



HAL
open science

Proposition d'une approche de transformation digitale de la composante gestion de projets de la gouvernance IT à base de modèles

Ibrahim Hamzane

► **To cite this version:**

Ibrahim Hamzane. Proposition d'une approche de transformation digitale de la composante gestion de projets de la gouvernance IT à base de modèles. Informatique [cs]. Université Hassan II Casablanca (Maroc), 2021. Français. NNT: . tel-03323596

HAL Id: tel-03323596

<https://theses.hal.science/tel-03323596>

Submitted on 22 Aug 2021

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.

Université Hassan II de Casablanca
Thèse de Doctorat

Présentée par :
Ibrahim HAMZANE

Spécialité :

Informatique

Sujet de la thèse :
Proposition d'une approche de transformation digitale de la composante gestion de projets de la gouvernance IT à base de modèles

Thèse présentée et soutenue à la Faculté des Sciences Ben M'Sik,

Le 27/03/2021

Devant le jury composé de :

El Houssine LABRIJI	PES	Faculté des Sciences Ben M'Sik - Casablanca	Président
Abderrahim MARZOUK	PES	Faculté des Sciences et Techniques - Settat	Rapporteur
Mohamed AZZOUAZI	PES	Faculté des Sciences Ben M'Sik - Casablanca	Rapporteur
Fadoua GHANIMI	PH	Ecole Nationale Supérieure de Chimie - Casablanca	Rapporteur
Abdellah MENOUI	PH	Académie Internationale Mohamed VI de l'Aviation Civile - Casablanca.	Examineur
Ahmed EDDAOUI	PH	Faculté des Sciences Juridiques, Economiques et Sociales d'Ain Sebâa - Casablanca	Examineur
Abdessamad BELANGOUR	PH	Faculté des Sciences Ben M'Sik Casablanca	Directeur de thèse

Etablissement : Faculté des Sciences Ben M'Sik

CEDoc : Sciences et Applications

Formation Doctorale : Mathématiques et Informatique et Traitement de l'Information (MITI)

Nom du laboratoire : Laboratoire De Technologie De L'information Et De Modelisation (LTIM)

Remerciements

Les travaux présentés dans cette thèse ont été effectués au sein du Laboratoire de Technologie de l'Information et Modélisation du Centre d'Etudes Doctorales Sciences et Applications Université Hassan II de Casablanca, Faculté des Sciences Ben M'Sik.

Je remercie mon directeur de thèse Monsieur Abdessamad BELANGOUR, je le prie d'accepter à cette occasion le témoignage de ma reconnaissance pour ses conseils prodigués tout au long de mes recherches, et sa participation au jury de cette thèse.

Mes sincères remerciements vont à Monsieur Mohamed AZOUAZI pour ses conseils et son suivi au début de mon projet doctoral.

Je remercie Messieurs les membres du jury pour m'avoir honorée en acceptant d'évaluer ce travail.

Un remerciement spécial à mes collègues dans l'équipe du Professeur Monsieur Abdessamad BELANGOUR pour leurs conseils et leur soutien tout au long de cette thèse. Plus spécialement Mouad BANANE, Alae ERRAISSI, Fatima Zahra KALNA. Aussi à mon cher ami Zouhair IBN BATOUTA.

Je ne peux clôturer mes remerciements sans me retourner vers les personnes qui me sont le plus cher et qui ont un rôle essentiel dans ma réussite :

Merci à mes chers parents pour leurs inspirations, leur confiance en moi, leur amour et leurs prières.

Merci à ma femme pour son soutien quotidien, ses encouragements aux moments les plus difficiles et sa compréhension.

Merci à toute ma famille.

Résumé

Les technologies de l'information (IT) représentent une grande partie des investissements des organisations, de nos jours l'IT est devenu le facteur clé de succès d'une entreprise avec l'émergence de la transformation digitale et l'apparition de plusieurs nouvelles technologies telles que : BIG DATA, Intelligence Artificielle, Machine Learning, etc. Des recherches antérieures ont identifié le lien entre investissement informatique et productivité, ainsi de nombreux facteurs affectent la valeur qu'une organisation peut tirer de son investissement informatique. Cependant, la diversité des référentiels utilisés dans la gouvernance IT rend la tâche plus compliquée aux décideurs vu l'émergence des nouvelles technologies qui proposent des solutions rapides et puissantes. Aussi, une deuxième problématique consiste à la transformation digitale du patrimoine existant vers les nouvelles méthodes et technologies devenue une nécessité.

Nous répondons à la complication du choix d'un référentiel de la gouvernance IT avec un outil qui aide les décideurs à choisir le référentiel en adéquation avec leurs attentes en utilisant une méthode d'analyse multicritère basée sur le modèle des sommes pondérées Weighted Sum Model (WSM).

Aussi, en se basant sur l'ingénierie dirigée par les modèles (IDM) nous avons proposé une approche pour générer un méta-modèle unique à partir de l'élaboration d'un cadre de référence en se basant sur les concepts de l'IDM ainsi que les techniques de matching. Hors, Vu la diversité des domaines inclus dans la gouvernance IT (gestion de projets, gestion des services informatiques, gestion de la qualité, gestion des ressources humaines et chaîne d'approvisionnement), nous avons appliqué notre approche sur le domaine de gestion de projets dans un premier temps et nous avons proposé un méta-modèle générique pour ce domaine, puis nous nous sommes basé sur deux méthodologies de gestion de projets les plus utilisées par les professionnels à savoir : les méthodes Prédictives et les méthodes agiles.

Nous avons validé notre approche en appliquant une transformation depuis les méthodes prédictives vers les méthodes agiles chose qui aidera les entreprises à effectuer une transformation digitale de leurs projets, nous avons appliqué la transformation sur un cas de projet réel d'entreprise par la suite.

Dans cette thèse, nous vous présentons les défis rencontrés, la démarche méthodologique suivie, ainsi que l'implémentation de notre approche, ses phases de génération.

Mots-clés: Gouvernance IT, COBIT, ITIL, ISO 38500, TOGAF, CMMI, IDM, Méta-modélisation, Prise de décision, Analyse multicritère, SCRUM, PRINCE 2.

Abstract

Information technology (IT) accounts for a large proportion of the investments made by organizations. Today, IT has become the key factor in a company's success with the emergence of digital transformation and several new technologies such as BIG DATA, Artificial Intelligence, Machine Learning, etc. Previous research has identified the link between IT investment and productivity, so many factors affect the value an organization can derive from its IT investment. However, the diversity of repositories used in IT governance makes the task more complicated for decision-makers given the emergence of new technologies that offer fast and powerful solutions. A second issue is the digital transformation of existing heritage into new methods and technologies that have become a necessity.

We respond to the complication of choosing an IT governance repository with a tool that helps decision-makers choose the repository in line with their expectations by using a multi-criteria analysis method based on the Weighted Sum Model (WSM).

Also, based on model-led engineering (MDI) we proposed an approach to generate a single meta-model from the development of a frame of reference based on MDI concepts as well as matching techniques. But, given the diversity of areas included in IT governance (project management, IT services management, quality management, human resources management, and supply chain), we first applied our approach to the project management area and proposed a generic meta-model for this area, and then we built on two project management methodologies that are most used by professionals: Predictive methods and agile methods.

We validated our approach by applying a transformation from predictive methods to agile methods, which will help companies to digitally transform their projects, we applied the transformation to a real business project case afterward.

In this thesis, we present the challenges encountered, the methodological approach followed, as well as the implementation of our approach, its generation phases, its prerequisites as well as the practical application of concrete cases of completed projects.

Key words: IT Governance, COBIT, ITIL, ISO 38500, TOGAF, CMMI, MDA, Metamodeling, Decision Making, Multicriteria Analysis, SCRUM, PRINCE 2.

Liste des Acronymes

Désignation	Abréviation
APF	Adaptive Project Framework
APM	Adaptive Project Management
ATL	ATLAS Transformation Language
CAB	Change Advisory Board
CCB	Change Control Board
CIM	Computational Independent Model
COBIT	Control Objectives for Information and related Technology
COS	Condition of satisfaction
DL	Deep Learning
DSDM	Dynamic System Development Methodology
DSDM	Dynamic Systems Development Method
FDD	Feature Driven Development
GSI	Gouvernance du système d'information
IA	Intelligence Artificiel
IDM	Ingénierie Dirigée par les Modèles
IT	Technologie de l'Information
ITIL	Information Technology Infrastructure Library
ITSM	Information Technology Service Management
MCDA	Multicriteria decision analysis
MDA	Model Driven Architecture

MDE	Model Driven Engineering
ML	Machine Learning
MSP	Managed Service Provider
PDM	Platform Description Model
PIM	Platform-Independent Models
PMBOK	Project Management Body of Knowledge
PMI	Project Management institute
PO	Product Owner
POS	Project Overview Statement
PRINCE 2	PRojects IN Controlled Environments
PSM	Model Specific Platform
QVT	Query View Transformation
RAD	Rapid Application Development
SI	Système d'information
SM	Scrum Master
SOA	Service Oriented Architecture
SWOT	Strength, Weaknesses, Opportunities and Threats
TDD	Test Driven Development
US	User Story
WSM	Weighted Sum Model
XMI	XML Metadata Interchange
XML	eXtensible Markup Language
XP	Extreme Programming

