l'information et des communications (TIC) dans le développement économique et social du Grand Montréal, avec une position de moteur économique pour l'ensemble du Québec

R-2 Que le conseil d'agglomération appuie l'objectif retenu par TechnoMontréal de croissance de 50 % en cinq ans de l'industrie des TIC à Montréal et mandate le Service 2 de la mise en valeur du territoire et du patrimoine pour qu'il mette en œuvre des moyens pour concrétiser cet appui.

R-3 Que le Conseil d'agglomération mandate le Service de la mise en valeur du territoire et du patrimoine pour qu'il étudie plus en profondeur, avec la collaboration de TechnoMontréal, des mesures de soutien et d'accélération du développement de l'industrie des TIC sur son territoire afin d'en maximiser les retombées sociales et économiques.

Promotion de l'industrie et des carrières

R-4 Que le conseil d'agglomération reconnaisse Montréal comme centre de créativité et d'innovation dans le domaine des TIC et qu'il mandate le Service de la mise en valeur du territoire et du patrimoine :

- pour qu'il contribue à la promotion internationale du savoir-faire montréalais dans les TIC;

- pour qu’il mette en place des mesures pour appuyer et promouvoir le secteur des TIC sur son territoire, en appui à TechnoMontréal et aux partenaires de la Grappe;
- pour qu’il encourage la création d’événements et d’activités et appuie les divers événements existants qui font la promotion de l’industrie des TIC et des carrières qu’elle propose, comme le Festival Euréka ou la Boule de cristal du CRIM.

Constat général émergent de l’analyse du secteur industriel visé par le projet

Il y a plus de 20 ans, M. Claude Y. Laporte, alors Professeur au Collège Militaire Royal de Saint-Jean, avait proposé alors la création d’un centre de génie logiciel à Montréal. Une étude avait dégagé les constats suivants (CRIM 1992) :

- La croissance accélérée de la demande
- La productivité et la qualité insatisfaisantes des logiciels
- La difficulté de maîtriser le génie logiciel
- La pénurie de personnel qualifié
- Un effort majeur de développement à l’échelle mondiale

Fort est de constater, qu’une étude du même type dégagerait aujourd’hui des constats similaires. Ce qui explique, en grande partie, cette situation, c’est que la taille et la complexité des systèmes n’ont cessé de croître à un taux constant. Par exemple, on développe en ce moment, dans tous les coins du monde, des systèmes comportant de 10 à 100 millions de lignes de code.

Initiative internationale pour répondre aux besoins des TPOs

Un groupe de travail de l’ISO déjà à l’œuvre depuis 2005

En 2005, l’Institut de Standardisation Industrielle de Thaïlande (TASI) a défrayé à Bangkok, deux réunions, d’un groupe de travail spécial auquel le Professeur Claude Y. Laporte, de l’ÉTS, participait. L’objectif était de définir le mandat d’un futur groupe ISO qui développerait une norme internationale répondant aux besoins des TPOs. Ce mandat a été approuvé par 12 pays lors d’un vote piloté par l’ISO. En résultat de ce vote, le nouveau groupe de travail 24 (*Working Group 24*) a été établi. Le professeur Claude Y Laporte de l’ÉTS a été nommé, par l’ISO, l’éditeur de ce groupe. Les pays suivants marquèrent leur engagement à participer au nouveau groupe de travail : Afrique du Sud, Belgique, Canada, Corée, É.-U, Grande-Bretagne, Irlande, Italie, Japon, Luxembourg, République Tchèque et Thaïlande. Depuis 2005, d’autres pays ont décidé de participer au groupe 24: la Colombie, l’Inde et le Mexique.

Mandat du groupe de travail 24 de l’ISO

- offrir aux TPOs le moyen d’être reconnues comme producteurs de systèmes logiciels de qualité sans le coût initial pour mettre en œuvre et maintenir l’utilisation d’un ensemble complet et coûteux de normes de génie logiciel et d’ingénierie de systèmes ainsi que pour effectuer des évaluations complètes;
- produire des guides faciles à comprendre, économiques et utilisables par des TPOs;
- produire un ensemble de profils qui s’appuient sur les processus existants d’une TPOs, ou les améliorent, ou qui fournissent un guide pour établir de tels processus;
- s’adresser aux besoins du marché des TPOs en permettant des profils et niveaux spécifiques de domaines;
- fournir des exemples de façon à inciter les TPOs à adopter et à suivre les processus qui conduisent à un logiciel de qualité et répondent aux besoins, problèmes et risques liés à leur domaine;
- fournir une base de départ quant à comment plusieurs TPOs peuvent collaborer ou être évaluées en tant qu’équipe de projet pour des projets pouvant être plus complexes que ceux pouvant être conduits par une seule TPOs;
- développer des profils et guides, pouvant être mis à l’échelle, de façon à ce que la conformité aux normes ISO/CEI 12207 et/ou ISO 9001 et l’évaluation deviennent possibles avec un minimum de remaniement des processus des TPOs.

Pour atteindre ces objectifs, le groupe a développé un ensemble de normes et de rapports techniques. Ces documents ont été soumis pour révision et approbation aux pays membres de l'ISO. Ils seront publiés par l'ISO en 2010. Même si le mandat du groupe 24 vise les très petites organisations, c.à.d. les entreprises, les départements ou projets de 25 personnes et moins, le groupe de travail de l'ISO a décidé, dans un premier temps, de concentrer ses efforts sur les très petites entreprises car les travaux de normalisation effectués pour celles-ci s'appliqueront en grande partie aux très petits départements ou très petits projets.

Normes internationales pour les TPOs élaborées par le groupe de l'ISO

Le groupe de travail 24 de l'ISO a élaboré, dans un premier temps, un ensemble de documents de normalisation pour les TPOs qui développent du logiciel générique (*generic software*). Plus tard, le groupe développera des documents de normalisation pour les développeurs qui œuvrent dans des domaines spécifiques tels que les applications critiques (*critical software*).

Le groupe de travail a utilisé une approche innovatrice pour le développement de la norme pour les TPOs. Étant donné que les normes existantes sont inadéquates, il a été décidé d'extraire, des normes existantes, les parties jugées importantes et adéquates pour les TPOs. Cette approche produit, ce qui appelé par l'ISO, un 'profil'. Un profil est comme une matrice qui indique, pour une norme spécifique, ce qui est utilisé et ce qui ne l'est pas pour produire le document formel ISO appelé '*International Standardized Profile'* (ISP).

Pour guider les TPOs à mettre en œuvre des pratiques de génie logiciel adaptées à leurs besoins et à leurs croissances, le groupe a développé un parcours (*roadmap*) composé de 4 étapes ou profils :

- L'étape 1 s'adresse aux TPOs en démarrage (*Start-up*) et aux TPOs qui développent des projets de 6 personnes-mois ou moins;
- L'étape 2 s'adresse aux TPOs qui exécutent un seul projet logiciel à la fois;
- L'étape 3 s'adresse aux TPOs qui exécutent plusieurs projets logiciels à la fois;
- L'étape 4 s'adresse aux TPOs qui veulent améliorer significativement la gestion de leurs affaires (*Business Management, Portfolio Management*).

Chaque document, développé par le groupe de travail 24, vise une clientèle spécifique. La figure 5 illustre les documents développés à ce jour:

- Le document intitulé *Overview,* est un rapport technique ISO, qui présente les concepts nécessaires à la compréhension de l'ensemble des documents relié à la norme c.-à-d. les documents intitulés *Framework and Taxonomy* et *Specification of Profile* de l'ensemble, et les rapports techniques intitulés *Assessment Guide* et *Management and Engineering Guide*. Le document *Overview* est destiné à une clientèle qui désire comprendre la norme: les TPOs, les donneurs d'ordre, les évaluateurs, les auteurs d'un profil, les développeurs d'outils, les consultants, etc.;
- Le document intitulé *Framework and Taxonomy* est une norme ISO. Ce document explique le concept et la structure des profils ainsi que la terminologie spécifique à cette norme;
- Le document intitulé *Assessment Guide* est un rapport technique ISO. Il donne des guides pour développer une méthode d'évaluation ou pour effectuer l'évaluation d'un profil;
- Le document intitulé *Specification of Profile* est une norme ISO. Il énumère les éléments des normes, telles que la ISO/CEI 12207 (ISO 2008a) et la ISO/CEI 15289 (ISO 2006), qui sont utilisés pour la description d'un profil spécifique;
- Le document intitulé *Management and Engineering Guide* est un rapport technique ISO. Il donne des guides d'application en matière de gestion et d'ingénierie pour un profil spécifique.

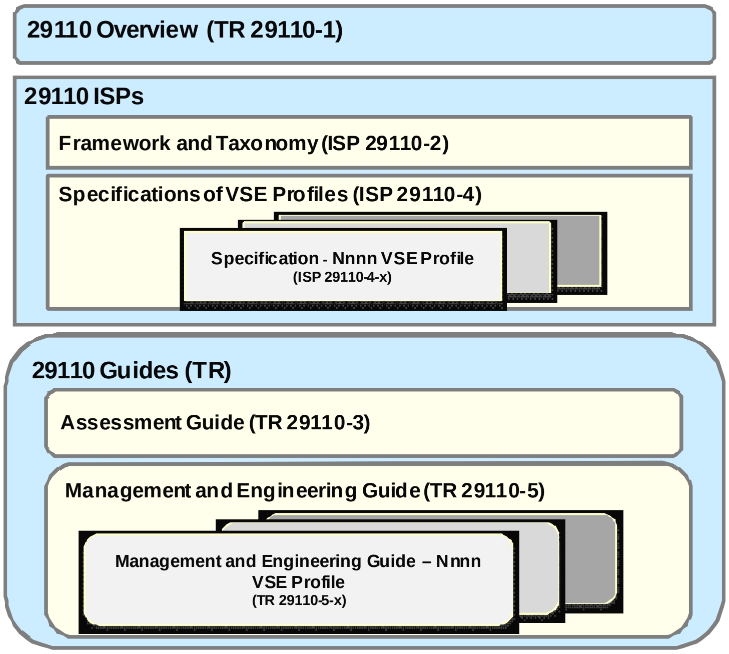

29110 Overview (TR 29110-1)
29110 ISPs
Framework and Taxonomy (ISP 29110-2)
Specifications of VSE Profiles (ISP 29110-4)
Specification - Nnnn VSE Profile
(ISP 29110-4-x)
29110 Guides (TR)
Assessment Guide (TR 29110-3)
Management and Engineering Guide (TR 29110-5)
Management and Engineering Guide – Nnnn VSE Profile
(TR 29110-5-x)

Figure 5 Ensemble des documents ISO (ISO 2009)

Étant donné que les documents de l'ISO sont normalement vendus à un coût d'environ 150$ chacun, le groupe de travail 24 a demandé à l'ISO de rendre les rapports techniques, décrits ci-dessus, **gratuits** afin que ces documents soient plus accessibles aux TPOs.

Même si le groupe de travail a produit un guide de gestion et d'ingénierie, la plupart des TPOs ne possèdent pas l'expertise pour transformer ce guide en un processus utilisable et utile. Le Professeur Laporte a donc proposé aux délégués du groupe 24, lors de la réunion à Moscou en 2007, le développement de matériel utilisable 'tel quel' par les TPOs. Ce matériel est appelé trousse de déploiement.

Trousses de déploiement des normes adaptées aux TPOs

Une trousse de déploiement (TD) est un ensemble d'artefacts pour faciliter et accélérer l'implantation de la norme ISO dans les TPOs en leur donnant des processus prêts à être utilisés: p. ex. processus documentés comportant les activités, rôles, intrants, extrants, liste de vérification, gabarits, exemples et outils de support. Les éléments typiques d'une trousse de déploiement sont énumérés à la figure 6.

1. Introduction
   Purpose of this document
   Key Definitions
2. Why this Topic is important
3. Overview of Main Tasks
   3.1 Tasks
   3.2 Roles and artefacts
   3.3 Activity Lifecycle and examples of lifecycles

Annex A Templates
Annex B Checklists
Annex C Coverage Matrices (ISO 29110 Part 5, ISO 12207, ISO 9001, CMMI)
Annex D Tools
Annex E Deployment Package Evaluation Form

Figure 6 Table des matières des trousses de déploiement (Laporte et al. 2008b)

Pour le premier profil, dont la norme sera publiée officiellement par l'ISO en 2010, les membres du groupe de travail ont élaboré bénévolement les trousses de déploiement énumérées au tableau 5.

Nom de la trousse de déploiement	Pays responsable
Analyse des exigences	Belgique, Canada
Architecture et Conception détaillée	Canada
Construction (c.-à-d. codage et tests unitaires)	Mexique
Intégration et tests	Colombie
Vérification et validation	Pérou
Gestion des versions	Thaïlande
Gestion de projets	Irlande
Livraison du produit	Thaïlande
Auto-évaluation	Finlande
Conduite de projets pilotes	Canada, Uruguay

Tableau 5 Liste des trousses de déploiement

Ces trousses de déploiement forment un ensemble cohérent qui permet la mise en place de la norme, pièce par pièce, tel qu'illustré à la figure 7, afin de répondre aux besoins des TPOs et à leurs capacités spécifiques à implémenter et utiliser de nouvelles pratiques. Par exemple, une TPO qui éprouve des difficultés à gérer les versions des documents et logiciels mettra en application les pratiques de la trousse '*version control*'. En ce moment, les trousses ne sont disponibles qu'en anglais puisqu'elles sont développées et révisées par les membres du groupe 24. Par la suite, elles seront traduites, par le délégué d'un pays, pour satisfaire les besoins des TPOs de son pays.

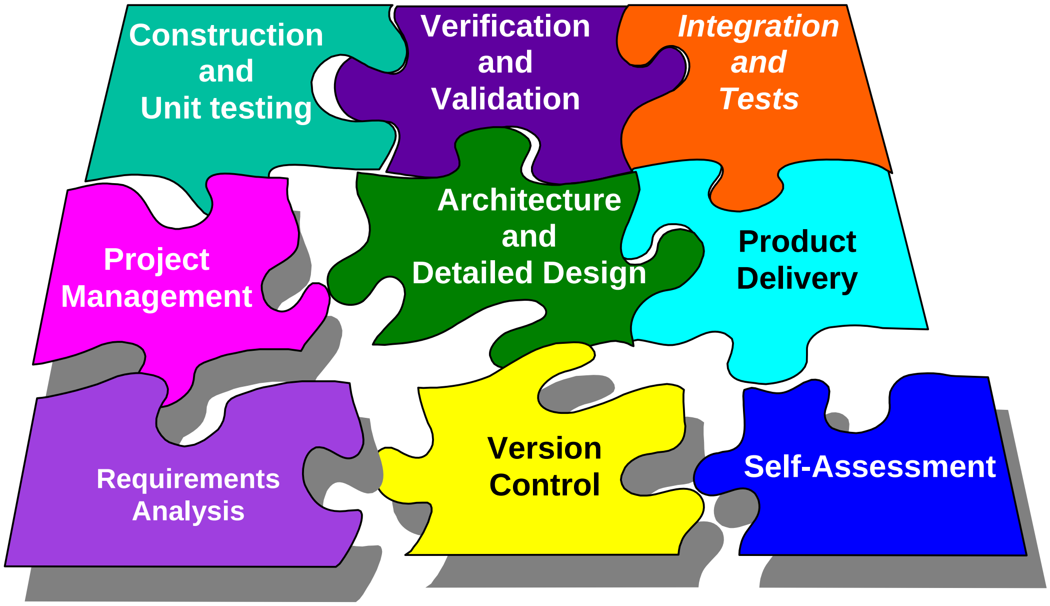


Figure 7 Ensemble de trousses de déploiement pour un profil (Laporte 2009)

Afin de s'assurer que les normes, documents techniques et les trousses de déploiement satisferont les besoins des TPOs, des projets pilotes seront effectués en 2009. Comme plusieurs membres du groupe 24 travaillent dans des centres de transfert technologique, il a été proposé de mettre sur pied un réseau de collaboration entre ces centres. Chaque centre a la responsabilité de déployer les trousses et d'offrir d'autres produits et services compte tenu des particularités nationales.

Le réseau international de centres de support aux TPOs

Le professeur Laporte, lors de la réunion du groupe de travail de l'ISO à Mexico en 2008, a proposé la création d'un réseau international de support aux TPOs. Les principaux objectifs du réseau sont d'accélérer le déploiement de la norme de l'ISO et des guides pour les TPOs et d'accélérer le développement et l'application de guides et des trousses de déploiement. Un document décrivant les modalités de collaboration a été rédigé et signé entre chacun des membres du réseau et l'ÉTS (voir annexe B). La figure 8 montre le réseau des collaborateurs actuels. D'autres pays ont manifesté un intérêt à contribuer aux travaux de ce réseau particulier dont les produits et services sont conçus spécifiquement pour les besoins des TPOs.

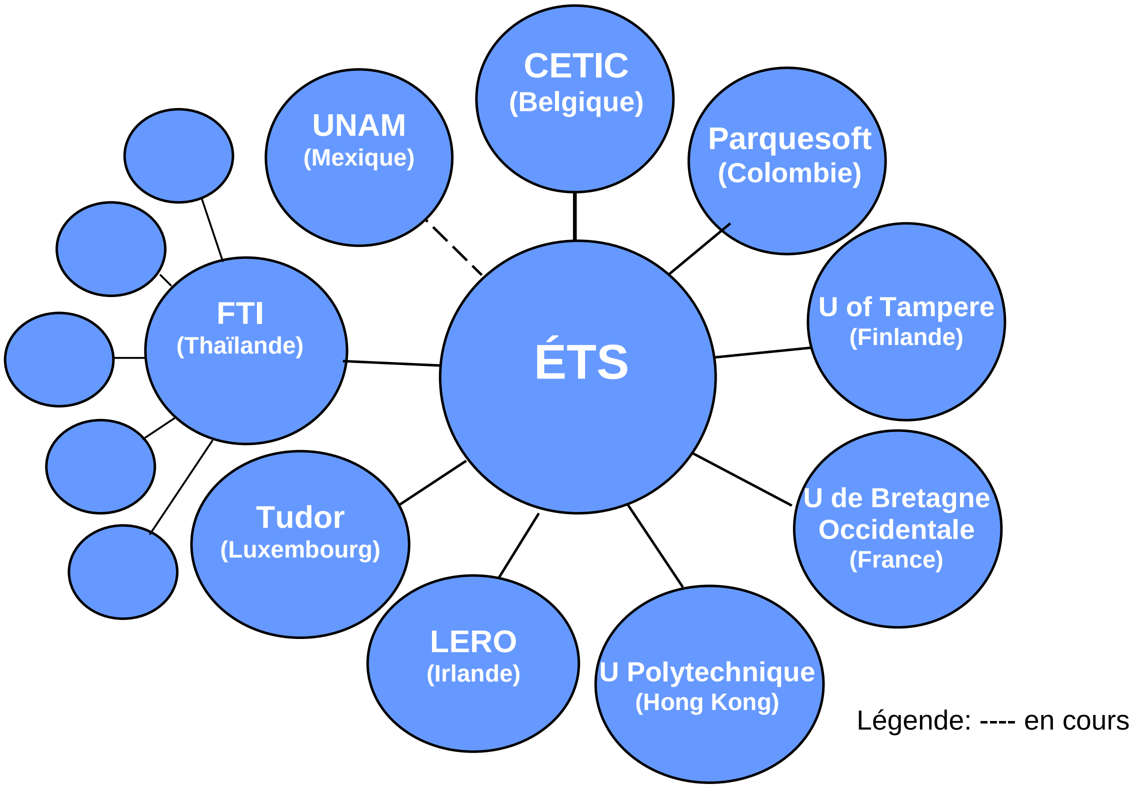


Figure 8 Réseau des centres de support aux TPOs (Laporte 2009)

Les participants au réseau sont[3] :

- Belgique - Centre d'Excellence en Technologies de l'Information et de la Communication (CETIC)
- Colombie - Parquesoft Foundation
- Finlande - Université de technologie de Tampere, Pori
- France - Université de Bretagne Occidentale
- Hong Kong - Université Polytechnique
- Irlande - Lero, The Irish Software Engineering Research Centre
- Luxembourg - Centre de Recherche Public Henri Tudor
- Thaïlande – Federation of Thai Industries

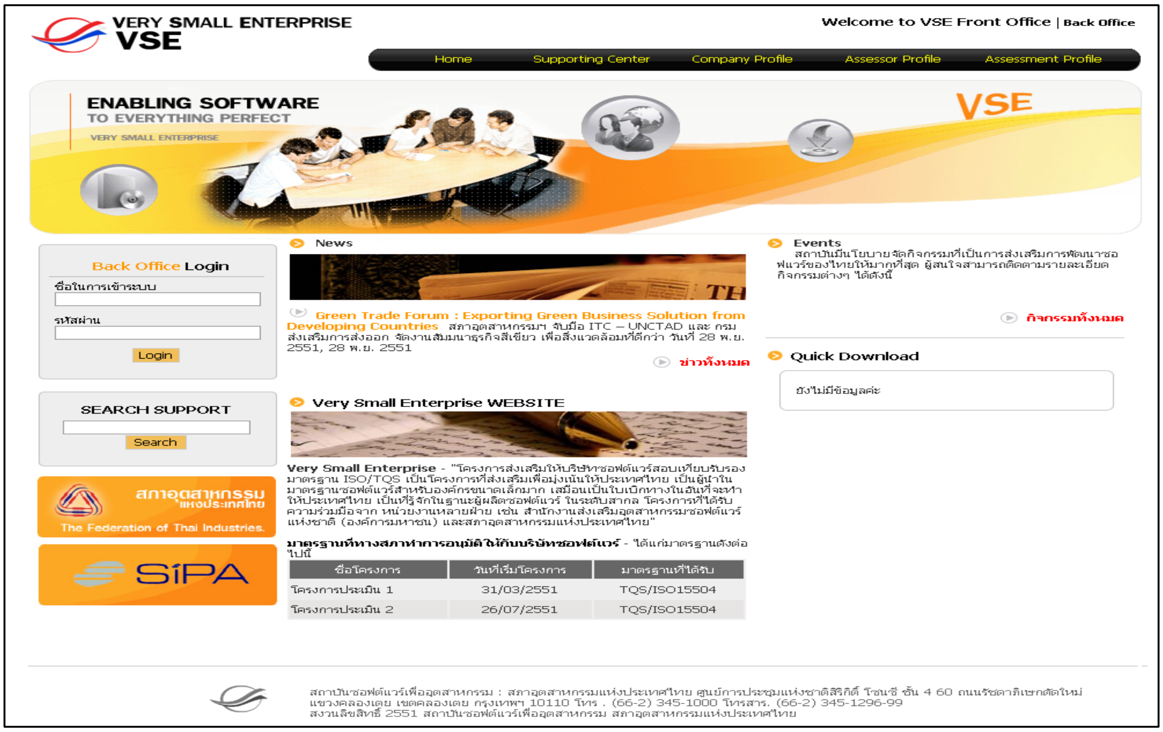
La Thaïlande annoncé, lors de la réunion du groupe 24 en Inde en mai 2009, un réseau de collaboration, dont elle est le chef de file (*regional hub*), composé des 10 pays de l'organisation ASEAN[4]: Thaïlande (2,000[5]), Cambodge (100), Laos (200), Myanmar (200), Vietnam (500), Malaisie (1200), Singapour (1000), Indonésie (1500), Philippines (2500) et Brunei (500). La Thaïlande a déjà mis en ligne, tel qu'illustré à la figure 9, son site de support aux TPOs.

L'Universidad Nacional Autónoma de México (UNAM) a aussi manifesté un intérêt à devenir membre du réseau. D'autres centres de transfert ainsi que d'autres universités seront contactés et invités à se joindre au réseau en 2009 (p.ex. Corée, Japon, Chine, Afrique du Sud).

[3] Une courte description des participants au réseau se trouve à l'annexe A.

[4] Asia South East Countries

[5] Le nombre entre parenthèses indique le nombre approximatif de TPOs visées dans ce pays.