Table des figures

Figure 1 : Gouvernance d'entreprise et gouvernance IT	21
Figure 2 : Gouvernance IT par business unit	22
Figure 3 : Rôle global des méthodes de gestion de projets.....	30
Figure 4 : Méthodologie prédictive de développement de produits	32
Figure 5 : Types de la transformation digitale	40
Figure 6 : Modèle de l'approche IDM	47
Figure 7 : Architecture à 4 niveaux de l'approche IDM.....	48
Figure 8 : Vue d'ensemble sur la transformation ATL [90].....	51
Figure 9 : Vue d'ensemble sur la transformation ATL	52
Figure 10 : Radar Multicritère des différents référentiels	65
Figure 11 : Interface de l'application de choix d'un référentiel de la gouvernance IT.....	67
Figure 12 : Ecran de pondération remis à la société USA HOME.....	68
Figure 13 : Décision multicritère de l'entreprise USA HOME	69
Figure 14 : Approche de construction d'un méta-modèle	72
Figure 15 : Modélisation du cadre de référence	73
Figure 16 : Inclusion du cadre de référence dans le PIM	74
Figure 17 : Construction d'un méta-modèle	75
Figure 18 : Modélisation du cadre de référence	79
Figure 19 : Cadre de référence pour la gestion de projets.....	81
Figure 20 : Méta-modèle générique de Gestion de projets [88]	85
Figure 21 : Méta-modèle détaillé de Gestion de projets.....	85
Figure 22 : Aperçu d'un exemple de cadre de projet prédictif	89
Figure 23 : Cadre de référence des méthodes Prédictives	91
Figure 24 : Méta-modèle PRINCE 2 [25].....	94
Figure 25 : Processus de la méthode PRINCE 2.....	97
Figure 26 : Méta-modèle PRINCE 2.	98
Figure 27 : Méta-modèle des méthodes Prédictives.	100
Figure 28 : Cadre de référence des méthodes Agiles.....	108
Figure 29 : Aperçu du cadre de projet Agile	109
Figure 30 : Exemple d'un Sprint Backlog.....	118
Figure 31 : Méta-modèle SCRUM.	119
Figure 32 : Méta-modèle des méthodes Agiles.....	120
Figure 33 : Modèle de transformation de modèle dans l'ingénierie des modèles	124
Figure 34 : Graphe de transformation en utilisant trois types de projets.....	130
Figure 35 : Organisation projet de la Solution cible Tax Authority	133
Figure 36 : Planning du projet Solution cible Tax Authority	134
Figure 37 : Nouveau planning du projet Solution cible Tax Authority	135
Figure 38 : Backlog restitué du projet Tax Authority.....	136
Figure 39 : Nouvelle organisation projet de la Solution cible Tax Authority	136

Liste des tableaux

Tableau 1 : Comparaison de diverses méthodologies dans une perspective de gestion de projets [63].....	37
Tableau 2 : Comparaison de diverses méthodologies dans une perspective de gestion de projets (Charvat J. , 2003)	37
Tableau 3 : Différence fonctionnelles entre l'approche prédictive et agile	38
Tableau 4 : Matrice SWOT minimale du CMMI.....	57
Tableau 5 : Matrice SWOT minimale de COBIT 5	57
Tableau 6 : Matrice SWOT minimale du TOGAF.....	57
Tableau 7 : Matrice SWOT minimale de l'ISO 38500	58
Tableau 8 : Matrice SWOT minimale du référentiel ITIL.....	58
Tableau 9 : Matrice SWOT minimale de PMBOK	58
Tableau 10 : Matrice SWOT minimale de PRINCE 2	59
Tableau 11 : Matrice SWOT minimale de SCRUM.....	59
Tableau 12 : Familles d'analyses multicritère [17]	60
Tableau 13 : Matrice de choix multicritère.....	64
Tableau 14 : Matrice de choix multicritère relative à l'entreprise USA HOME.....	69
Tableau 15 : Liste des concepts remonté dans le cadre de référence.....	82
Tableau 16 : List des concepts remontés.	83
Tableau 17 : Résultat du matching interprétative des concepts.....	84
Tableau 18 : Structure de la méthode PRINCE 2.	99
Tableau 19 : Structure du méta-modèle des méthodes Prédictives.	101
Tableau 20 : Différences entre les méthodologies de gestion de projets Agile.	107
Tableau 21 : Exemple du backlog Produit	117
Tableau 22 : Structure de la méthode SCRUM.....	120
Tableau 23 : Structure du méta-modèle des méthodologies Agile.	122
Tableau 24 : Aperçu du code ATL de la transformation.....	129
Tableau 25 : Temps de transformation en utilisant trois types de projets	130
Tableau 26 : Comitologie du projet Solution cible Tax Authority	134

Table des matières

REMERCIEMENTS	3
RESUME	4
ABSTRACT	5
LISTE DES ACRONYMES	6
TABLE DES FIGURES	8
LISTE DES TABLEAUX	9
TABLE DES MATIERES	10
INTRODUCTION GENERALE	13
CONTEXTE ET MOTIVATION	13
OBJECTIFS ET CONTRIBUTIONS DE LA THESE	15
STRUCTURE DE LA THESE	16
CHAPITRE I : ETAT DE L'ART	17
I.1. INTRODUCTION	17
I.2. GOUVERNANCE IT	18
I.2.1. Définition de la gouvernance IT	18
I.2.2. Rôle de la gouvernance IT	18
I.2.3. Relation avec la gouvernance d'entreprises	20
I.2.4. Importance de la gouvernance IT	22
I.2.5. Référentiels Standards de la gouvernance IT	23
I.2.6. Bonnes pratiques de la gouvernance IT	24
I.3. GESTION DE PROJETS	26
I.3.1. Définition de la gestion de projets	26
I.3.2. Rôle de la gestion de projets	27
I.3.3. Approches de la gestion de projets	30
I.3.4. Comparaison entre méthodes prédictives et méthodes agiles	35
I.4. TRANSFORMATION DIGITALE	38
I.4.1. Définition de la transformation digitale	38
I.4.2. Relation avec la gestion de projets	40
I.5. INGENIERIE DES MODELES	43
I.5.1. Définition	43
I.5.2. Niveaux d'abstraction	44
I.5.3. Approche MDA	45
I.5.4. Approche IDM	45
I.5.1. Standards de l'OMG	47
I.5.2. Transformation de modèles	48
I.5.2.1.1. Approches de transformation de modèles	48
I.5.2.1.2. Langages de transformation de modèles	50
I.6. CONCLUSION	53
CHAPITRE II : PROPOSITION D'UNE METHODE POUR LE CHOIX D'UN REFERENTIEL DE LA GOUVERNANCE IT	55
II.1. INTRODUCTION	55
II.2. ANALYSE SWOT APPLIQUEE A LA GOUVERNANCE IT	55
II.2.1. Définition de la méthode SWOT	56
II.2.2. Application de l'analyse SWOT	56
II.2.3. Matrice SWOT minimale des référentiels de la gouvernance IT	56
II.3. ETUDE COMPARATIVE MULTICRITERE DES REFERENTIELS DE LA GOUVERNANCE IT	59
II.3.1. Méthodes d'analyse multicritère	59

II.3.1. Méthode Weighted Sum Method (WSM)	60
II.3.2. Critères de comparaison multicritères	60
II.3.3. Poids des critères	63
II.3.4. Matrice de choix multicritères	63
II.3.5. Discussion	65
II.4. DEVELOPPEMENT D'UN OUTIL DE CHOIX D'UN REFERENTIEL DE LA GOUVERNANCE IT	66
II.5. ETUDE DE CAS	67
II.5.1. Pondération des critères suivant l'étude de cas	67
II.5.2. Résultat obtenus	69
II.5.3. Justification des résultats	69
II.6. CONCLUSION	70
CHAPITRE III : PROPOSITION D'UNE APPROCHE GENERIQUE DE CONSTRUCTION DE META-MODELES & APPLICATION SUR LA GESTION DE PROJETS	71
III.1. INTRODUCTION	71
III.2. PROPOSITION D'UNE APPROCHE GENERIQUE DE CONSTRUCTION DE META-MODELES.....	72
III.2.1. Définition du cadre de référence	72
III.2.2. Constitution des concepts	73
III.2.3. Analyse des concepts remontés	74
III.2.4. Construction du méta-modèle.....	75
III.3. PROPOSITION D'UN META-MODELE GLOBAL POUR LA GESTION DE PROJETS.....	75
III.3.1. Gestion de projets.....	75
III.3.2. Application de l'approche proposé sur la gestion de projet.....	79
III.3.2.1. Définition du cadre de référence.....	79
III.3.2.2. Constitution des concepts	81
III.3.2.3. Analyse des concepts remontés.....	84
III.3.2.4. Construction du méta-modèle	84
III.4. CONCLUSION	85
CHAPITRE IV : PROPOSITION D'UN META-MODELE POUR LES METHODES PREDICTIVES DE LA GESTION DE PROJETS	88
IV.1. INTRODUCTION	88
IV.2. PRINCIPES DES METHODES PREDICTIVES.....	89
IV.2.1. Collecte et documentation des exigences	89
IV.2.2. Implémentation du projet	90
IV.2.3. Test du projet.....	90
IV.2.4. Livraison et déploiement du projet.....	90
IV.2.5. Maintenance	90
IV.3. PROPOSITION D'UN CADRE DE REFERENCE.....	90
IV.4. CONSTITUTION DES CONCEPTS.....	92
IV.5. METHODE PREDICTIVE PRINCE 2	93
IV.5.1. Rôles.....	94
IV.5.2. Principes	95
IV.5.3. Thèmes	96
IV.5.4. Processus	96
IV.5.5. Livrables.....	97
IV.5.6. Proposition d'un méta-modèle pour PRINCE 2	98
IV.6. PROPOSITION D'UN META-MODELE GENERIQUE POUR LES METHODES PREDICTIVES.....	99
IV.7. CONCLUSION	101
CHAPITRE V : PROPOSITION D'UN META-MODELE POUR LES METHODES AGILES DE LA GESTION DE PROJETS	103
V.1. INTRODUCTION	103
V.2. PRINCIPES DES METHODES AGILES.....	104
V.3. PROPOSITION D'UN CADRE DE REFERENCE.....	107
V.4. CONSTITUTION DES CONCEPTS.....	108
V.4.1. Cadrage du projet	109
V.4.2. Planification du cycle	110
V.4.3. Cycle de développement du projet	111
V.4.4. Vérification Client.....	111