Green Trade Forum : Exporting Green Business Solution from Developing Countries สภาอุตสาหกรรมฯ จับมือ ITC – UNCTAD และ กรมส่งเสริมการส่งออก จัดงานสัมมนาธุรกิจสีเขียว เพื่อสิ่งแวดล้อมที่ดีกว่า วันที่ 28 พ.ย. 2551, 28 พ.ย. 2551

สถาบันมีนโยบายจัดกิจกรรมที่เป็นการส่งเสริมการพัฒนาซอฟแวร์ของไทยให้มากที่สุด ผู้สนใจสามารถติดตามรายละเอียดกิจกรรมต่างๆ ได้ดังนี้

ยังไม่มีข้อมูลค่ะ

Very Small Enterprise WEBSITE

Very Small Enterprise - "โครงการส่งเสริมให้บริษัทซอฟต์แวร์สอบเทียบรับรองมาตรฐาน ISO/TQS เป็นโครงการที่ส่งเสริมเพื่อมุ่งเน้นให้ประเทศไทย เป็นผู้นำในมาตรฐานซอฟต์แวร์สำหรับองค์กรขนาดเล็กมาก เสมือนเป็นใบเบิกทางในอันที่จะทำให้ประเทศไทย เป็นที่รู้จักในฐานะผู้ผลิตซอฟต์แวร์ ในระดับสากล โครงการที่ได้รับความร่วมมือจาก หน่วยงานหลายฝ่าย เช่น สำนักงานส่งเสริมอุตสาหกรรมซอฟต์แวร์แห่งชาติ (องค์การมหาชน) และสภาอุตสาหกรรมแห่งประเทศไทย"

มาตรฐานที่ทางสภาทำการอนุมัติให้กับบริษัทซอฟต์แวร์ - ได้แก่มาตรฐานดังต่อไปนี้

ชื่อโครงการ	วันที่เริ่มโครงการ	มาตรฐานที่ได้รับ
โครงการประเมิน 1	31/03/2551	TQS/ISO15504
โครงการประเมิน 2	26/07/2551	TQS/ISO15504

สถาบันซอฟต์แวร์เพื่ออุตสาหกรรม : สภาอุตสาหกรรมแห่งประเทศไทย ศูนย์การประชุมแห่งชาติสิริกิติ์ โซนซี ชั้น 4 60 ถนนรัชดาภิเษกตัดใหม่ แขวงคลองเตย เขตคลองเตย กรุงเทพฯ 10110 โทร . (66-2) 345-1000 โทรสาร. (66-2) 345-1296-99
สงวนลิขสิทธิ์ 2551 สถาบันซอฟต์แวร์เพื่ออุตสาหกรรม สภาอุตสาหกรรมแห่งประเทศไทย

Figure 9 Page d’accueil du site de support aux TPOs de la Thaïlande

Réseau international d’éducateurs

Lors de la réunion du groupe de travail de l’ISO en mai 2009 en Inde, le professeur Laporte a proposé aux 21 délégués, représentants des 11 pays présents, la mise sur pied d’un groupe d’intérêt sur l’éducation (*Education Interest Group*).

L'objectif de ce groupe est de développer un ensemble de cours pour des étudiants de premier et second cycle en informatique ou en génie logiciel/informatique de telle sorte qu’ils apprennent et appliquent les normes ISO pour les TPOs pendant qu’ils sont en classe plutôt que d’attendre qu’ils soient en industrie. Les cours développés seront, pour les universités, similaires aux trousses de déploiement développées pour les TPOs. L'objectif est de faciliter et d’accélérer l'enseignement des nouvelles normes ISO pour les TPOs par les universités en leurs fournissant des trousses d’enseignement comportant le plan de cours et du matériel pédagogique, tels que : du matériel de présentation, des exercices, des études de cas, le matériel de lecture, etc.

Six cours ont été proposés aux délégués:

- Cours 1 - Introduction aux normes ISO/CEI en génie logiciel;
- Cours 2 - Introduction aux normes, rapports techniques ISO/CEI 29110 et aux trousses de déploiement pour les TPOs;
- Cours 3 - Développement d'un processus d'ingénierie de logiciels (*Process Engineering*) utilisant le Rapport Technique ISO/CEI 29110 Partie 5 – Guide d’Ingénierie et de Gestion
- Cours 4 - Développement de logiciels utilisant le Rapport Technique ISO/CEI 29110 Partie 5 – Guide d’Ingénierie et de Gestion;
- Cours 5 - Évaluation de la conformité des processus de développement de logiciels à la norme ISO/CEI 29110;
- Course 6 – Conduite de projet pilotes pour implanter la norme ISO/ CEI 29110 dans une TPO.

Le professeur Laporte a développé un gabarit de trousse d'enseignement qui sera utilisé pour développer et documenter un cours. Les délégués de quatre pays ont offert de développer les cours suivants:

- Irlande- Introduction aux normes ISO/CEI en génie logiciel;
- Canada - Introduction aux normes, rapports techniques ISO/CEI 29110 et aux trousses de déploiement pour les TPOs;
- République Tchèque - Développement de logiciels utilisant le Rapport Technique ISO/CEI 29110 Partie 5 – Guide d'Ingénierie et de Gestion;
- Thaïlande - Conduite de projet pilotes pour implanter la norme ISO/ CEI 29110 dans une TPO.

Lors de la prochaine réunion du groupe de l'ISO, au Pérou en novembre 2009, les cours seront présentés aux délégués. Ils seront ensuite disponible, gratuitement sur Internet, aux professeurs des universités et des Cégeps.

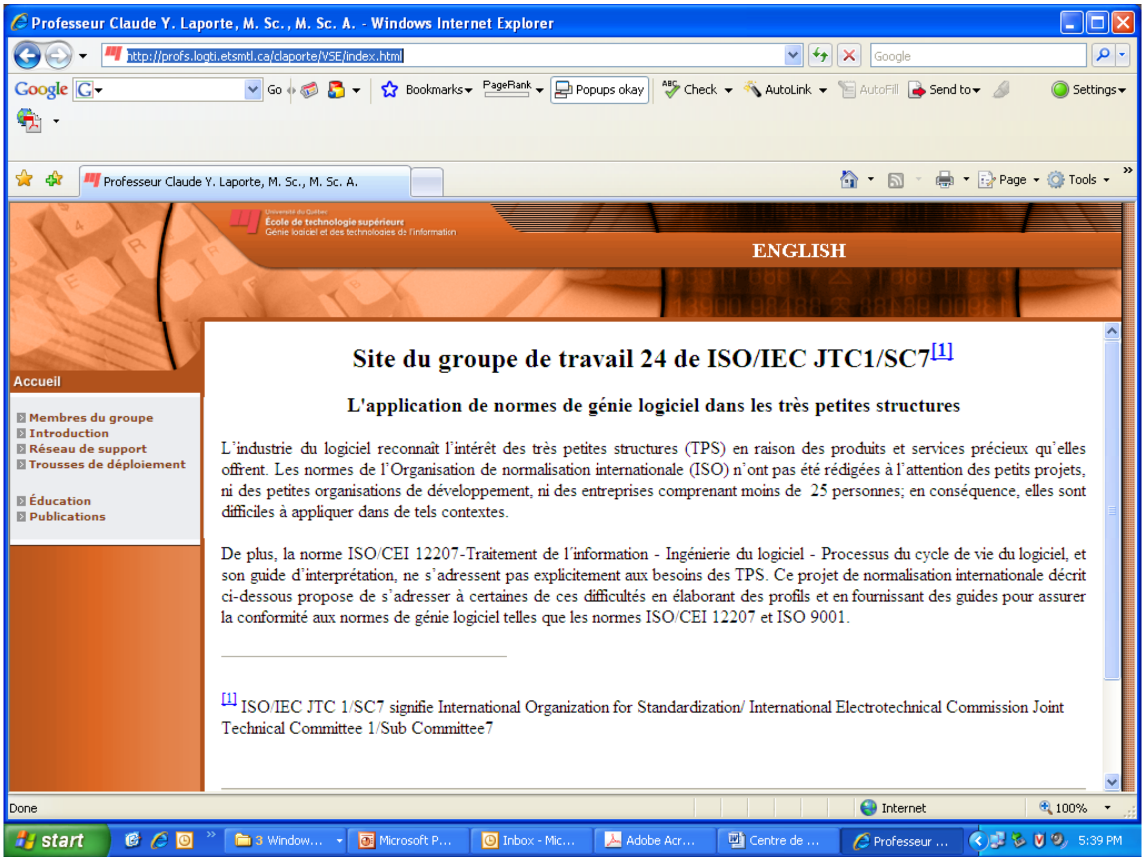
Site Internet des travaux du groupe 24 de l'ISO

Afin de publiciser, à l'extérieur du groupe de travail 24 de l'ISO, le travail effectué par celui-ci, le professeur Laporte a développé un site Internet en français[6] et un en anglais[7]. Ce site est hébergé à l'ÉTS et est une extension du site professionnel du professeur. La figure 10 illustre la page d'accueil du site français.

Ce site comporte l'information suivante :

- Une page d'accueil expliquant le mandat du groupe ISO;
- Une page qui énumère les membres du groupe de travail afin de bien illustrer la participation de plusieurs pays au groupe ISO;
- Une page qui décrit le réseau de support aux TPOs;
- Une page qui décrit les trousses de déploiement et qui permet de télécharger ces trousses (les trousses ne sont disponibles qu'en anglais présentement);
- Une page qui énumère les publications et communications effectuées par les membres du groupe ISO. La plupart des publications peuvent être téléchargées.

[6] http://profs.logti.etsmtl.ca/claporte/VSE/index.html
[7] http://profs.logti.etsmtl.ca/claporte/English/VSE/index.html



Site du groupe de travail 24 de ISO/IEC JTC1/SC7[1]

L'application de normes de génie logiciel dans les très petites structures

L'industrie du logiciel reconnaît l'intérêt des très petites structures (TPS) en raison des produits et services précieux qu'elles offrent. Les normes de l'Organisation de normalisation internationale (ISO) n'ont pas été rédigées à l'attention des petits projets, ni des petites organisations de développement, ni des entreprises comprenant moins de 25 personnes; en conséquence, elles sont difficiles à appliquer dans de tels contextes.

De plus, la norme ISO/CEI 12207-Traitement de l'information - Ingénierie du logiciel - Processus du cycle de vie du logiciel, et son guide d'interprétation, ne s'adressent pas explicitement aux besoins des TPS. Ce projet de normalisation internationale décrit ci-dessous propose de s'adresser à certaines de ces difficultés en élaborant des profils et en fournissant des guides pour assurer la conformité aux normes de génie logiciel telles que les normes ISO/CEI 12207 et ISO 9001.

[1] ISO/IEC JTC 1/SC7 signifie International Organization for Standardization/ International Electrotechnical Commission Joint Technical Committee 1/Sub Committee7

Figure 10 Page d'accueil du site de support aux TPOs du professeur Laporte

Depuis décembre 2008, la fréquentation du site a été mesurée. Le tableau 6 illustre que le site est fréquenté à chaque mois, par environ 200 visiteurs de 30 pays. Aussi, environ 450 pages du site pour les TPOs sont consultées à chaque mois.

Mois	Nombre de visiteurs	Nombre de pays	Nombre de pages accédés
Décembre 2008	180	38	575
Janvier 2009	269	33	521
Février 2009	252	35	482
Mars 2009	197	36	489
Avril 2009	171	30	433
Mai 2009	203	28	452

Tableau 6 Fréquentation du site internet non officiel du groupe 24

La fréquentation du site devrait augmenter après la publication des 9 trousses de déploiement à l'été 2009, des études de cas et des modules de formation à l'automne 2009 ainsi que des normes et rapports techniques ISO/CEI 29110 en 2010.

Identification des Forces et Faiblesses et Opportunités et Menaces des TPOs de la région de Montréal

Forces

- Expertise et expérience reconnues
- Masse critique d'entreprises et de clients
- Intégrées à une structure industrielle très diversifiée dont les besoins en produits et services TI sont en forte croissance

Faiblesses

- Absence de *roadmap* (parcours) pour les TPOs
- Absence de *Label-qualité* (p.ex. ISO 9000)
- Processus de gestion de la qualité peu implanté
- Ressources humaines et financières limitées pour implanter des pratiques logicielles éprouvées

Opportunités

- L'accroissement de la compétitivité
- Une plus grande confiance de la clientèle
- Une plus grande qualité des produits logiciels
- Une augmentation de l'engagement de la gestion de la TPOs pour l'amélioration des processus
- Une diminution des risques de développement
- Une mise sur le marché facilitée par une meilleure image
- Un meilleur potentiel à l'exportation
- Secteur public et privé en recherche de solutions TI et de ressources humaines qualifiées
- Secteur industriel dans lequel sont intégrées les TPOs en forte croissance
- Création de la richesse dans un secteur stratégique de notre économie nationale et globale
- Organismes structurants en TIC actifs et très au fait du contexte et des enjeux des TPOs: TechnoMontréal, Technocompétences, Coalition Bell, Chantier Main-d'œuvre en TI de la grande région de Montréal, entre autres
- Amélioration de la performance des processus de développement de logiciels
- Offrir des produits logiciels respectant le cahier de charge, le budget et le calendrier d'un client local ou d'un manufacturier
- Accéder plus facilement aux pratiques reconnues de développement afin de demeurer plus compétitives et que la satisfaction de leurs clients en soit rehaussée

Le Programme d'aide à la recherche industrielle du CNRC[8]

Le Programme d'aide à la recherche industrielle du CNRC peut offrir un support financier pour des interventions (visite ou aide technique) auprès des PMEs variant de 5 h jusqu'à 10 h par intervention avec un taux horaire jusqu'à 120 $/h. Les interventions admissibles sont*;* élaborer une formulation détaillée de la problématique technologique, - évaluer l'état de la technologie et les potentiels de PI, - identifier les incertitudes technologiques, - identifier les pistes de solution, - valider certaines hypothèses via des tests simples, - identifier les expertises requises au développement, - faire un estimé préliminaire de l'échéancier et des coûts pour une prise de décision.-étude de faisabilité. L'intervenant doit fournir l'information suivante pour recevoir la subvention :

- Description de la problématique
- Liste des livrables

[8] Programme d'aide à la recherche industrielle du CNRC, présenté par M Olivier Thomas, 18 mars 2009.

- Nombre d’heures requis pour l’intervention
- Échéancier de l’intervention

Les critères d’admissibilité à l’aide financière du Programme d’aide à la recherche industrielle du CNRC sont:

- Entité commerciale incorporée, à but lucratif;
- De 500 employés et moins;
- Le projet vise à développer et commercialiser des technologies pour le plus grand avantage du Canada;
- Prête à accepter les conditions de contribution incluant les clauses sur les bénéfices pour le Canada;
- Avec un plan d’affaires cohérent démontrant les compétences requises en mise en marché, en technologie et en gestion, ainsi qu’une capacité financière adéquate pour entreprendre le projet et en exploiter les résultats.

Les critères financiers du volet PARI-CNRC sont:

- Contribution financière non remboursable par l’entreprise (c.-à-d. une subvention et non un prêt).
- Maximum de 75 % du coût total du projet pour les activités admissibles.
- Contribution maximale de 1,000,000$ par projet par année, répartie de la façon suivante:
    - Jusqu’à 100 % des salaires du personnel technique interne
    - Jusqu’à 75 % des frais de main-d’œuvre de sous-traitance technique
    - Aucune contribution sur les frais généraux, l’achat d’équipement ou de matériel
- Contribution minimale de l’entreprise : 25 %

Le volet Stratégie emploi jeunesse du CNRC

Les critères d’admissibilité sont:

- 30 ans ou moins.
- Détenir un diplôme d’études post-secondaire (Cégep, université).
- Citoyen canadien ou ayant le droit de travailler au Canada,
- Disciplines techniques ou autres.
- N’a jamais bénéficié de la Stratégie emploi jeunesse.

Les critères financiers sont:

- Couvre une partie du salaire pendant 6 à 12 mois pour réaliser un projet avec des objectifs mesurables.
- Contribution maximale de 15 000 $ (pourrait être bonifié).
- Financement du ministère des Ressources humaines et du développement des compétences (RHDC).

Menaces

- Standards de l’ISO pas exigés par les donneurs d’ordre dans un marché global d’offre et de demande. Des pays ont déjà élaboré un plan d’implantation de la norme ISO 29110.
- Pertes du nombre de contrats
- Compétitivité croissante du marché international, surtout des TPOs des pays en émergence,
- Déplacement des projets et de la main-d’œuvre qualifiée hors du Québec et du Canada au profit d’entreprises plus efficaces et efficientes à l’étranger.

Identification des Forces et Faiblesses et Opportunités et Menaces du Centre de support aux TPOs

Forces

- L'Éditeur du groupe de travail 24 de l'ISO : Claude Y. Laporte, professeur à l'ÉTS
- Les leaders des Centres de support aux TPOs
- Valeur intangible du réseau et de la contribution des leaders des centres de support aux TPOs
- La publication en 2010 des premières normes ISO sur les TPOs
- L'adaptation des normes ISO aux besoins des TPOs
- Rayonnement international de l'ÉTS comme pivot du réseau des centres de support aux TPOs
- Les expertises du corps professoral du département de génie logiciel et des TI
- L'ÉTS est membre actif des comités de réflexion, d'orientation et de visibilité du Chantier main-d'œuvre des TIC coordonnés par TechnoMontréal
- Production des guides faciles à comprendre, économiques et utilisables par les TPOs
- Production d'un ensemble de profils (*roadmap*) qui s'appuient sur les processus existants d'une TPO, ou les améliorent, ou qui fournissent un guide pour établir de tels processus.
- Production d'exemples, des façons de faire pour inciter les TPOs à adopter et à suivre les processus qui conduisent à un logiciel de qualité et répondent aux besoins, problèmes et risques liés à leur domaine.
- Développement de profils et guides, pouvant être mis à l'échelle, de façon à ce que la conformité aux normes ISO/CEI 12207 et/ou ISO 9001 et l'évaluation deviennent possibles avec un minimum de remaniement des processus des TPOs.
- Le département de génie logiciel et des TI a déjà tissé de bons liens avec les professeurs en informatique des Cégeps du Québec. Ces professeurs pourraient agir comme partenaires de l'ÉTS, enseigner les nouvelles normes ISO pour TPOs et agir localement auprès de leurs TPOs.
- Plusieurs professeurs en génie logiciel, collaborateurs dans des groupes ISO, ont une vaste expérience industrielle.
- Offre de cours du second cycle en génie logiciel dont des éléments de contenu couvrent la future norme ISO pour les TPOs.
- Des projets auprès de TPOs, avec des étudiants de second cycle, sont déjà réalisés et d'autres projets sont en cours de réalisation (voir annexe C).

Faiblesses

- Projet piloté par une seule personne jusqu'à maintenant à l'ÉTS
- Budget départemental limité pour effectuer des projets de transfert auprès des TPOs
- Les activités de transfert technologique pourraient ne pas attirer des professeurs puisque celles-ci ne permettent pas de publier dans des revues de prestige du génie logiciel.
- Les fonds de démarrage d'un tel centre et le maintien d'une structure d'offre de produits et services sur une période raisonnable pour mesurer les bénéfices d'une telle opération d'amélioration des processus dans un secteur donné.

Opportunités

- Devenir le centre de référence canadien pour les TPOs comme le recommandait le Conseil canadien des normes :
    - *Strategy for the educational sector : Foster the use of standards in curricula across Canadian universities and follow-up on outcomes of the Academic Conference)* (SCC 2008)
- La future norme vise les TPOs du secteur logiciel.

- Il existe des normes internationales qui pourraient être utilisées pour offrir du soutien aux TPOs dans d'autres domaines tels que les services en TI (norme ISO/CEI 20000) et les TPOs impliquées en ingénierie de produits (norme ISO/CEI 15288).
    - Un projet de maîtrise, en cours de rédaction, vise à établir un premier profil pour l'utilisation de la norme ISI/IEC 20000 dans une TPO de Montréal.
    - Le professeur Laporte est le coprésident d'un nouveau groupe de travail visant à développer, comme le groupe 24, des profils et trousses, pour l'ingénierie de produits (*system engineering*)
- TechnoMontréal est partenaire avec l'ÉTS.
- Le Programme d'aide à la recherche industrielle du CNRC
    - Des professeurs du département de génie logiciel et des TI pourraient être rémunérés pour effectuer des interventions dans des TPOs.
    - Il serait possible, à l'aide de ce programme, d'embaucher quelques professeurs d'un Cégep pour effectuer des interventions. Lorsque plusieurs interventions auront été effectuées par un professeur, ils pourraient alors demander directement la subvention pour effectuer ses interventions. Le programme PARI peut aussi être utilisé pour effectuer des interventions de plus grandes envergures.
- Le volet Stratégie emploi jeunesse du CNRC permettrait, à un gradué de l'ÉTS, d'effectuer des interventions dans des TPOs.
- Possibilité de créer un centre de R & D pour TPOs en utilisant les mêmes leviers financiers que les grandes sociétés (création d'un comité scientifique) pour réduire les coûts de développement de leurs processus.
- Étant donné la pénurie d'emploi en TI, le gouvernement du Québec pourrait utiliser ce 'levier' pour augmenter la productivité et la qualité des TPOs du secteur logiciel.
- Des étudiants de troisième cycle, du Québec et de l'étranger, pourraient faire leurs recherches sur les TPOs à l'ÉTS.
- Un plus grand maillage avec les Cégeps et les TPOs québécoises pourraient faciliter le recrutement d'étudiants de premier cycle en LOG et TI.
- Un plus grand maillage avec les TPOs québécoises pourraient faciliter le recrutement d'étudiants de second cycle en LOG et TI.

Menaces

- Le CRIM, établi depuis plus de vingt ans, pourrait commencer à offrir des produits et services aux TPOs avec le budget et l'infrastructure existants.
- La publication de la norme, par l'ISO, pourrait être retardée.
- Les TPOs québécoises pourraient percevoir la future norme comme un mal non nécessaire et l'ignorer.
- Les donneurs d'ordres québécois ainsi que les organismes publics pourraient ne pas s'intéresser à la norme et ne pas l'imposer à leurs fournisseurs.

Recommandation des promoteurs du projet du Centre de support aux TPOs de l'ÉTS

Considérant

- La demande mondiale croissante en logiciels;
- les besoins des TPOs Québécoises pour demeurer compétitives et assurer leur développement;
- l'initiative de l'ÉTS de supporter la création d'un centre de support aux TPOs;
- l'offre de services des centres de support aux TPOs;
- la valeur et le potentiel que représentent le réseau des centres de support aux TPOs;
- l'Éditeur du groupe de travail 24 de l'ISO, M. Claude Y. Laporte est le maillon fort de la chaîne qui relie le futur centre de support aux TPOs de l'ÉTS au réseau des centres;

- le fait qu'aucun autre centre de support aux TPOs n'existe encore au Canada;

Il est recommandé de créer un centre de support aux TPOs à l'ÉTS.

Le Centre de support aux TPOs de l'ÉTS

Mission

La mission proposée pour ce centre serait d'accélérer les transferts technologiques vers les très petites et petites organisations du Québec qui développent des produits logiciels, des systèmes avec logiciels ou offrent des services en TI pour les rendre plus compétitives, tant au niveau national qu'international, en développant et déployant des pratiques de génie logiciel adaptées à leurs besoins.

Structure organisationnelle du centre de support aux TPOs

Claude Y. Laporte
Jean Belzile
Réseau des centres

Partenaires actuels

- Membres du réseau international de support aux TPOs (CETIC, LERO, SIPA, Finlande, UBO, etc.)
- TechnoMontréal
- Technocompétence

Partenaires potentiels

- CRIM
- Les professeurs d'informatique des Cégeps
- Industrie Canada
- MDEIE
- Bureaux de normalisation: Conseil canadien des normes (CCN) du Canada et celui du Québec.

Objectifs stratégiques

Pour réaliser cette mission, le centre aurait les objectifs stratégiques qui suivent :

- Identifier, promouvoir et diffuser les pratiques éprouvées en génie logiciel pour les TPOs;
- Accélérer le processus de transfert technologique en génie logiciel pour les TPOs;
- Fournir des renseignements techniques et stratégiques d'avant-garde aux décideurs des TPOs, aux donneurs d'ordre et aux organismes publics du Québec;
- Participer au développement de normes internationales pour les TPOs
- Promouvoir les normes internationales auprès des TPOs québécoises
- Promouvoir la recherche en matière d'ingénierie logicielle pour les TPOs
- Favoriser la formation et le développement de cours sur les normes pour les TPOs.

Le marché visé

Le marché visé par le centre de l'ÉTS est celui des très petites organisations (TPOs) québécoises comprenant les entreprises, les organismes publics, les départements et projets de moins de 25 personnes. Ces entités se retrouvent, selon l'Organisation de coopération et de développement économique (OCDE 1990), au sein trois grandes catégories suivantes :

- Les sociétés de logiciel (les producteurs de logiciels commerciaux ou des logiciels développés sur commande). Cette catégorie inclut aussi les sociétés-conseils;
- Les producteurs de systèmes à base de logiciels, comme par exemple, les télécommunications, l'aérospatial, le transport ferroviaire;

- Les sociétés qui développent, dans leurs départements, des logiciels qui serviront à supporter leurs affaires, telles que les banques, les compagnies d'assurances et les manufacturiers.

Ces TPOs québécoises représentent un bassin de clientèle potentielle. Le Grand Montréal regroupe, dans le secteur logiciel seulement, plus de 500 entreprises de moins de 25 employés (Gauthier 2004).

Services offerts

Les principaux services offerts par le centre de l'ÉTS appartiennent, tel qu'illustré au tableau 7, à quatre grandes catégories : les services relatifs aux normes, processus et outils pour les TPOs, la recherche, la formation et la consultation et la veille et l'éveil technologique.