V.4.5. Revue finale.....	112
V.5. METHODE AGILE SCRUM.....	112
V.5.1. Rôles.....	113
V.5.2. Evénements	115
V.5.3. Artefacts	116
V.5.4. Proposition d'un méta-modèle pour SCRUM	119
V.6. PROPOSITION D'UN META-MODELE GENERIQUE POUR LES METHODES AGILES.....	120
V.7. CONCLUSION	122
CHAPITRE VI : VALIDATION DE L'APPROCHE PROPOSEE.....	123
VI.1. INTRODUCTION	123
VI.1. TRANSFORMATION DES MODELES PREDICTIVES EN MODELES AGILES	124
VI.2. EXECUTION DE LA TRANSFORMATION.....	126
VI.3. TEST ET VALIDATION DE LA TRANSFORMATION	129
VI.4. DISCUSSION DES RESULTATS.....	130
VI.5. ETUDE DE CAS	131
VI.5.1. Projet « Solution cible Tax Authority ».....	131
VI.5.2. Transformation de la gestion du projet Tax Authority en agile.....	134
VI.6. CONCLUSION	136
CONCLUSION GENERALE	138
REFERENCES.....	140
ANNEXE A.....	147
A.1 - JOURNAUX INTERNATIONAUX (SCOPUS).....	148
A.2 - CONFERENCES INTERNATIONALES (SPRINGER & SCOPUS)	149
A.3 - CONFERENCES NATIONALES	149
ANNEXE B.....	151
B.1 - CERTIFICATS DE MANAGEMENT.....	152
B.2 - CERTIFICATS DE TRANSFORMATION DIGITALE	152
B.3 – AUTRES CERTIFICATS.....	153

Introduction générale

Contexte et Motivation

Cette thèse porte sur le domaine de la gouvernance des systèmes d'information avec un cas d'étude sur la gestion de projets.

La gouvernance d'entreprise est un mécanisme de régulation permettant de s'assurer que la stratégie de l'organisation est effectivement appliquée dans l'entreprise. C'est un mécanisme permettant de mettre en place une série de processus susceptibles de mettre en œuvre les objectifs stratégiques, de responsabiliser l'ensemble des acteurs et de faire en sorte que tous les acteurs s'approprient les processus, en totale transparence, avec une politique de communication bien identifiée, et des rôles clairement définis.

La gouvernance des systèmes d'information (GSI) fait partie intégrante de la gouvernance d'entreprise. Elle correspond à la mise en place des moyens par lesquels les parties prenantes peuvent s'assurer de la prise en compte de leurs préoccupations dans le fonctionnement du système d'information (SI). La mise en œuvre de la gouvernance informatique est importante pour piloter et faire évoluer le système d'information en accord avec les parties prenantes. Cette exigence est sérieusement amplifiée à l'heure du numérique face à toutes les nouvelles technologies qui ont été lancées récemment (Big DATA, Intelligence Artificielle IA, Machine Learning ML, Deep Learning DL ...).

Ainsi, sans un bon gouvernail, chaque entreprise risque de se perdre dans une mer aux objectifs sans fin et inaccessibles. Selon les statistiques fournies par le Club des grandes entreprises informatiques françaises (CIGREF) en 2017 [1], 51% des entreprises s'appuient sur COBIT pour la gouvernance de leurs systèmes d'information (IT), tandis que 16% utilisent des processus internes et 32% ne s'inspirent pas de COBIT dans le monde [1]. Par conséquent, il existe plusieurs référentiels de la gouvernance IT que les professionnels utilisent séparément. Existe-t-il une méthode pour aider les professionnels à choisir une référence selon leurs besoins ?

Aussi, les référentiels standards ont tendance à être formellement technique, et contiennent énormément d'information qu'il faut analyser et structurer par la suite pour

pouvoir les exploiter. Existe-t-il une manière unique de voir l'ensemble des référentiels ? Existe-t-il une agrégation de connaissance en matière de la gouvernance IT ? et plus spécialement en matière de gestion de projets ?

Dans l'ère de la transformation digitale, existe-t-il une méthode de transformation des projets géré avec les anciennes méthodes vers les nouvelles méthodes de gestion de projet dites agile.

Cette thèse propose de résoudre les problèmes liés au choix d'un référentiel de gouvernance selon le besoin des organisations, aussi elle propose une approche qui nous permettra de générer un méta-modèle à partir de plusieurs modèles, nous avons étudié comme cas d'exemple la gestion de projets en appliquant la transformation IDM des méthodologies de gestion de projets prédictives aux méthodologies de gestion de projets agiles. Ci-dessous plus de détail sur les résultats de notre thèse.

Objectifs et contributions de la thèse

La formulation de notre approche est passée par plusieurs étapes :

- **Un état de l'art** se rapportant à la gouvernance IT, la gestion de projet, la transformation digitale, l'ingénierie de modèle et les méthodes de transformation de méta-modèle.
- **Une étude multicritère sur les différents référentiels existants** en se basant sur l'analyse **SWOT** (*Strenght, Weaknesses, Opportunities and Threats*), sur la méthode **WSM** (*Weight Sum Method*), ainsi que l'aide à la prise de décision multicritère (**MCDA**).
- Nous avons produit une **application aux industriels** qui permettra aux décideurs de choisir un référentiel de gouvernance adapté selon le besoin des entreprises.
- Proposition d'une approche de construction d'un méta-modèle fiable en utilisant la théorie des ensembles.
- Application de l'approche de création de méta-modèle en prenant comme **étude de cas la gestion de projets** et Génération d'un **méta-modèle générique** de gestion de projets.
- Nous avons ensuite construit un **méta-modèle** des méthodes Agiles.
- Nous avons ensuite construit un **méta-modèle** des méthodes Prédictives.
- Nous avons ensuite effectué une **transformation du méta-modèle** prédictive vers le méta-modèle agile.
- Effectuer une transformation digitale d'un projet réel d'une entreprise du prédictive vers l'agile.

Structure de la thèse

Ce document, qui constitue le mémoire de la thèse, compte six chapitres organisés comme suit :

- Le premier chapitre du rapport dresse l'état de l'art de la gouvernance IT et sa relation avec la gouvernance d'entreprise. A ce stade, nous avons mis le point sur les principaux référentiels de gouvernance utilisés dans le monde de l'informatique, aussi nous allons passer en revue les méthodes de gestion de projet, la transformation digitale, l'ingénierie dirigée par les modèles et les méthodes de transformation des méta-modèles.
- Le deuxième chapitre est consacré à l'étude comparative multicritère, entre les différents référentiels de la gouvernance IT ; Le chapitre commence par une étude analytique visant à identifier les avantages et les inconvénients de chaque référentiel en se basant sur l'analyse SWOT. Une étude multicritère, fondée sur un ensemble de critères clés, est réalisée par la suite. Puis une application a été réalisée sur une entreprise marocaine.
- Le troisième chapitre présente notre approche de génération d'un méta-modèle selon un processus constitué de plusieurs étapes, ce processus nous permettra de définir un méta-modèle basique que nous allons enrichir avec plusieurs concepts qui seront matchés pour construire un méta-modèle optimisé, nous avons par la suite appliqué cette approche sur la gestion de projet.
- Le quatrième et cinquième chapitre présentent la création de méta-modèle de deux méthodologies de gestion de projets prédictive et Agile.
- Dans le sixième chapitre nous validons notre modèle en faisons une transformation du méta-modèle prédictive vers le méta-modèle agile en passant par un cas d'étude de projet réel.
- En guise de conclusion, nous offrons une synthèse des points élaborés dans notre recherche, en jetant la lumière sur les perspectives de recherches à venir.

.

Chapitre I : Etat de l'art

I.1. Introduction

Dans ce chapitre, nous allons distinguer la gouvernance d'entreprise et la gouvernance IT dans le but de présenter une vue globale sur les pratiques dans le monde de l'entreprise, nous allons passer en revue les différents référentiels et bonnes pratiques utilisés pour piloter les différentes activités IT (gestion de projets, gestion des services informatique, gestion de la qualité, gestion des ressources humaines et la chaîne d'approvisionnement). Puis nous allons nous focaliser sur la composante gestion de projet et la transformation digitale. Par la suite nous allons nous orienter vers l'ingénierie des modèles et sa relation avec la gestion de projets entant qu'axe principale de notre recherche.

I.2. Gouvernance IT

La gouvernance informatique est un concept que de nombreuses entreprises utilisent pour aider à gérer les risques, contrôler les coûts informatiques et assurer leurs fonctions informatiques à leur plein potentiel. En règle générale, les clients qui travaillent avec des fournisseurs de services veulent avoir l'assurance que la prestation sera en mesure de leur offrir les mêmes avantages que les grandes entreprises bénéficient d'une gouvernance informatique interne. En effet, il est possible de réaliser d'importantes économies en mettant en œuvre un cadre de gouvernance informatique. Proctor & Gamble, après avoir mis en place un cadre de gouvernance informatique, a économisé 500 millions de dollars sur quatre ans. La gouvernance informatique a aidé la marque à réduire ses coûts d'exploitation et à rationaliser ses équipes technologiques pour réaliser des économies significatives.

I.2.1. Définition de la gouvernance IT

La gouvernance est l'ensemble des processus qui tendent à harmoniser le monde des affaires, des projets et de l'expérience. Selon le CIGREF, la gouvernance est étroitement liée à la notion de gestion d'entreprise. Gouvernance informatique [2]: "Capacité organisationnelle exercée par le comité de direction, la direction générale et le SI gestionnaires pour superviser la formulation et la mise en œuvre du Stratégie SI".

L'ITGI fournit aux dirigeants d'entreprise et aux conseils réalisateurs avec recherche originale, ressources en ligne et cas des études pour les aider à assumer leurs responsabilités dans le domaine des TI la gouvernance. Il a conçu et rédigé COBIT V4.1, essentiellement ressource pédagogique pour les gestionnaires de l'information, général gestion, gestion et contrôle des systèmes d'information professionnels [3].

À l'extérieur, lorsque nous nous concentrons sur les pratiques de gouvernance spécifiquement dans la gouvernance informatique, nous rencontrons plusieurs sources (ITIL, CNIL, SOX, CMMI, ISO 9001, ISO 9002, ISO 14000, ISO 17799, ISO 27001, 27002, 27003, 27005 ...), chaque référence offre une vue partielle de la gouvernance informatique. Ainsi, pour une meilleure gouvernance de cet article, nous ferons d'abord une distinction de vocabulaire entre ces références puis nous définirons la gouvernance informatique.

I.2.2. Rôle de la gouvernance IT

L'équipe de gouvernance informatique fait partie de l'équipe de direction globale, car l'objectif de la gouvernance informatique est d'aligner la fonction informatique d'une organisation sur les objectifs commerciaux de l'entreprise. Dans ce contexte, les MSP sont souvent invités à aligner leurs services informatiques sur les pratiques commerciales internes existantes. Lorsque la gouvernance informatique fonctionne bien, l'entreprise et le MSP travaillent ensemble pour créer des structures qui aident une organisation à rester sur la bonne voie pour atteindre ses KPI et atteindre les livrables attendus par les parties prenantes.

L'un des principaux rôles de la gouvernance informatique est d'aider les organisations à se conformer aux réglementations régissant la protection des données, la responsabilité financière et la reprise après sinistre. En plus des réglementations fédérales et internationales comme le RGPD, une entreprise peut également avoir des exigences internes pour le traitement des informations confidentielles et la conservation des données. Un programme formel de gouvernance informatique peut établir un système de meilleures pratiques et de freins et contrepoids pour s'assurer que ces exigences sont respectées.

Dans la pratique, les rapports sur la gouvernance informatique sont directement transmis à la direction de l'organisation. Le rôle d'une initiative de gouvernance informatique sera de veiller à ce que les décisions informatiques favorisent les progrès de l'entreprise vers ses objectifs plus larges. Le MSP sera chargé d'assurer :

- L'informatique de l'organisation offre des fonctionnalités et des services qui aident à réaliser les principales priorités de l'entreprise
- Tout nouvel investissement dans les technologies informatiques permet à l'entreprise d'obtenir de nouveaux résultats tangibles qui ne seraient pas possibles sans cet investissement
- L'informatique conduit à une meilleure satisfaction des parties prenantes - clients, partenaires et investisseurs
- Les ressources sont utilisées à leur plein potentiel tandis que le retour sur investissement des équipements et services informatiques est optimisé
- Les risques et responsabilités liés aux TI sont gérés de manière proactive

L'un des avantages de considérer la gouvernance informatique dans le cadre de vos offres MSP est que la valeur ajoutée pourrait dépasser de loin l'effort. Vos clients peuvent être

assurés que vous serez chargé de prendre une vue à 360 degrés de haut niveau de la stratégie informatique.

I.2.3. Relation avec la gouvernance d'entreprises

Dans les années 1800, l'énergie de vapeur a révolutionné le monde avec la mécanisation, par la suite dans les années 1900 le monde a subi l'électrification avec les lignes d'assemblage et la production en masse de Toyota, puis dans les années 1970 l'industrielle subit une autre transformation avec la venue de l'automatisation, l'installation des robots et des automates programmables a révolutionné le monde, en même temps les premiers briques de l'ère informatique est déjà en place avec l'essor de la technologie / le hardware et séparation des processus métiers et la naissance des softwares [4].

Maintenant nous sommes au début d'une nouvelle ère industrielle avec la digitalisation (Industrie 4.0) avec l'arrivée de l'internet des objets et le big data. En même temps en informatique nous assistons à une 3eme ère informatique avec le cloud, la transformation des business model et l'essor des plateformes.

Ainsi, la DSI en tant qu'organisation dans l'entreprise (Direction des Systèmes d'Information) est au cœur de la transformation digitale. Elle doit désormais adapter continuellement le système d'information au gré des changements d'organisation et de développement de nouvelles activités.

La digitalisation bouleverse la stratégie de l'IT ainsi que la gouvernance de l'IT au sein des entreprises [5]. Différents modèles de gouvernance d'entreprise sont de plus en plus examinés et analysés à mesure que la mondialisation s'installe sur les marchés mondiaux. Il est également devenu de plus en plus clair que les environnements et les structures d'entreprise peuvent varier de manière substantielle, même lorsque les objectifs commerciaux sont généralement universels.

La gouvernance au niveau des entreprises est née de la séparation entre les détenteurs de titres de propriété (les actionnaires) et les officiers en charge de l'administration (la direction) [6]. Ce principe de séparation est l'essence de la gouvernance. En effet, les propriétaires historiques des entreprises furent très vite remplacés par des fonds d'investissements. L'équilibre fragile qui existait entre les différentes parties prenantes

(propriétaires historiques, actionnaires, direction, employés, etc.) fut alors brisé par le monde de la finance.

Par conséquent, les entreprises furent obligées d'acheter, re-acheter, fusionner, restructurer et accumuler de grandes dettes. Ce phénomène est appelé le principe de concentration [7]. Des lors, les analystes américains commencèrent à se préoccuper des orientations prises par les grands groupes ainsi que du manque et de l'insuffisance des réglementations. Au début des années 2000, plusieurs groupes de travail pilotes par le gouvernement US ont été chargés de redéfinir les structures d'autocontrôle au niveau managérial. Une série de lois et de réglementations ont été alors proposées, et c'est à partir de là que la gouvernance des entreprises est née.

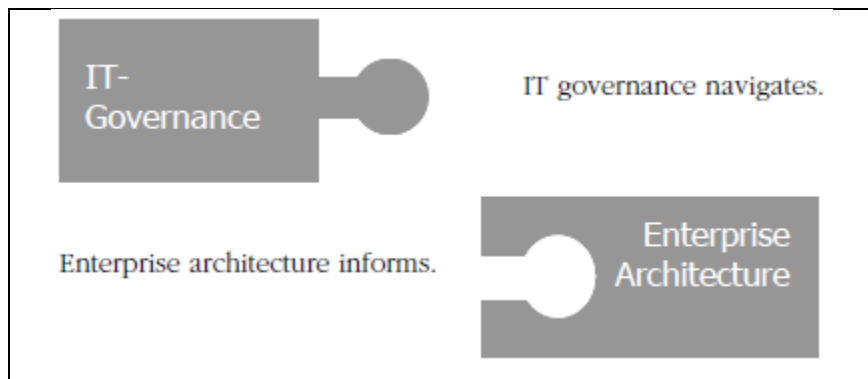


Figure 1 : Gouvernance d'entreprise et gouvernance IT

La gouvernance IT (au niveau des systèmes d'information) est fortement liée à la gouvernance corpo (la gouvernance au niveau des entreprises). En effet, les systèmes d'information des entreprises ne sont pas généralement développés uniquement dans une perspective de gouvernance des entreprises [8]. Ces systèmes sont plutôt développés pour soutenir les processus opérationnels, le business, et en même temps fournir les données essentielles qui seront utilisées pour des considérations de gouvernance d'entreprise [9]. Pour une grande partie des entreprises nord-américaines et européennes, la gouvernance IT est une conséquence directe des lois SOX "Sarbanes-Oxley" de régulation financière qui impose un niveau de contrôle de l'information au sein de l'organisation ainsi que le suivi et l'intégration de l'IT en charge de la partie financière de l'organisation. Ces lois définissent plusieurs opérations et règles de contrôle des compagnies cotées en bourse. Le résultat de ces lois est que les organisations devront adapter leurs infrastructures IT aux réglementations.