1. **Normes, processus et outils pour les TPOs**
   - Développement de trousses de déploiement en support à la norme ISO 29110
   - Développement de trousses spécifiquement pour les TPOs en démarrage
   - Conduite de projets pilotes des trousses de déploiement en support à la norme ISO 29110 (Intervention)
   - Évaluation informelle des processus logiciels et amélioration des pratiques de génie logiciel (Intervention)
   - Évaluation formelle de la conformité des processus à la norme ISO 29110 (Intervention)

2. **Projets de recherche**
   - Projets de recherche sur les TPOs québécoises
     - o Encadrement des étudiants de maîtrise et de doctorat
   - Participation à des projets internationaux sur les TPOs
   - Mesure de l'adoption de la norme ISO 29110 et de son impact dans les TPOs Québécoises
   - Développement de profils pour des domaines spécifiques tels que:
     - o Industrie aérospatiale
     - o Industrie pharmaceutique
     - o Industrie du jeu

3. **Formation**
   - Développement de cours et d'ateliers
   - Développement de modules d'auto apprentissage (c.-à-d. *e-learning*)
   - Formation
     - o Gestionnaires de très petites entreprises
     - o Directeurs de très petits départements
     - o Gestionnaires de très petits projets
     - o Développeurs dans les TPOs
     - o Donneurs d'ordre
     - o Agences gouvernementales
     - o Organismes sans but lucratif (OSBL)
     - o Consultants
   - Formation des éducateurs (Université et Cégeps) québécois
   - Certification des professionnels en génie logiciel

4. **Veille et éveil technologique auprès des TPOs**
   - Représentation du Canada au groupe de travail ISO sur les TPOs
   - Encadrement d'un groupe d'intérêt pour les TPOs québécoises
   - Développement d'un site web de support à la documentation pour les TPOs

Tableau 7 Les principaux services offerts par le centre de l'ÉTS

Définition des services

Normes, processus et outils pour les TPOs

Tel que décrit plus haut, les TPOs n'ont pas l'expertise nécessaire pour développer et documenter, à partir des normes ISO, leurs processus de génie logiciel. Le centre de l'ÉTS développera, en collaboration avec les membres de son réseau international, les trousses en français, pour les profils et domaines d'applications de ses clients. Que ce soit des TPOs en voie de démarrage ou des TPOs qui œuvrent depuis quelques années.

Le centre pourra par la suite aider les TPOs à mettre en œuvre les trousses, à effectuer des projets pilotes, à effectuer des évaluations informelles de conformité à la norme 29110. Si la TPOs désire être évaluée formellement, une personne du centre, n'ayant pas contribué à ni à l'amélioration ni à l'évaluation informelle pourra effectuer cette évaluation.

Le centre pourra permettre à des TPOs de se qualifier pour des contrats d'industriels canadiens et étrangers ainsi que pour des organismes publics québécois.

Dans cette catégorie de services offerts, on y retrouvera les activités suivantes :

- Développement de trousses de déploiement en support à la norme ISO 29110;
- Développement de trousses spécifiquement pour les TPOs en démarrage;
- Conduite de projets pilotes des trousses de déploiement en support à la norme ISO 29110;
- Évaluation informelle et amélioration des pratiques de génie logiciel;
- Évaluation formelle de la conformité à la norme ISO 29110.

Méthode

Le centre devra développer les trousses en collaboration avec les membres du réseau, les traduire et les adapter au contexte québécois et les valider à l'aide de projets pilotes auprès de quelques TPOs. Le centre devra développer une méthode d'évaluation formelle et une méthode informelle. Du matériel de formation sera développé et des spécialistes seront formés à ces méthodes d'évaluation. Le centre développera les gabarits des rapports d'évaluation. Dans le cas des évaluations informelles, des pistes d'améliorations seront proposées à la TPOs évaluée. Des rencontres auront lieues avec la TPO pour identifier ses besoins, réaliser les évaluations, présenter les résultats de ses évaluations et réaliser des projets pilotes afin d'améliorer ses performances.

Organisation

- Ressources humaines :
  - Spécialistes en génie logiciel pour développer et mettre à l'essai, dans des projets pilotes, les trousses de déploiement
  - Spécialistes en génie logiciel pour réaliser des évaluations informelles et formelles
- Ressources matérielles :
  - Trousses de déploiement en français
  - Guide de conduite de projets pilotes
  - Guide d'évaluation informelle
  - Gabarit de plan d'amélioration des pratiques logicielles
  - Guide d'évaluation formelle
  - Matériel de formation
  - Équipement informatique et équipement pour présentations

Implantation

À court terme, le centre développera et offrira des trousses de déploiement et développera les méthodes d'évaluation. Les méthodes d'évaluations pourront être développées conjointement avec le CETIC de la Belgique. Le CETIC, membre du réseau de support aux TPOs, a déjà développé une méthode d'évaluation pour les TPOs. Au cours de la première année, le centre pourra effectuer quelques projets pilotes et quelques évaluations informelles. Un professeur de l'ÉTS, a manifesté un intérêt à implanter la norme dans son laboratoire de recherche. À moyen et long terme, des propositions d'amélioration des pratiques seront faites aux TPOs et des évaluations formelles seront effectuées.

À court terme, des professeurs des CEGEP, qui se joindront au centre de l'ÉTS, assisteront à des formations sur la norme et les guides ISO ainsi que sur les trousses de déploiement. Ils assisteront, dans un premier temps à titre d'observateurs, à des projets pilotes. À moyen et à long terme, ils pourront effectuer des évaluations informelles ainsi que des implantations, par le biais de projets pilotes, de la norme ISO.

Projets de recherche

Les TPOs, étant donné leurs très petites tailles, budgets et certaines expertises ne peuvent pas effectuer des projets de recherche pour améliorer la performance de leurs processus de développement logiciels. Le centre étant situé dans un contexte universitaire, il sera possible d'effectuer des projets de recherche ayant des impacts à moyen et à court terme pour les TPOs québécoises en plus de participer activement au développement de nouvelles normes pour les TPOs et à l'amélioration continue de la norme ISO qui sera publiée en 2010.

Dans cette catégorie de services offerts, on y retrouvera les activités suivantes :

- Projets de recherche sur les TPOs québécoises
  - Mesure de l'adoption de la norme et de son impact dans les TPOs
  - Mesure de l'impact de la norme dans les TPOs
- Participation à des projets de recherche internationaux sur les TPOs
- Encadrement d'étudiants internationaux de maîtrise et doctorat
  - Des ententes de collaboration, décrites plus haut, ont été signées avec des universités de la France, de la Finlande et de l'Irlande. D'autres ententes sont en cours de discussion avec d'autres universités, p.ex. l'UNAM et l'École Polytechnique de Hong Kong.
  - D'autres ententes ont été signées avec des centres affiliés à des universités telles que le CETIC (affilié à l'Université de Namur) et l'entente avec la Thaïlande (Université de Chiang Mai)
- Développement de profils pour des domaines spécifiques tels que:
  - Industrie aérospatiale
  - Industrie pharmaceutique
  - Industrie du jeu

  Ces profils pourront être soumis au groupe de travail ISO pour que celui-ci l'étudie et développe des nouvelles normes internationales.

Méthode

Les étudiants des cycles supérieurs participeront à des projets de 2 façons différentes :

- Dans le cadre de certains cours de maîtrise tels que MGL 950 et MGL 805, depuis plusieurs années les étudiants font des interventions dans des organisations. Dorénavant, les projets seront orientés vers les TPOs.
- Dans le cadre des projets de maîtrise (p.ex. MGL 940 (9 crédits), ETI 8020 (15 crédits)), les étudiants feront des projets de recherche ou d'application.
- Dans le cadre de projet de doctorat, les étudiants développeraient et valideraient de nouveaux profils. Ces profils seraient proposés, à titre de contribution du Canada, au groupe de travail 24 de l'ISO.

Les étudiants du premier cycle en LOG et TI pourraient participer à des projets de 2 façons différentes:

- Dans le cadre des projets synthèses
- Dans le cadre du dernier stage en entreprise

Le centre pourra expérimenter et valider certaines technologies avant de les présenter aux TPOs.

Organisation

- Ressources humaines
  - Professeurs en génie logiciel
  - Spécialistes en génie logiciel
- Ressources matérielles
  - Laboratoires
  - Matériel informatique

Implantation

À court terme, les étudiants des cours MGL 805, dès l'hiver 2010 et ceux du cours MGL 950, dès l'automne 2010 feront des interventions dans des TPOs. De plus, des projets d'application MGL 940 (9 crédits) et ETI 8020 (15 crédits) seront effectués pour mettre à l'essai et améliorer les trousses de déploiement dans des TPOs. À moyen terme, l'Université de Chiang Mai (Thaïlande) a indiqué qu'elle souhaiterait envoyer un ou des professeurs, en informatique et en génie logiciel, pour effectuer un stage doctoral en laboratoire à l'ÉTS dans le cadre de la norme ISO.

Formation

La formation est un élément clé au succès de l'implantation de la norme dans les TPOs. Par contre, étant donné que les TPOs, surtout les très petites organisations, n'ont pas beaucoup de temps et de budget à consacrer à la formation, on offrira de la formation pour répondre à ces contraintes. De plus, on développera des modules d'auto-apprentissage (*self-learning*).

La mondialisation croissante de l'industrie du logiciel suppose qu'un professionnel en ingénierie logiciel travaillera possiblement dans différents pays ou pour des organisations à l'extérieur de son pays au cours de sa carrière. Il est donc important de développer les moyens d'accroître la portabilité de la certification des professionnels dans ce domaine. Une nouvelle norme internationale ISO (ISO/IEC 24773: 2008, Software engineering — Certification of software engineering professionals — Comparison framework) répond aux besoins de plusieurs organisations multinationales ou de développeurs de logiciels pour un client étranger (y compris l'offshoring) qui exige des professionnels certifiés en génie logiciel. La norme ISO facilitera la portabilité des certifications de professionnels en génie logiciel entre différents pays. La norme ISO demande l'utilisation du document ISO intitulé le Guide du corpus des connaissances en génie logiciel (ISO 2005), communément appelé le SWEBOK (*Software Engineering Body of Knowledge*), pour la description des connaissances qui seront décrites dans le programme de certification. Étant donné que le développement du SWEBOK a été piloté par 2 professeurs de l'ÉTS (Abran, Bourque), l'ÉTS possède une grande expertise dans ce sujet.

Dans cette catégorie des services offerts, on y retrouvera les activités suivantes :

- Développement de cours et d'ateliers
- Développement de modules d'auto apprentissage (p.ex. *e-learning*)
- Formation ciblée:
    - Gestionnaires de très petites entreprises
    - Directeurs de très petits départements
    - Gestionnaires de très petits projets
    - Développeurs dans les TPOs
    - Donneurs d'ordre
    - Agences gouvernementales
    - Organismes sans but lucratif
- Formation des éducateurs (Université et Cégep) québécois de façon à faciliter l'enseignement de la norme ISO dans leurs cours et ainsi accélérer le déploiement dans les TPOs Québécoises.
- Certification de professionnels en génie logiciel.

Méthode

Le matériel didactique sera préparé par les spécialistes de l'ÉTS et ceux du réseau de collaborateurs. Plusieurs composantes seront initialement développées par les membres du réseau pour lesquels l'anglais est la langue de travail. Le matériel didactique sera traduit et offert en français.

Certains cours et ateliers seront offerts dans les locaux de l'ÉTS ou à tout autre endroit pertinent (p.ex. dans un Cégep). D'autres formations seront offertes, à distance, à l'aide des médias électroniques de l'ÉTS.

Étant donné que les ressources disponibles pour offrir de la formation sont limitées au Québec, on favorisera l'approche de former des formateurs (*Train-the-trainer*).

Les cours, ateliers et séminaires prévus ainsi que leur durée prévue :

- Survol des normes et rapport techniques ISO pour les TPOs, d’une durée d’environ 3 heures;
- Présentation détaillée des normes et rapport techniques ISO pour les TPOs, d’une durée d’environ 6 heures;
- Description détaillée d’une trousse de déploiement, d’une durée d’environ 3 heures;
    - Liste des trousses disponibles pour le profil de base :
        - Gestion de projets
        - Analyse des exigences
        - Conception
        - Construction (c.-à-d. codage et tests unitaires)
        - Intégration et tests
        - Gestion des versions
        - Vérification et validation
        - Livraison du produit
        - Auto-évaluation
        - Conduite de projets pilotes
    - Présentation du contenu de la trousse (activités, rôles, documents à produire, liste de vérification, outil)
    - Présentation d’un exemple d’application de la trousse
- Présentation détaillée de la méthodologie d’auto-évaluation pour les TPOs, d’une durée d’environ 6 heures;
- Présentation détaillée de la méthodologie d’évaluation formelle pour les TPOs, d’une durée d’environ 6 heures;
- Présentation détaillée de la conduite de projets pilotes dans les TPOs, d’une durée d’environ 3 heures.