Ainsi, il faut distinguer plusieurs composantes de la gouvernance IT au sein de l'entreprise, voir la figure ci-dessous :

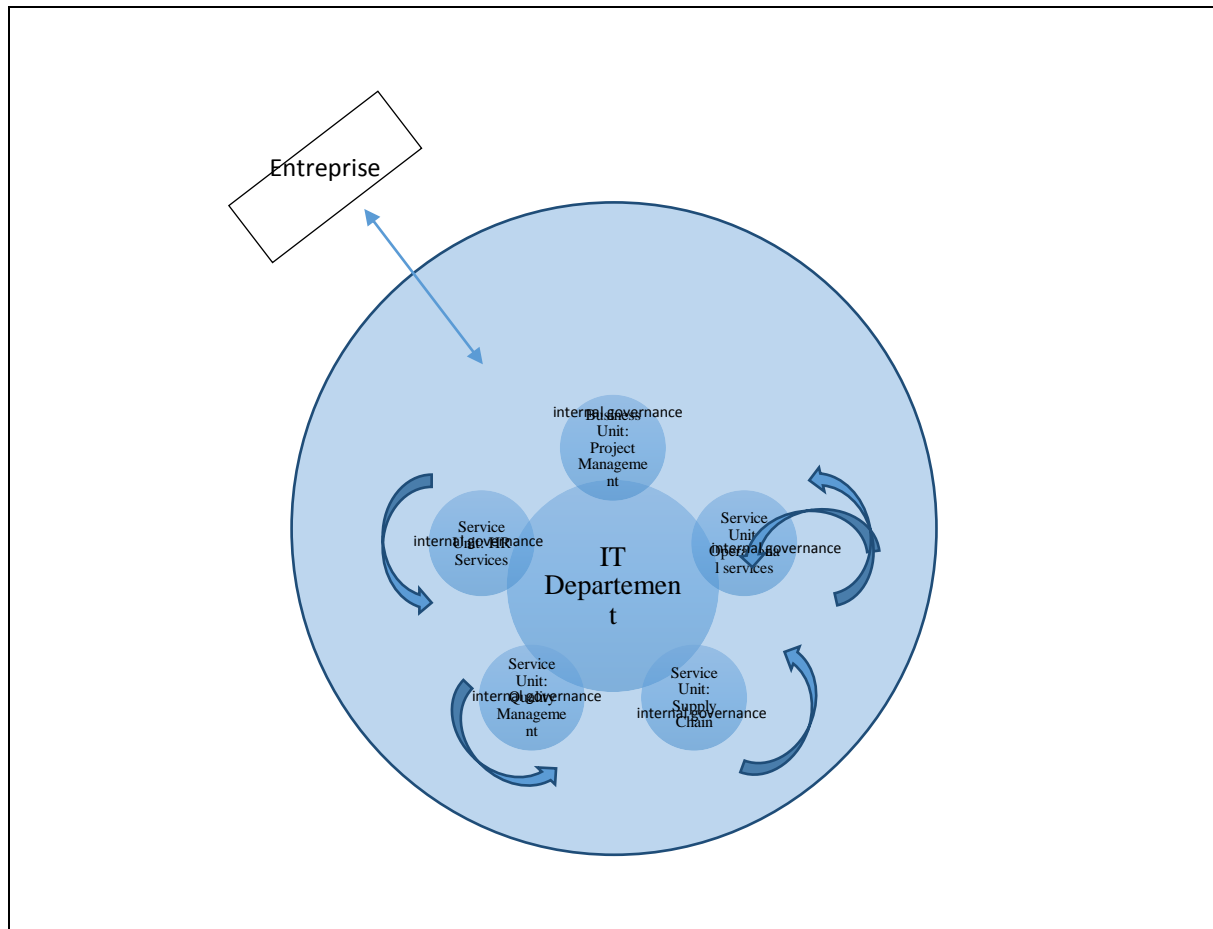


Figure 2 : Gouvernance IT par business unit

En résumé, la gouvernance IT est, d'une part, une conséquence directe des contraintes législatives qui imposent un certain nombre de mécanismes de contrôle et de gestion dans le système d'information, et d'autre part, la gouvernance IT est liée aux besoins de plus en plus croissants de compétitivité et de business des organisations.

I.2.4. Importance de la gouvernance IT

Tout d'abord, la gouvernance informatique signifie que vous pouvez aider vos clients à atteindre les résultats souhaités. Les investissements informatiques sont soigneusement pris en compte et chaque achat est associé à un retour attendu sur la satisfaction client, la productivité ou la gestion des ressources. Encourager une organisation à aligner ses dépenses informatiques sur les priorités de l'entreprise peut améliorer les économies de coûts, réduire les redondances et tirer le meilleur parti de nouveaux produits ou services.

La planification de la gouvernance informatique conduit également à une meilleure harmonisation et réactivité par rapport aux objectifs de l'organisation. L'articulation des priorités d'une entreprise signifie que le MSP et l'organisation seront transparents quant à leurs besoins et attentes informatiques et établiront des normes de réactivité à tout goulot d'étranglement de flux de travail ou à tout défi de mise en œuvre [10]. La standardisation des processus et des plates-formes peut rationaliser les opérations informatiques et supprimer les formalités administratives.

De plus, la gouvernance permet une prise de décision objective pour les MSP investissant leurs ressources dans diverses solutions informatiques. Un modèle de gouvernance informatique peut aider un MSP à prendre des décisions concernant la gestion et le contrôle des activités informatiques chez ses clients. Grâce aux directives, les MSP peuvent gérer correctement leurs propres ressources critiques et transmettre ces avantages aux clients. Par exemple, la gouvernance informatique peut dicter comment et quand un équipement informatique peut être remplacé ou un abonnement renouvelé.

I.2.5. Référentiels Standards de la gouvernance IT

Selon l'Organisation internationale de normalisation (ISO) et la Commission électrotechnique internationale (CEI) : la norme est "un document établi par consensus et approuvé par un organisme reconnu, qui prévoit, pour une utilisation courante et répétée, des règles, des directives ou fonctionnalités, pour des activités ou des résultats qui garantissent un niveau d'ordre optimal dans un contexte donné." [11].

En conséquence, les normes reposent sur un modèle hiérarchisé pouvant être reconnu (certifié ou évalué) par leur propre organisation, c'est une liste ou un ensemble d'exigences à remplir pour revendiquer la certification des systèmes de gestion d'une organisation, dans un domaine de gestion donné (services, sécurité de l'information, gouvernance, etc.).

Retrouvez ci-dessous quelques exemples de normes.

- **ISO 38500** : cette norme permet d'aider les chefs d'entreprise à s'assurer que l'utilisation des actifs informatiques est positive et contribue à la performance de l'organisation en répondant aux exigences de la norme ISO / CEI 38500, les organisations sont en mesure de surveiller l'utilisation des technologies de l'information, d'assurer la continuité et la durabilité de l'entreprise, d'aligner les

actifs informatiques sur les besoins de l'entreprise et assurer une mise en œuvre et un fonctionnement appropriés des actifs informatiques [12].

- **ISO 9001** : Norme qui établit les exigences d'un système de gestion de la qualité. Il aide les entreprises et les organisations à devenir plus efficaces et à accroître la satisfaction des clients [13] [14].

I.2.6. **Bonnes pratiques de la gouvernance IT**

Alors que les meilleures pratiques sont un ensemble de comportements à adopter et à adapter pour gérer efficacement un aspect donné de l'informatique (gouvernance, valeur et qualité des services, projets, processus, développement d'applications, architecture de développement, etc.) qui sont considérés comme essentiels par la plupart des professionnels de Dans le domaine, il peut être trouvé sous forme de guides de bonnes pratiques tels que SCRUM ou adopté par des organisations telles que PRINCE 2 (organisation), ITIL (organisation) ou COBIT (ISACA).

Les meilleures pratiques sont généralement basées sur 2 types de modèles :

- **Modèles étagés** ; structurée par niveaux de maturité ; Par exemple CMMI, eSCM-CL ...
- **Modèles "continus" ou "cycliques"** ; structuré par domaine de compétence ou de phase ; Par exemple ITIL 2011, TOGAF 9.2, COBIT 5 ...

Il est à noter que CMMI propose les deux approches.

Dans ce qui suit quelques définitions des meilleures pratiques utilisées dans les services informatiques :

- **CMMI** : Le modèle de maturité des capacités intégré est un modèle pour évaluer le niveau de maturité d'une entreprise en termes des développements informatiques [15]. Développé en 1987 par le Génie logiciel de l'Université Carnegie Mellon CMMI est un modèle de développement et de maintenance des systèmes et applications informatiques conçu pour saisir, évaluer et améliorer la opérations d'entreprises d'ingénierie [16].