En ce qui concerne la certification de professionnels en génie logiciel, il s’agira de développer le modèle de certification (*certification scheme*) en utilisant le SWEBOK. Il serait peut-être intéressant de mener ce projet en collaboration avec les acteurs qui sont préoccupés par la relève tels que TechnoCompétence et TechnoMontréal. Il faudra aussi se familiariser avec la norme ISO qui règlemente le processus de certification d’individus (ISO 2003).

Organisation

- Ressources humaines
    - Spécialistes de la norme ISO pour TPOs
    - Professeurs et formateurs
- Ressources matérielles
    - Trousses de déploiement
    - Guide de conduite de projets pilotes
    - Guide d’évaluation informelle
    - Gabarit de plan d’amélioration des pratiques
    - Guide d’évaluation formelle
    - Matériel de formation
    - Équipement informatique et équipement pour présentations
    - Salle multimédia pour vidéo-conférence

Implantation

À court terme, les trousses de déploiement seront traduites en français. Les 6 cours développés par le groupe 24 seront aussi traduits. À moyen terme, des professeurs des Cégeps seraient formés à la norme ISO à l’aide du matériel développé. La formation pourra se donner à distance grâce à l’équipement multimédia de l’ÉTS. Ceux-ci pourraient alors intégrer cette connaissance dans leurs cours respectifs. Une session spéciale de formation pourrait se donner, au printemps 2010, lors du colloque annuel avec les professeurs en informatique de Cégeps qui se tient à l’ÉTS. Des études de cas d’implantation (*case studies*) de la norme ISO pour les TPOs seront rédigées et intégrées au matériel de formation.

À moyen terme, le développement du modèle de certification sera développé en collaboration avec les acteurs locaux. Par la suite, il sera possible de débuter la certification d'individus.

Veille et éveil technologique auprès des TPOs

Cette catégorie de service vise à assurer une veille technologique, c.-à-d. un service de surveillance du développement technique et stratégique, et à contribuer à l'éveil technologique, c.-à-d. un service qui vise à accentuer la sensibilisation aux impacts des nouvelles technologies, des TPOs, des donneurs d'ordre, des organismes publics et des formateurs et professeurs afin de mieux orienter leurs stratégies et de les faire bénéficier de la technologie la plus récente.

Le centre jouera un rôle de catalyseur auprès des grandes industries et des organismes publics. Il s'agira de leurs faire connaître la norme ISO pour qu'ils l'appliquent soit à l'interne dans leurs nombreuses très petites organisations ou qu'ils l'imposent contractuellement à leurs fournisseurs.

Dans cette catégorie de services offerts, on y retrouvera les activités suivantes :

- L'organisation de conférences et d'ateliers;
- Des visites de centres de transfert du réseau;
- La démonstration d'applications pratiques de nouvelles technologies;
- La promotion du centre de l'ÉTS au Québec, au Canada et à l'étranger.

Méthode

Saisir et maintenir à jour l'information sur les technologies les plus récentes en génie logiciel pour les TPOs. Analyser cette information pour en dégager des tendances, des impacts et des occasions à saisir pour les diffuser.

Représenter le Canada aux groupes de travail ISO en rapport avec les TPOs, comme le groupe 24 et le groupe 25 traitant de la norme ISO/CEI 20000 – Services en TI. Participer au groupe de travail sur l'ingénierie des systèmes dans les TPOs commandité par l'INCOSE et l'Association Française en Ingénierie Système (AFIS).

Encadrer un groupe d'intérêt pour les TPOs québécoises. Ce groupe d'intérêt rassemblera des spécialistes, des gestionnaires et développeurs de TPOs, des donneurs d'ordre et des organismes publics en vue de faciliter les échanges d'information et le transfert technologique.

Maintenir le site web de l'ÉTS de support aux normes ISO pour les TPOs québécoises. Ce site permettra aux TPOs d'accéder à des documents (p.ex. trousses de déploiement, exemples d'applications, outils) et permettra aussi d'établir un réseau d'entraide au Québec entre les TPOs (p.ex. par l'animation de forum de discussion). Des sites semblables seront aussi développés par des membres du réseau international de support aux TPOs. Les TPOs québécoises pourront aussi accéder à l'information des sites des membres du réseau.

Organisation

- Ressources humaines
    - o Spécialistes de la norme ISO pour TPOs
    - o Professeurs et formateurs
- Ressources matérielles
    - o Salles de cours
    - o Matériel informatique
    - o Salle multimédia pour vidéo-conférence
    - o Documentation

Implantation

À court terme, des ateliers de sensibilisation à la nouvelle norme ISO seront offerts. Un réseau d'intérêt sera formé. Des exemples d'application de la norme ISO seront présentés aux participants du réseau. Les ateliers et présentations pourront se donner à distance à l'aide de l'équipement multimédia de l'ÉTS.

Stratégies du Centre

Dans cette section sont présentées succinctement les stratégies suivantes:

- Stratégie globale
- Stratégie de prix
- Stratégie de promotion
- Stratégie juridique
- Stratégie de financement

Stratégie globale

- La niche visée par le centre est l'ensemble des très petites organisations en TI du Québec;
- Le créneau des produits et services visé par le centre est centré sur les normes produites par l'ISO pour les très petites organisations;

Stratégie de prix

- **Volet normes, processus et outils pour les TPOs**

Produit et service	Stratégie de prix
Développement de trousses pour les TPOs	Gratuit
Développement de trousses pour les TPOs en démarrage	Gratuit
Intervention - Conduite de projets pilotes des trousses de déploiement en support à la norme ISO 29110	Programme CNRC
Intervention - Évaluation informelle des processus logiciels et amélioration des pratiques de génie logiciel	Programme CNRC
Intervention - Évaluation formelle de la conformité des processus à la norme ISO 29110	Programme CNRC

Tableau 8 Volet normes, processus et outils pour les TPOs

- **Volet projets de recherche**

Produit et service	Stratégie de prix
Projets de recherche sur les TPOs québécoises	Gratuit + Frais de déplacement
Encadrement d'étudiants de maîtrise et de doctorat	Gratuit
Participation à des projets internationaux sur les TPOs	Frais de déplacement
Mesure de l'adoption de la norme ISO 29110 et de son impact dans les TPOs Québécoises	Gratuit + Frais de déplacement
Développement de profils pour des domaines spécifiques (p.ex. Industrie aérospatiale, Industrie pharmaceutique, Industrie du jeu)	Gratuit + Frais de déplacement

Tableau 9 Volet projets de recherche

- **Volet formation**

  Afin de réduire les coûts de formation et de faciliter l'accès aux TPOs en région, il est proposé d'offrir des ateliers et conférences à l'aide de media électronique tel que la vidéo-conférence (p.ex. *Webinar* à l'aide de Synchro-média ou Skype).

Produit et service	Stratégie de prix
Développement de cours et d'ateliers	Gratuit

Développement de modules d'auto apprentissage (c.-à-d. *e-learning*)	Gratuit
Formation: • Cours 1 - Introduction aux normes ISO/CEI en génie logiciel; • Cours 2 - Introduction aux normes, rapports techniques ISO/CEI 29110 et aux trousses de déploiement pour les TPOs; • Auditoire visé: • Gestionnaires de très petites entreprises • Directeurs de très petits départements • Gestionnaires de très petits projets • Développeurs dans les TPOs • Donneurs d'ordre • Agences gouvernementales • Organismes sans but lucratif (OSBL) • Consultants	Gratuit
Formation: • Cours 3 - Développement d'un processus d'ingénierie de logiciels (*process engineering*) utilisant le Rapport Technique ISO/CEI 29110 Partie 5 – Guide d'Ingénierie et de Gestion • Cours 4 - Développement de logiciels utilisant le Rapport Technique ISO/CEI 29110 Partie 5 – Guide d'Ingénierie et de Gestion; • Cours 5 - Auto-évaluation de la conformité de processus de développement de logiciels à la norme ISO/CEI 29110; • Course 6 – Conduite de projet pilotes pour implanter la norme ISO/ CEI 29110 dans une TPO. • Auditoire visé: • Gestionnaires de très petites entreprises • Directeurs de très petits départements • Gestionnaires de très petits projets • Développeurs dans les TPOs • Agences gouvernementales • Organismes sans but lucratif (OSBL)	Gratuit
Formation: • Cours 3 - Développement d'un processus d'ingénierie de logiciels (*process engineering*) utilisant le Rapport Technique ISO/CEI 29110 Partie 5 – Guide d'Ingénierie et de Gestion • Cours 4 - Développement de logiciels utilisant le Rapport Technique ISO/CEI 29110 Partie 5 – Guide d'Ingénierie et de Gestion; • Cours 5 - Auto-évaluation de la conformité de processus de développement de logiciels à la norme ISO/CEI 29110; • Course 6 – Conduite de projet pilotes pour implanter la	

norme ISO/ CEI 29110 dans une TPO. - Auditoire visé: - Donneurs d'ordre - Consultants	Facturé
Formation des éducateurs (Université et Cégeps) québécois	Gratuit
Certification des professionnels en génie logiciel	Facturé

Tableau 10 Volet formation

- **Volet veille et éveil technologique auprès des TPOs**

Produit et service	Stratégie de prix
Représentation du Canada au groupe de travail ISO sur les TPOs	Frais de déplacement
Encadrement d'un groupe d'intérêt pour les TPOs québécoises à l'ÉTS	Gratuit
Développement et maintenance d'un site web, à l'ÉTS, de support à la documentation pour les TPOs	Gratuit

Tableau 11 Volet veille et éveil technologique auprès des TPOs

Stratégie de promotion

La promotion du centre sera effectuée principalement par le biais du site internet de l'ÉTS, des conférences et sessions de formation. Pour l'instant, le site du professeur Laporte est accessible à tous, c.-à-d. qu'il n'est pas à accès contrôlé. À l'avenir, du matériel pourrait être offert qu'à des TPOs dûment enregistrées. De même, le matériel développé par les éducateurs, pour les éducateurs, ne serait disponible qu'aux éducateurs dument enregistrés.

Stratégie de développement

Le groupe de travail de l'ISO complétera un des 4 profils (étapes), pour le développement de logiciels génériques, d'ici le début 2010. Le tableau suivant indique l'échéancier pour le développement des autres profils.

Profils ou Étapes	Échéancier
L'étape 1 s'adresse aux TPOs en démarrage (*Start-up*) et aux TPOs qui développent des projets de 6 personnes-mois ou moins;	Février 2011
L'étape 2 s'adresse aux TPOs qui exécutent un seul projet logiciel à la fois;	Février 2010
L'étape 3 s'adresse aux TPOs qui exécutent plusieurs projets logiciels à la fois;	Février 2011
L'étape 4 s'adresse aux TPOs qui veulent améliorer significativement la gestion de leurs affaires (p. ex. *Business Management, Portfolio Management*).	Février 2012

Tableau 12 Échéancier pour le développement des autres profils

Il sera possible de développer des profils pour d'autres domaines tels que le développement de logiciels critiques dès la publication par l'ISO du premier profil en 2010. Ce seront des étudiants de troisième cycle qui pourraient élaborer et valider, en industrie, les nouveaux profils. Une fois validés, ces profils seront soumis, à titre de contribution du Canada, au groupe de travail 24 de l'ISO.

Si l'industrie est intéressée, le centre pourrait démarrer, dès 2010, le développement de profils pour la norme ISO/CEI 20000 qui traite des services en TI.

Si l'industrie est intéressée, le centre pourrait démarrer, dès 2010, le développement de profils pour le domaine de l'ingénierie de produit. Ce groupe de travail, auquel participe le professeur Laporte, pourrait collaborer au développement et au pilotage des profils et trousses de déploiement et ainsi augmenter l'offre des services et produits du centre de l'ÉTS.

Stratégie juridique

Il est proposé que le centre de support en génie logiciel pour les TPOs soit une entité juridique de type à être défini et soit localisé au département de génie logiciel et des TI de l'ÉTS.

- Protection du nom du centre : enregistrement juridique
- Protocole d'entente du réseau des centres (voir annexe B)
- Droit de diffusion et d'utilisation des trousses par la clientèle visée
- Trousses et documentation : Propriété intellectuelle (p. ex. le copyright)

Stratégie de financement

Pour l'instant, ce projet est piloté par un professeur de l'ÉTS. Il n'y donc pas de dépense associée directement au projet jusqu'à présent. Les dépenses de voyage pour participer aux réunions des groupes de travail 20 et 24 de l'ISO sont défrayées, en partie par le Conseil canadien des normes (CCN). Le CCN défraie l'équivalent du coût du billet d'avion tandis que les autres dépenses de voyage sont défrayées par les fonds de recherche du professeur, soit environ 2,000$ par réunion, 2 fois par année.

Des professeurs du département de génie logiciel et des TI pourraient aussi, moyennant rétributions financières, apporter leurs expertises aux TPOs que ce soit par le biais d'intervention ou en donnant des ateliers.

Il serait possible d'utiliser des salles de cours et le réseau internet de l'ÉTS pour la formation à distance.

Le Programme d'aide à la recherche industrielle du CNRC couvrirait une partie du coût des interventions dans les TPOs. Son volet Stratégie emploi jeunesse pourrait peut-être défrayer certaines interventions.

La vente de formations permettra de défrayer quelque peu les dépenses du centre.

Par contre, si le centre veut produire un impact structurel sur les TPOs du Québec, il faudra que du personnel s'ajoute pour effectuer les nombreuses interventions et formations au Québec.

Conditions de démarrage

Pour assurer un démarrage du centre de support aux TPOs, les éléments suivants sont indispensables :

- Participation de l'ÉTS,
- Publication de la norme par l'ISO,
- Accessibilité aux fonds du Programme d'aide à la recherche industrielle du CNRC,
- Participation d'au moins un autre professeur du département de génie logiciel et des TI de l'ÉTS.

Il serait très souhaitable que TechnoMontréal et Technocompétence collaborent avec le centre.

Il serait très souhaitable que quelques professeurs de quelques Cégeps acceptent de participer aux travaux du centre dans leurs régions.

Il serait souhaitable que le gouvernement du Québec participe au financement du centre, à sa promotion.

Annexe A – Membres du réseau de support aux TPOs

On décrit brièvement ci-dessous les centres qui participant au réseau de support aux TPOs.

Centre d'Excellence en Technologies de l'Information et de la Communication

- La mission du Centre d'excellence en technologies de l'information et de la communication (CETIC) de la Belgique est définie en trois volets (CETIC 2007):
    - Aider les organisations à concevoir des produits et services de meilleure qualité, à en assurer la fiabilité, la sécurité, le respect des normes internationales, en leur apportant un soutien méthodologique. C'est l'objectif du département « Software & System Engineering ».
    - Aider les organisations à exploiter plus rapidement les nouvelles architectures informatiques réparties, dynamiques, orientées services, à accélérer le processus de transformation d'information en connaissance par les technologies sémantiques, à exploiter les réelles opportunités du logiciel libre, en mettant à leur disposition une expertise technologique de pointe. C'est l'objectif du département « Software & Services Technologies »
    - Aider les organisations à embarquer plus d'intelligence et plus de connectivité dans les systèmes qu'elles développent, mettre au point les démonstrateurs technologiques, les prototypes, en exploitant et intégrant les nouvelles technologies électroniques. C'est l'objectif du département « Embedded & Communication Systems »

Le Centre de recherche public Tudor du Luxembourg[9]

Le centre Henri Tudor contribue à l'amélioration et au renforcement de la capacité d'innovation des organisations et des organisations publiques pour lesquelles il propose une large palette de services et d'activités:

- recherche appliquée et expérimentale,
- recherche doctorale,
- développement d'outils, de méthodes, de labels, de certifications et de normes,
- assistance technologique, conseil et services de veille,
- transfert de savoir et pré-incubation d'entreprises.

La formation et la qualification de haut niveau viennent compléter cette gamme : près de 240 formations, disponibles via un catalogue (téléchargeable sur www.sitec.lu), sont dispensées chaque année aux professionnels.

Ses activités s'orientent principalement vers les domaines scientifiques et technologiques suivants:

- technologies de l'information et de la communication,
- technologies des matériaux,
- organisation et gestion des entreprises,
- technologies environnementales,
- technologies pour la santé.

Le Centre Henri Tudor cible les secteurs des services, de la production, de la santé et de la sécurité sociale, ainsi que le secteur public tout en portant une attention particulière aux PME.

[9] http://www.tudor.lu/

L'Irish Software Engineering Research Centre (LERO)[10]

The Irish Software Engineering Research Centre (Lero) was established in November 2005 with support from Science Foundation Ireland's CSET (Centre for Science, Engineering and Technology) program. Lero focuses on specific domains, especially those where reliability is crucial, including automotive, medical devices, telecommunications and financial services. We develop models, methods and tools that make it cheaper, faster or easier to produce this crucial software. The vision, mission and goals of LERO are listed in Table 13. Two researchers of Lero are members of WG24 and are actively participating to the conduct of the pilot projects described in a next chapter.

Lero's Vision is of Ireland as a world-leading location for the development of high-quality software intensive systems of high economic and social value.
Lero's mission is to deliver world-leading research in software engineering with a special emphasis on Evolving Critical Systems. Lero will enhance the quality and competitiveness of the Irish software industry through shared projects, knowledge transfer and education
Lero's Goals: **Establish a Sustainable, National Centre** • Establish the Lero brand in Ireland and abroad • Operate the centre efficiently, effectively and transparently • Develop and implement a long-term business plan for the centre • Build-up and maintain strong national university-industry research links • Impact on national software engineering education and training **World Class Research** • Produce internationally recognised research outputs • Establish close links with international research institutions **Focus on Strategic Industrial Domains** • Tackle research problems of industrial relevance • Validate and improve research results with industry • Help to make Ireland attractive for software engineering R&D

Tableau 13 Vision, mission and goals of LERO

The Association of Thai Software Industry

The Association of Thai Software Industry (ATSI) developed the Thai Quality Software (TQS) standard[i] to provide Thai Very Small Enterprises (VSEs) with a way to improve their process quality using a standard as a reference model. TQS is a staged implementation of ISO 12207. TQS was developed to respond to the following issues:

- Thai SMEs are not ready to implement the entire ISO 12207 standard.
- Not all ISO 12207 activities are suitable for SME operations.
- There is no assessment model for the ISO/IEC 12207 standard.
- Most software developers are not document-oriented.

To address those issues, ATSI, in collaboration with the Software Industry Promotion Association of Thailand (SIPA), financially supported a Special Working Group for the creation of a framework tailored to the needs of VSEs. The Special Working Group also developed the justifications that led to the establishment of an ISO/IEC JTC1 SC7 Working Group (WG24).

The ParqueSoft Organization of Columbia

The Software Technology Park Foundation (Fundación Parque Tecnológico del Software), ParqueSoft, is a not-for-profit organization established in 1999 for the purpose of creating and developing enterprises providing goods and services to the IT market. ParqueSoft is consolidating Southwestern Colombia's Science and Technology Corridor, integrating 14 Parks located in the

[10] Adapted from http://www.lero.ie/

following cities: Cali, Popayán, Pasto, Buga, Tuluá, Palmira, Buenaventura, Armenia, Manizales, Ibague, Villavicencio, Medellin, Sincelejo and Pereira.

To date, ParqueSoft and its network of Software Technology Parks house more than 250 VSEs where more than 1,000 software engineering professionals specializing in the industries' latest technologies, along with 200 other professionals, provide support in administrative and business development processes. These VSEs have, on average, 6 employees each.

ParqueSoft's goals for the year 2010 is to develop 400 competitive and productive IT enterprises which will export their software products and services to the international markets, and create 4,000 jobs in an innovative Science and Technology sector contributing to the regional economy by more than US $100 Million annually (Arenas 2007).

ParqueSoft has created an innovative support model encompassing five macro objectives supported by 16 synergistic strategies to promote enterprise development and research and development (R&D). The five macro objectives and their corresponding strategies are listed in Table 14.

1. To Provide an Infrastructure for Business Development and Support
To provide to all the companies a logistical support to facilitate their businesses development. Some of them are:

- Competitive physical and technological infrastructure
- Technological Support (Telco, Networking, Videoconference, Data Center)
- Effective Communications (Internet, Intranet, and Media,)
- Intellectual Property and Legal Support

2. To Develop the Best People in the Industry
So that the sector can be competitive in the world-wide markets requires human talent of world-wide class specialized and certified in specific areas of work, that is why ParqueSoft had developed a program of qualification of the human talent.

3. To Develop a Financial Strength
To provide support to facilitate to all the companies their Financial development. Some of them are: Entrepreneurship Promotion Funds, Risk Capital Funds and Savings Funds

4. To Support Enterprise Development
Provide support to facilitate to all the companies their development. Some of them are: Market Intelligence, Creative Marketing, Business Knowledge, Business Development, Business support and update.

5. To Become more Innovative, Reliable and Competitive Products
As a result of a Strategy Quality Program which concern in creating and improving the software process improvements model and quality assurance set of practices for small organizations in ParqueSoft, which must be simple, based as possible in common sense, fast and not too expensive to be implemented. A model to really fit VSE needs.

Tableau 14 Parquesoft's five macro objectives and their corresponding strategies (Arenas 2007)

ParqueSoft has completed the implementation of its Quality Management System based on ISO 9001. This certification turns ParqueSoft into the first enterprise incubator in Colombia to certify its quality processes. This is also being achieved by 17 of its VSEs. The next goal is to certify all ParqueSoft VSEs in the next 4 years. The delegate from Columbia on WG24 works at the ParqueSoft Foundation as Quality Assurance ParqueSoft Colombia Director.

CMM Fast-track Programme for Hong Kong SME Software Companies

This project is organized by the Hong Kong Computer Society and funded by the SME Development Fund of the Trade and Industry Department, HKSAR Government.

The main objectives of this project are to (Garcia et al. 2006):

- Provide an alternative means for Hong Kong Software SMEs to reach development capability assessed to Level 2 (Repeatable) or 3 (Defined) of CMMI® (Capability Maturity Model® Integration).
- Provide avenues for increased collaboration between Software SMEs in Hong Kong and partnering software companies located in the Pearl River Delta region.
- Foster relationships while encouraging cooperation, collaboration and communication amongst IT professionals, the software industry in Hong Kong, and international experts.

The project developed a toolkit to expedite understanding and adoption of CMMI by a model group of small and medium software organizations (Hareton et al. 2006). The government funded 15 companies (out of 600 who applied) and provided expertise and training to support them in pursuing model-based improvement. Five of these companies achieved CMMI maturity level 2 during the course of the project (May 2003-Dec 2004) (Garcia et al. 2006).

Annexe B – Articles de l'entente de collaboration

Article 1 : Objet

La présente convention a pour objet de promouvoir, de développer, de faciliter et d'intensifier la collaboration scientifique, technologique et académique entre les deux établissements dans le domaine du génie logiciel et des technologies de l'information.

L'organisation et l'ÉTS s'engagent chacun pour ce qui les concerne, à favoriser toutes les formes de rapprochement de coopération qu'ils jugeront opportunes.

Article 2 : Domaines de coopération

L'organisation et l'ÉTS, en vue d'atteindre l'objectif défini à l'Article 1, pourront mettre en place des coopérations, notamment dans les sujets suivants :

1. Établissement d'un réseau d'expertise international en génie logiciel afin d'aider les Très petites organisations (TPOs – organisations de moins de 25 employés) actives dans le développement logiciel
2. Établissement d'un catalogue intégré de leurs services respectifs à promouvoir conjointement au niveau des régions concernées.

Article 3 : Conventions spécifiques

Des conventions spécifiques sous formes d'Avenant(s) à la présente Convention précisant notamment l'objet de la coopération et partenariat, les modalités de financement, seront établies au cas par cas, soumises à l'approbation des autorités compétentes dans chacun des établissements contractants.

Article 4 : Consultations

Les deux partenaires procéderont à des consultations régulières afin d'évaluer le développement des activités communes, de dresser le bilan des résultats obtenus et des actions en cours de réalisation, et d'identifier de nouveaux axes de collaboration.

Article 5 : Responsables scientifiques

Les parties désignent chacune les responsables scientifiques suivants pour la mise en place et le suivi des initiatives visées par cette convention :

Pour l'organisation :
XXXX

Pour l'ÉTS :
M. Claude Y. LAPORTE, professeur
Département de génie logiciel et des technologies de l'information
Membre du GÉLOG (Laboratoire de recherche en génie logiciel)

Article 6 : Propriété intellectuelle

La propriété intellectuelle qui résulterait des travaux réalisés dans le cadre de cet accord, sera soumise aux dispositions légales applicables et aux procédures spécifiques souscrites par les parties à cet effet. Ceux qui seront impliqués dans la réalisation des travaux recevront la reconnaissance due, selon les pratiques institutionnelles et les politiques de propriété intellectuelle de chacune des parties.