- **COBIT** : Un cadre pour le développement, mises-en œuvre, supervision, amélioration des pratiques de gouvernance et d'administration de l'information systèmes. Le cadre COBIT est publié par l'Institut de gouvernance et ISACA. Son but est de fournir un langage commun aux chefs d'entreprise discuter ensemble de leurs buts, objectifs et résultats [12].
- **TOGAF** : L'Open Group Architecture Framework est un ensemble de concepts et une norme de l'industrie couvrant domaine des architectures informatiques d'entreprise [17].
- **ITIL** : Bibliothèque de l'infrastructure des technologies de l'information cadre permet aux technologies de l'information (TI) de jouer un rôle de prestation de services plutôt que simplement spécialisé prise en charge [18], [19], [20], [21], [22]. Directives ITIL et les meilleures pratiques visent à adapter les actions et le budget besoins de l'entreprise et les modifier à mesure que l'entreprise se développe ou change de direction. L'objectif est d'améliorer l'efficacité et atteindre des niveaux de service prévisibles [23].
- **PMBOK** : Le corpus de connaissances en gestion de projets est un guide de gestion de projets conçu et produit par le PMI Project Management Institute. C'est assez un guide complet vise à stabiliser et à structurer la connaissance actuelle nécessaires pour mener à bien un projet conditions [24]. C'est une base de connaissances sur le projet la gestion. Il définit les domaines de connaissances liés dans ce domaine et décrit les bonnes pratiques mettre en place.
- **PRINCE 2** : Projets en environnement contrôlé est une méthode développée par le gouvernement britannique et accrédité par Axelos, basé sur les meilleures pratiques du projet la gestion. Cette méthodologie flexible s'applique à tous entreprises et uniquement pour un projet à portée définie. "Construit sur sept principes, thèmes et processus, PRINCE2 peuvent être adaptés pour répondre à des exigences spécifiques » [25].
- **SCRUM** : Méthodologie de gestion de projets agile, elle est utilisée notamment dans le développement de logiciels. C'est un cadre dans lequel les gens peuvent aborder des problèmes complexes problèmes d'adaptation, tout en étant productif et créatif fournir des produits de la valeur la plus élevée possible. » [26].

I.3. Gestion de projets

I.3.1. Définition de la gestion de projets

Audet avait défini la gestion de projets en 1986 comme « Un champ de connaissances est l'espace occupé par l'ensemble des personnes qui prétendent produire des connaissances dans ce domaine et cet espace est en même temps un système de relations entre ces personnes. Ces personnes sont des concurrents pour prendre le contrôle de la définition des conditions et des règles de production des connaissances » vis-à-vis du comportement des ordres professionnels, des auteurs, des universitaires. Les relations entre les organismes professionnels établis (Project Management Institute [PMI®], International Project Management Association [IPMA], etc.) et leurs modes de développement par l'individualisme et la collaboration (PMI à travers A Guide to the Project Management Body of Knowledge [PMBOK® Guide] ; IPMA à travers une base de compétences partagées, IPMA Competence Baseline [ICB], est contextualisée en fonction des besoins nationaux des associations nationales. Par exemple, le siège de PMI s'inspire du Global Project Management Forum (une sorte d'organe super institutionnel essayant de promouvoir une base commune en termes de connaissances) le souhait de créer des normes mondiales, et en outre PMI est très actif dans le soutien de la recherche dans des domaines tels que; l'établissement d'une théorie de la gestion de projet, la démonstration de la valeur de la gestion de projet pour les cadres et la réalisation de la stratégie d'entreprise à travers projets réussis. L'évolution des corpus de connaissances (Guide PMBOK®, APM BOK, etc.) est attestée par thèmes dans des articles et des livres, citant des techniques à la psycho-sociologie des groupes temporaires jusqu'à la création de connaissances et l'apprentissage organisationnel. De plus, le domaine, actuellement caractérisé par cette abondance d'initiatives, l'élaboration de normes, l'utilisation croissante des méthodes et techniques de gestion de projet, est dans une phase paradigmatique au sens de Kuhn en 1983. Cette phase est le lieu de la révolution, inaugurée par une subdivision croissante mais toujours étroite au sein de la communauté de gestion de projet que le paradigme positiviste existant a cessé de fonctionner adéquatement dans l'exploration de la nature. Un deuxième aspect plus profond sur lequel la signification de la Tout d'abord, le succès de la révolution nécessite l'abandon total ou partiel d'un ensemble d'institutions au profit d'un autre.

I.3.2. Rôle de la gestion de projets

Les organisations obtiennent des avantages commerciaux, produisent et améliorent des produits, conçoivent et développent des systèmes et des services, et investissent dans l'infrastructure de l'entreprise principalement par le biais d'activités de projet [27] [28]. Rien qu'au Royaume-Uni, environ 21% de la valeur ajoutée brute dans la fabrication et la construction proviennent de projets complexes de développement de produits et de systèmes [27]. En conséquence, il y a une tendance à normaliser les activités du projet au moyen de méthodes projets formalisés et génériques [29] - par exemple, la volonté du gouvernement et du secteur public de promouvoir et d'utiliser les méthodes projets [25] au cours des dernières années pour le développement et la gestion de grands projets informatiques / SI complexes. Avec la pression pour une livraison réussie des projets dans un environnement commercial en constante évolution, de nouvelles méthodes de gestion de projets sont de plus en plus développées, mis en œuvre et adaptés pour guider et soutenir la conception de projets, développement et livraison ([30], [27], [31], [32], [33]). Malgré la popularité croissante de certaines méthodologies (par exemple, PRINCE2) dans différentes organisations, y compris les départements du secteur public, des limitations et des faiblesses associées aux méthodes de gestion de projets structurés ont été signalées [34]. Certaines des difficultés auxquelles sont confrontés les organisations et les départements du secteur public sont l'indifférence des méthodologies à leurs intérêts commerciaux organisationnels et leurs avantages au-delà de ceux d'un seul projet; complexité de l'adaptation et de la modification; leadership et stratégie; et leur dépendance à la documentation et leur rigidité à faire face au changement [35] [36].

Le secteur privé représente environ 45% du marché des projets IT / IS. Ces organisations n'annoncent pas nécessairement leurs plans TI / SI à l'avance, et ne publient pas non plus les échecs, même si, de manière anecdotique, on pense qu'il y a des dépenses importantes mal placées et des projets qui ont échoué. Le secteur public représente 55% de l'ensemble du marché britannique des technologies de l'information [37]. Dans le secteur public, les projets de réforme des services publics dépendent fortement de nouveaux systèmes informatiques à grande échelle. Le gouvernement britannique dépense 14 milliards de livres par an en systèmes et services informatiques - le chiffre le plus élevé d'Europe - et, ces dernières années, le double de la somme dépensée en 1999. Malgré des efforts substantiels

investis depuis les années 1960 pour mieux comprendre l'échec des projets améliorations de l'informatique / SI, les taux de réussite des projets ont mis du temps à se matérialiser et des exemples d'échecs catastrophiques peuvent encore être trouvés: «Le documentaire de Channel 4 Dispatches,« How They Squander Our Billions »[09.03.09] a enquêté sur une variété de publics controversés projets, qui ont eu des millions, voire des milliards, dépensés pour eux, y compris le Programme national pour l'informatique du NHS [NPfIT], qui a été initialement annoncé comme coûtant 2,3 milliards de livres. Ce montant est devenu par la suite 6,2 milliards de livres sterling, et 12,7 milliards de livres sterling est la dernière estimation » [38].

Un rapport britannique récent sur le projet C-NOMIS (2009) illustre encore le problème: « Le Service pénitentiaire de Sa Majesté et le Service national de probation ont créé le Système national d'information sur la gestion des délinquants (NOMIS) en 2004. En 2007, le NOMIS avait dépensé 155 millions de livres. et le projet avait deux ans de retard et, en 2008, la décision a été prise de suspendre le projet. Le projet a été réduit de portée, pour un coût de 41 millions de livres sterling et a été redémarré en janvier 2008, la date d'échéance initiale. On s'attend maintenant à ce que les coûts à vie s'élèvent à 513 millions de livres, bien que cela ne soutienne plus la gestion de bout en bout des délinquants comme prévu initialement » [39]; il s'agit d'un exemple classique d'un projet livré en retard, sous-performant par rapport à sa portée spécifiée et plusieurs fois au-dessus du budget. Pendant des périodes telles que le ralentissement économique actuel, des pertes financières importantes infligent ainsi de lourdes conséquences commerciales aux maîtres d'ouvrage. Cela peut entraîner l'annulation d'un projet avec à la fois la perte pure et simple des ressources investies à ce jour et la perte des avantages escomptés.

Depuis 1999, les méthodes de gestion de projets sont fermement placées comme l'un des dix principaux facteurs contribuant à l'échec d'un projet, selon le Standish Group (2010). Dans la version 2008 du rapport CHAOS, il était indiqué que, bien que des améliorations du taux de réussite des projets (35%) soient obtenues, le taux des échecs (19%) et les performances des projets contestés (46%) restent à des niveaux qui méritent davantage d'attention.

Le lien entre les méthodes de gestion de projets utilisés pour gérer les projets IT / SI et les mauvaises performances n'est pas entièrement mis au jour, et il existe peu de recherches sur le rôle et la contribution des méthodes de gestion de projets à la performance globale du projet. À un niveau général, on peut dire qu'une mise en œuvre efficace des méthodes de

gestion de projets ne signifie pas automatiquement un résultat positif ou amélioré du projet; De plus, les performances médiocres ne résultent pas nécessairement d'une faible mise en œuvre des méthodes de gestion de projets. Pourtant, les méthodes de gestion de projets sont conçues et mis en œuvre pour réguler les processus de gestion de projets afin de rester concentré sur d'autres questions de performance critiques. Par conséquent, les méthodes de gestion de projets sont importantes. La revue de la littérature a révélé moins de cinq études abordant et examinant les méthodes de gestion de projets, très peu se concentrant sur la discipline IT / IS et encore moins utilisant des preuves empiriques dans leurs arguments. White et Fortune [34], dans leur étude quantitative, ont examiné une combinaison de méthodes, d'outils, de techniques et de méthodologies, concluant par déduction qu'un certain nombre de problèmes rencontrés par les organisations sont associés aux méthodes de gestion de projets, tels que PRINCE et PRINCE2. L'étude est approfondie et complète, bien que son approche ait limité les résultats à des hypothèses prédéterminées de succès. Le contexte dans lequel les méthodes de gestion de projets ont été examinés n'a pas été pris en compte.

Une étude de Davies et Hobday [27] a examiné l'effet de l'utilisation des méthodes de gestion de projets en cascade pour gérer un projet informatique à partir d'un projet logiciel complexe au sein du ministère de la Défense (MoD). Ils ont révélé une divergence entre les méthodes de gestion de projets (processus rationnel) et les processus souples, soulignant l'échec des méthodes de gestion de projets, comme la cascade, à refléter la réalité de la pratique. Ces domaines de déviation sont importants mais laissent des questions sans réponse, telles que la justification du choix d'une approche en cascade pour un projet logiciel complexe et quels sont les facteurs qui influencent cette décision. L'une des raisons associées à la complexité de la gestion de projets TI / SI est que les projets sont souvent mal gérés parce que les entreprises utilisent les mauvais processus pour les gérer ou appliquent des critères financiers inappropriés pour la sélection des projets [40].

À ce jour, aucun ensemble unique des méthodes de gestion de projets n'a été identifié comme garantissant la bonne exécution des projets IT / SI. Il a été estimé qu'il existe 1 000 méthodologies «de marque» utilisées par les organisations dans le monde [41]. Chaque méthodologie prétend être la réponse aux problèmes rencontrés par les projets et leurs organisations ou une réponse à des problèmes courants dans des contextes particuliers [42]. Cependant, bon nombre de ces nouvelles méthodologies ont été accusées d'être simplement un reconditionnement de concepts qui ont émergé dans les méthodes de gestion de projets

antérieurs trouvés dans le génie logiciel et la fourniture de TI / SI [43]. Les méthodes de gestion de projets sont considérés «comme un fétiche utilisé avec une rigidité pathologique pour lui-même; cela démontre le recours important des organisations à l'application de méthodologies comme une routine où le raisonnement derrière l'adoption et l'application de méthodologies est normalement sans réponse et laissé flou » [43].

La figure plus bas montre un résumé de la recherche globale entreprise pour examiner le rôle global des méthodes de gestion de projets, mettant en évidence la sous-section sélectionnée pour discussion dans cet article sur l'examen des avantages des méthodes de gestion de projets.

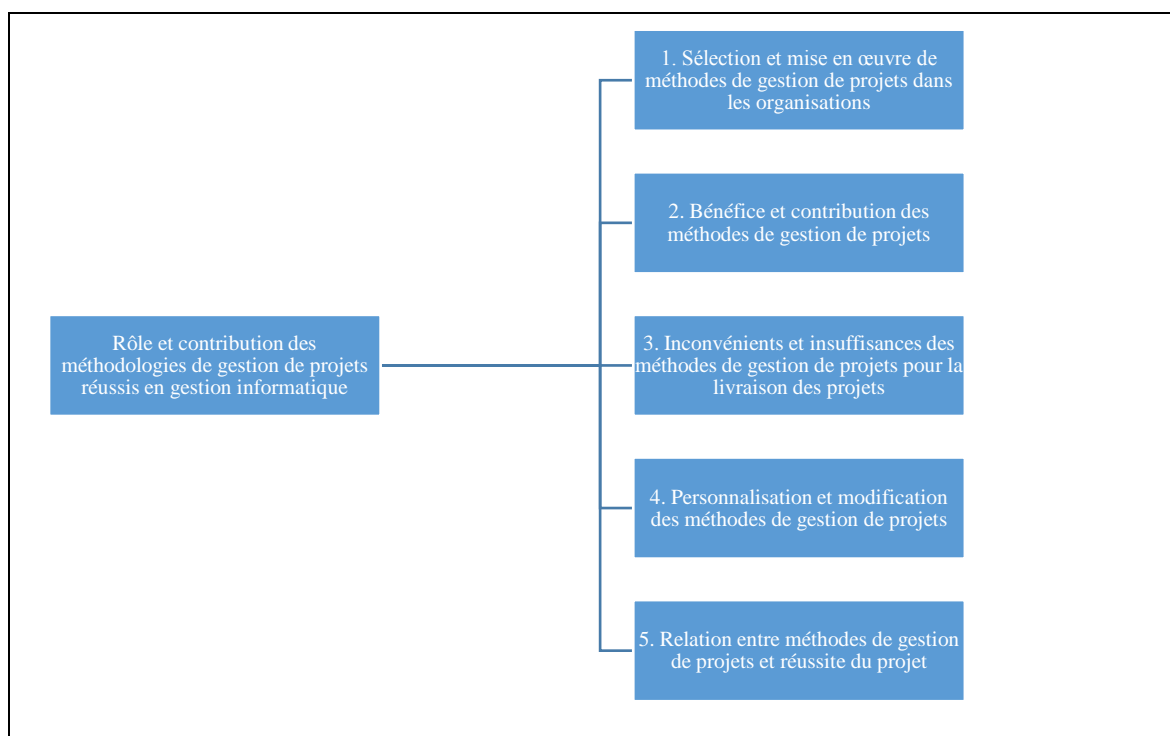


Figure 3 : Rôle global des méthodes de gestion de projets

Ainsi, la littérature présentée précédemment éclaire la justification de la nécessité de mieux comprendre comment les méthodes de gestion de projets soutiennent la performance, c'est-à-dire la réalisation réussie de projets TI / SI.

I.3.3. Approches de la gestion de projets

Le terme approche de gestion de projets est le plus souvent utilisé comme un ensemble de principes et de lignes directrices qui définissent la manière dont un projet spécifique est géré [44] [45]. La signification presque similaire a un terme de cadre de gestion de projets,

qui représente un ensemble opérationnel de règles, processus, méthodes et modèles à utiliser pendant le cycle de vie du projet.

I.3.3.1.1. **Gestion de projets prédictive**

La gestion de projets est appliquée dans le monde des affaires d'aujourd'hui à une variété de projets différents. Les principes établis dans les années 50 ont prescrit que les méthodes et procédures devraient être appliquées à chaque projet de manière uniforme. Cette mise en œuvre uniforme devrait garantir la robustesse et l'applicabilité à un large éventail de projets (projets simples et petits aux plus complexes et grands). L'idée de base derrière cette approche prédictive, rationnelle et normative est que les projets sont relativement simples, prévisibles et linéaires avec des limites clairement définies, ce qui facilite la planification en détail et le suivi de ce plan sans trop de changements [46] [47] [48]. Le but ultime de l'approche prédictive de gestion de projets est l'optimisation et l'efficacité dans le suivi du plan de projet détaillé initial, ou, après avoir dit de manière habituelle, pour finaliser le projet dans les délais, le budget et la portée prévus [49] [36].

En outre, cette approche prévaut dans la plupart des corpus de connaissances disponibles produits par les organisations de gestion de projets. La raison de cette domination pourrait être trouvée dans le fait que les premières versions des corpus de connaissances ont été développées dans les années 1980, lorsque l'approche prédictive était en fait la seule pratique de gestion de projets. Au cours des éditions suivantes des corpus de connaissances, les mises à jour des pratiques réelles ont été mises en œuvre, mais le rythme du changement n'a pas toujours été conforme aux attentes des praticiens.

Même si l'approche prédictive de la gestion de projets met l'accent sur la robustesse comme l'un de ses avantages (voir la figure plus bas), prescrivant que les mêmes méthodes et techniques puissent être appliquées uniformément à tous les projets, elle est de plus en plus mentionnée comme l'un des inconvénients cruciaux d'une telle approche. Aujourd'hui, un nombre croissant d'auteurs insiste sur le fait que «la taille unique ne convient pas à tous» [50] [36]. Les projets, comme les environnements commerciaux en général, deviennent progressivement complexes, avec un nombre plus élevé de tâches et des interrelations complexes, tandis que l'approche de gestion de projets prédictive est basée sur des relations de tâches principalement hiérarchiques et linéaires et ne peut pas refléter correctement toute la complexité et la dynamique des projets d'aujourd'hui. [51] [52].

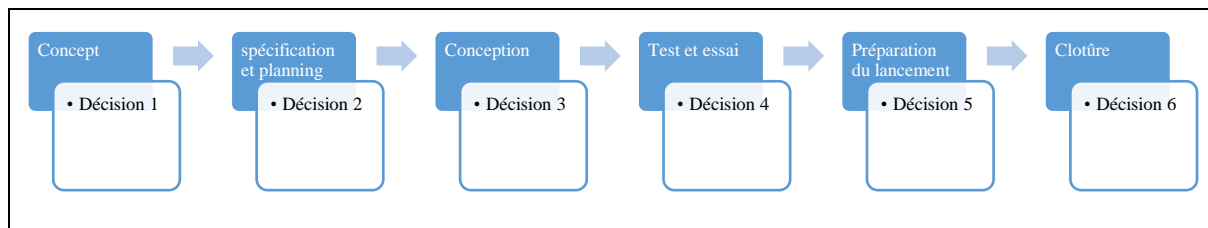


Figure 4 : Méthodologie prédictive de développement de produits

De plus, l'hypothèse que le projet est isolé de son environnement provoque le deuxième inconvénient majeur de l'approche prédictive [53] [28] [54]. Le changement, quelle qu'en soit la forme, est la réalité des environnements commerciaux actuels et des projets dans ces environnements sont inévitables en raison d'ajustements à des changements imprévisibles et dynamiques dans l'environnement du projet ou au sein du projet lui-même [55]. De plus, il est parfois très difficile de créer un projet complet planifier au début du projet en raison de l'incapacité de définir clairement les objectifs du projet.