Article 7 : Financement

Les parties s'engagent à mettre tout en œuvre pour obtenir le financement nécessaire à la mise en œuvre de la présente convention.

Article 8 : Durée

La présente convention est conclue pour une durée initiale de trois (3) ans, prenant effet à la date de signature. Elle pourra être renouvelée par l'accord des deux Parties.

Article 9 : Résiliation
La présente convention pourra être résiliée anticipativement par l'une ou l'autre partie moyennant préavis de six (6) mois, sans toutefois porter préjudice aux collaborations particulières en cours.

La résiliation de cette convention n'empêche pas la poursuite de collaborations particulières si les deux Parties en conviennent par écrit.

Annexe C - Projets d’étudiants de second cycle auprès des TPOs

Le tableau suivant énumère les projets d’étudiants en maîtrise réalisés dans le cadre des TPOs.

- Bégnoche, Luc, Développement de trousses de déploiement de processus pour les très petites entreprises du secteur logiciel, 2008.

- Brahimi Ahmed, Implantation de la Norme ISO/CEI 20000 dans la commission scolaire Marguerites Bourgeoys, 2008

- Bentoumi, Miloud, Étude sur les cas d'échec ou de réussite partielle en amélioration des processus logiciels dans des sociétés québécoises, 2008.

- Foisy, Marco, Rétro-ingénierie, développement et implantation de processus de tests logiciels et création d’une interface graphique dans un projet de recherche sur les éoliennes, 2007.

- Sédion, Yacouba, Développement et implantation d’un bureau de projets et de processus logiciels dans la société Vircom Inc, 2007.

- Bluteau, Karine, Étude et élaboration des besoins des très petites entreprises en développement de logiciels sur l'utilisation des normes en génie logiciel de l'Organisation internationale de normalisation (ISO), 2007.

- Desfossés, Yves, Méthode d’amélioration des services de TI basée sur ITIL dans les entreprises québécoises. École des technologies de l’information (ÉTI), Projet dirigé en codirection avec le professeur Alain April de l’ÉTS, 2006.

- Benchérif, Khaled, Développement et implantation d’un processus d’ingénierie de logiciels sécuritaires pour un aéronef autonome, 2006.

- Stambollian, Anabel, Amélioration de la performance des processus logiciels dans les petites et moyennes entreprises. Projet codirigé avec le professeur J-M Desharnais de l’ÉTS, 2006.

- Abou E Fattah, Mounir, Projet d’amélioration de la compétitivité des petites entreprises Québécoises de développement et de maintenance de logiciels, École des technologies de l’information (ÉTI), Projet dirigé en codirection avec le professeur J-M Desharnais de l’ÉTS, 2006.

- Rheault, Frédéric, Évaluation et amélioration des processus logiciels chez Loto Québec, École des technologies de l’information (ÉTI), Montréal. Codirection avec le professeur Michel Plaisant de l’UQAM, 2005.

- Deniger, Frédéric, Développement et implantation d’un cadre de gestion de projet pour la société Fog Studio, École des technologies de l’information (ÉTI), 2005.

- Piedboeuf, Dominique, Évaluation et amélioration des performances des processus de gestion des exigences et d’estimation des projets de développement et de maintenance de logiciels du Centre de génie logiciel de la défense nationale. Département de génie logiciel et des technologies de l’information, École de technologie supérieure, Montréal, 2005.

- Belkebir, Youssef, Élaboration d’un bilan des connaissances en génie logiciel chez

Bombardier Transport, Département de génie électrique, École de technologie supérieure, Montréal, 2003. Codirection avec le professeur Pierre Bourque de l'ÉTS.

- Mougja, Rachid, Technology Change Management project on Cyclomatic Complexity measurement tool introduction at Motorola Canada Software Center, Département de génie électrique, École de technologie supérieure, Montréal, 2003.

- Yagoub, Said, Adaptation du modèle d'évolution des capacités logiciel pour son utilisation dans un cadre d'enseignement du génie logiciel., Département de génie électrique, École de technologie supérieure, Montréal, 2003.

- Quenum, William, Mise en œuvre d'un programme de mesures (logiciel) chez Bombardier Transport, Département de génie électrique, École de technologie supérieure, Montréal, 2002.

- Bouët, Mickaël, Mise en place d'un groupe de support aux projets d'ingénierie (logiciel) chez Oerlikon Aérospatiale, Département de génie électrique, École de technologie supérieure, Montréal, 2002.

- Guessous, Youssef, Développement et déploiement des Processus Logiciels chez Astec APS, Département de génie électrique, École de technologie supérieure, Montréal, 2001.

- Amar, Fouzia, Ingénierie des exigences d'un jukebox numérique, Département de génie électrique, École de technologie supérieure, Montréal, 2001.

Tableau 15 Liste des projets d'étudiants réalisés dans les TPOs

Le tableau suivant énumère les projets d'étudiants en maîtrise en cours de réalisation dans le cadre des TPOs.

- Guillemot, Frédérick, Développement et implantation d'un processus de génie logiciel pour un groupe de recherche en neuroscience et d'une trousse de déploiement pour la conception logicielle inspirée de la future norme ISO pour les très petites structures.

- Kabli, Samia, Conception, réalisation et mise à l'essai de trousses de déploiement pour faciliter et accélérer l'implémentation de la norme ISO/CEI 20000 par les très petites structures.

- Goyette, Martin, Recherche, Développement et Implantation d'une gestion de projet améliorée chez AXON Intégration & Développement.

Tableau 16 Liste des projets d'étudiants en cours de réalisation dans les TPOs

Références

(AFNOR 2003) Ingénierie systèmes, Processus de vie des systèmes, Norme Française, AFNOR, NF Z 67-288, 2003.

(Anacleto et al. 2004) Anacleto, A., von Wangenheim, C.G., Salviano, C.F., Savi, R. “Experiences gained from applying ISO/IEC 15504 to small software companies in Brazil,” 4th International SPICE Conference on Process Assessment and Improvement, Lisbon, Portugal, pp. 33-37 (Avril 2004).

(Arenas 2007) Arenas, Liliana, “Small Entrepreneurships Software Enterprises Quality Process Assessment ParqueSoft Colombia, Real Case”, Director of Quality Assurance ParqueSoft, Colombie, 2007.

(Boehm 1987) Boehm, B., Improving Software Productivity, IEEE Computer, Septembre 1987, p 43.

(Bluteau 2007) Bluteau, C. Étude et élaboration des besoins de très petites entreprises en développement de logiciels sur l’utilisation des normes en génie logiciel de l’organisation internationale de normalisation (ISO), Masters Thesis, École de technologie supérieure, Montréal, Canada 2007.

(CETIC 2007) Rapport annuel 2007, Centre d’excellence en technologies de l’information et de la communication, www.cetic.be

(Charette 2005) Charette, R. N., Why Software Fails, IEEE Spectrum, septembre 2005, p 42-49

(CITA 1997) Cellule Interfacultaire de Technology Assessment, “Utilisation des Systèmes d'Information Inter-Organisationnels [SIO] par les PME Belges.” SIO *Research Report*, Cita-Computer Sciences Dept., University of Namur, Novembre 1997.

(Coleman 2008) Coleman, G.; O'Connor, R.; Investigating Software Process in Practice: A Grounded Theory Perspective, Journal of Systems and Software, Vol. 81, No. 5, pp 772-784, 2008.

(CRIM 1992) Applied Software Engineering Center, Proposal to Industry, Science and technology Canada and Ministère de l'Industrie, du Commerce et de la Technologie du Québec, Février 1992.

(Dion 1992) Dion, R.., Elements of a Process Improvement Program, Raytheon. IEEE Software, Vol. 9, no. 4, pp. 83-85. juillet 1992.

(Gauthier 2004) Gauthier, R., "Portrait et vision de l'industrie des technologies de l'information et des communications (TIC) au Québec," présenté à Conférence la Boule de Cristal CRIM, Montréal, 2004.

(Gibson et al. 2006) Gibson, D., Goldenson, D., Kost, K., Performance Results of CMMI®-Based Process Improvement, SEI, Technical Report CMU/SEI-2006-TR-004, August 2006.

(Haley 1996) Haley, T. J., Software process improvement at Raytheon. IEEE Software 13, no. 6, 33-41. Figure tirée de la revue IEEE Software, 1996.

(Hofer 2002) Hofer, C.; Software Development in Austria: Results of an Empirical Study among Small and Very Small Enterprises, Proceedings of the 28th Euromicro Conference, 2002: 361-366

(Humphrey 2002) Humphrey, W.S., Winning with Software - An Executive Strategy, Addison Wesley, 2002.

(ISO 2003) ISO/IEC 17024:2003, Conformity assessment — General requirements for bodies operating certification of persons, International Organization for Standardization/ International Electrotechnical Commission: Geneva, Switzerland.

(ISO 2005) ISO/IEC TR 19759:2005, Software Engineering Body of Knowledge, Technical Report, International Organization for Standardization/ International Electrotechnical Commission: Geneva, Switzerland.

(ISO 2006) ISO/IEC 15289 Systems and software engineering - Content of systems and software life cycle process information products (Documentation). International Organization for Standardization/ International Electrotechnical Commission: Geneva, Switzerland.

(ISO 2008a) ISO/IEC 12207:2008, Information technology – Software life cycle processes, International Organization for Standardization/ International Electrotechnical Commission: Geneva, Switzerland.

(ISO 2008b) ISO/IEC 24773: 2008, Software engineering — Certification of software engineering professionals — Comparison framework, International Organization for Standardization/ International Electrotechnical Commission: Geneva, Switzerland.

(ISO 2009) ISO/IEC PDTR 29110-5-1 Software Engineering - Lifecycle Profiles for Very Small Entities (VSEs) - Part 5-1: Management and Engineering Guide - Basic Profile, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue detail.htm?csnumber=51160, 15/01/2009.

(Krasner 1998) Krasner, H., Using the cost of quality approach for software, Crosstalk - The Journal of Defense Software Engineering, Volume 11, Numéro 11, novembre 1998.

(Laporte et al. 2005) Laporte, C.Y.; Renault, A.; Desharnais, J. M.; Habra, N.; Abou El Fattah, M.; Bamba, J. C.; Initiating Software Process Improvement in Small Enterprises: Experiment with Micro-Evaluation Framework, SWDC-REK, International Conference on Software Development, University of Iceland, Reykjavik, Iceland, Mai 27-Juin 1, 2005, pp. 153-163.

(Laporte et al. 2006) Laporte, C.Y.; April, A. and Renault, A.; Applying ISO/IEC Software Engineering Standards in Small Settings: Historical Perspectives and Initial Achievements, Proceedings of SPICE Conference, Luxembourg, 2006.

(Laporte et al. 2008a) Laporte, C.Y., Alexandre, S, Renault, A., *Developing International Standards for Very Small Enterprises*, IEEE Computer, Mars 2008, p 82-85.

(Laporte et al. 2008b) Laporte, C.Y., Alexandre, S., O'Connor, R., A Software Engineering Lifecycle Standard for Very Small Enterprises, in R.V. O'Connor et al. (Eds.): EuroSPI 2008, CCIS 16, pp. 129–141.

(Laporte 2009) The Development of International Standards for Very Small Entities: Historical Perspectives, Achievements and Way Forward, National Association of Software and Services Companies, NASSCOM Quality Summit, Inde, 23 mai 2009.

(McFall et al. 2003) McFall, D., Wilkie, F.G., McCaffery, F., Lester, N.G., Sterritt, R. Software processes and process improvement in Northern Ireland. 16th International Conference on Software & Systems Engineering and their Applications, Paris, France, Decembre 2003, 1–10, ISSN: 1637-5033.

(Mtigwe 2005) Mtigwe, B., The entrepreneurial firm internationalization process in the Southern African context: A comparative approach, International Journal of Entrepreneurial Behavior & Research, Vol. 11 No. 5, 2005, pp. 358-377

(OCDE 1990) Le génie logiciel: l'expérience des différents pays membres de l'OCDE et le défi posé à l'action internationale, Organisation de coopération et de développement économique, Direction de la science, de la technologie et de l'industrie, Paris, sept. 1990

(SCC 2008) Corporate Plan Summary 2008–2009 to 2012-2013, Standard Council of Canada, p 50

(Shintani 2006) Shintani, K, Empowered Engineers are Key Players in Process Improvement. Présentation à: the First International Research Workshop for Process Improvement in Small Settings, Software Engineering Institute, CMU/SEI-2006-SR-01, Pittsburgh, PA, 2006.

[i] http://www.atsi.or.th/atsi_th