Nous pouvons résumer que les principales raisons de l'inadéquation de l'approche prédictive de la majorité des projets d'aujourd'hui sont la complexité structurelle, l'incertitude dans la définition des objectifs et les contraintes de temps pour les projets.

I.3.3.1. Approche de gestion de projets agile

Toutes les objections à l'approche prédictive de gestion de projets, ainsi que les demandes croissantes d'innovations continues qui ont eu un impact sur toutes les industries et avec les tendances de réduction des coûts, ont abouti à l'avènement de nouvelles approches de gestion de projets [54]. Mais l'avènement de nouvelles approches est étroitement lié au domaine de l'ingénierie logicielle et du développement logiciel, et de nouvelles approches de gestion de projets apparaissent ainsi que de nouvelles approches du développement logiciel.

Ces nouvelles approches sont apparues sous plusieurs noms différents, mettant tous l'accent sur la différence avec l'approche prédictive même avec le nom. Le nom le plus souvent utilisé est l'approche agile, alors que presque la même idée et la même approche se retrouvent sous les noms d'approche lean, approche extrême et approche adaptative [56].

De plus, les nouvelles approches ne concernent pas seulement le suivi pur des processus, mais davantage la communication et la collaboration entre les membres de l'équipe de projet [54] [52]. Les membres de l'équipe sont beaucoup plus impliqués dans la prise de décision, et la communication est à la fois formelle et informelle [57].

Tout ce qui précède nécessite un changement de manière de penser et par conséquent des changements au sein de l'organisation spécifique qui essaie d'adopter l'une des nouvelles approches.

Certaines idées qui caractérisent les nouvelles approches de gestion de projets, comme l'approche itérative, ont émergé et ont été utilisées avant même, mais ce n'est qu'en 2001 avec le Manifeste Agile que ces idées ont gagné visibilité plus significative. Manifeste for Agile Software Développement, rédigé par le groupe d'auteurs, a mis en place quatre valeurs fondamentales de l'approche agile: «les individus et les interactions sur les processus et les outils, le travail des logiciels sur la documentation complète, la collaboration client sur la négociation des contrats, la réponse au changement plan » [27]. En mettant l'accent sur les éléments de gauche, cela ne signifie pas que les éléments de droite ne sont pas importants, mais seulement moins importants que les éléments de gauche. Même si Manifesto a été écrit pour le développement de logiciels agiles, toutes les valeurs fondamentales peuvent également être appliquées presque directement à la gestion de projets agile [54].

Le mot qui a été choisi pour différencier la nouvelle approche de l'existante était l'agilité. L'agilité peut être considérée comme la capacité de créer et de répondre au changement afin de créer de la valeur dans un environnement commercial turbulent. L'agilité, comme presque tous les efforts de recherche sont basée sur plusieurs principes commerciaux tels que l'innovation continue, l'adaptation des produits, le raccourcissement des délais de livraison, l'ajustement des personnes et des processus et des résultats fiables. L'agilité est aussi la capacité d'équilibrer la flexibilité et la stabilité. De plus, l'environnement agile est défini par Chin [50] comme celui qui contient avant tout une certaine incertitude et nécessite des connaissances spécifiques et souligne la nécessité de livrer le projet le plus rapidement possible. Par conséquent, Wysocki [49] déclare qu'un projet agile typique serait celui qui sera caractérisé avec une grande quantité d'incertitude et sera forcé de livrer très rapidement, avec des changements majeurs pendant l'exécution du projet.

Pour être plus comparable à l'approche prédictive, les auteurs mettent généralement en place une approche agile en plusieurs phases, similaires aux phases prédictives du cycle de vie des projets. [58] a proposé cinq phases de l'approche de gestion de projets agile: Envision (définir la vision, la portée du projet et l'organisation du projet), Spéculer (développer un modèle défini par les caractéristiques du produit et les contraintes de temps, et un plan d'itération pour la mise en œuvre de la vision), Explorer (livrer des pièces testées en peu de

temps et chercher continuellement un moyen de réduire les risques et l'incertitude du projet), Adapter (vérifier les livrables, la situation actuelle et le comportement de l'équipe pour s'adapter si nécessaire) et Fermer (fermer le projet, créer les leçons apprises et célébrer) . De la même manière, Chin [50] met en place son modèle de projet flexible avec quatre phases itératives: Visionate, Speculate, Innovate et Reevalute, et avec la phase de clôture Disséminer.

Comme déjà indiqué, l'idée de l'approche agile est d'embrasser les changements au cours du projet, donc l'approche agile est dans sa base d'approche itérative. Chaque itération, de préférence courte, comprend toutes les phases et la portée finale du projet est construite dynamiquement à chaque itération, et selon la portée du projet pouvait être modifiée jusqu'à 30% à chaque itération.

Ainsi, non seulement cette approche itérative aide à définir la portée finale du projet, mais elle peut également contribuer à une exécution plus rapide du projet en fournissant des avantages précoces et elle peut aider à mieux contrôler les projets incertains. Aguanno [54] déclare de la même manière que les principaux avantages de l'utilisation d'une approche agile sont la réduction des risques de ne pas définir la portée du projet et par conséquent le risque de qualité du produit, une meilleure maîtrise du projet, mais aussi une meilleure communication du projet.

D'un autre côté, les opposants à l'approche agile remarquent généralement que ces approches ne sont qu'une excuse pour ne pas utiliser les principes de base et nécessaires du développement logiciel et de la gestion de projets [59], et qu'il y a encore un manque de preuves empiriques d'une application réussie des méthodes agiles.

Mais, ces derniers temps, il y a de plus en plus de preuves issues de la recherche empirique d'une application réussie de l'approche agile [60]. L'une de ces recherches [61] a révélé que les facteurs de succès critiques de l'approche agile comprennent l'utilisation appropriée des méthodes agiles,

Une équipe de projet hautement qualifiée et une bonne stratégie de livraison, tandis qu'un processus de gestion, un environnement organisationnel et une implication des clients appropriés sont des facteurs susceptibles de contribuer au succès du projet. De même, Boehm et Turner [48] affirment que les défis les plus importants de la mise en œuvre agile sont contraintes, et donc de distinguer les obstacles dans les domaines des processus de

développement, des processus métier et de la gestion des personnes. Mais, la plupart d'entre eux sont une question de perception et ils peuvent être évités avec succès en reconnaissant et en comprenant les différences entre les approches prédictives et agiles, par une préparation minutieuse, de la patience et par le travail.

I.3.4. Comparaison entre méthodes prédictives et méthodes agiles

Les approches prédictives et agiles ont leurs avantages et leurs inconvénients, il n'est donc pas possible d'affirmer uniformément qu'une approche est meilleure qu'une autre [54]. Mais, il est souvent nécessaire d'utiliser les deux approches. Le besoin d'approches différentes de la gestion de projets pourrait être visible au sein de l'organisation au niveau du portefeuille de projets, en fonction des différentes catégories de projets en ce qui concerne les caractéristiques du projet, voire sur un seul projet dans l'utilisation de méthodes et techniques spécifiques, en fonction des demandes de phase de projet spécifique, et encore en fonction des caractéristiques du projet. Il faut garder à l'esprit la pertinence de l'approche d'un projet spécifique [47], car il est possible qu'une approche inappropriée ne contribue pas à la réussite du projet, mais au contraire, cela pourrait causer des problèmes supplémentaires et conduire à l'échec du projet.

L'approche prédictive est plus appropriée pour les projets avec des exigences utilisateur initiales claires et avec des objectifs de projet clairs, donc avec un très faible niveau d'incertitude. On s'attend à ce que ces projets aient un taux de changement des exigences très faible, et il n'est pas nécessaire d'impliquer fortement les utilisateurs finaux dans le projet [49]. Dans ces situations, l'accent sera mis sur la planification et, sur la base du plan initial, sur suivi prévisible et linéaire de ce plan de projet avec un objectif d'optimisation des activités du projet et d'efficacité dans leur exécution. L'approche prédictive est également appropriée pour les projets où une documentation formelle est requise à tout moment du projet [48]. Les projets typiques incluraient des projets de routine opérationnels avec une manière prévisible et vérifiée d'atteindre les objectifs du projet, comme les projets de construction ou d'ingénierie typiques [50].

Il est également à noter que les projets plus importants, peu importe si la taille est déterminée par le nombre de membres de l'équipe de projet, ou par la quantité et la

complexité d'exigences clairement définies [48], ou même par la durée [62], sont plus appropriées pour l'approche prédictive de gestion de projets. Comme déjà souligné, l'un des facteurs clés de succès de la sélection de l'approche est l'environnement organisationnel. D'une manière générale, l'organisation peut ne pas être préparée ou même ne pas vouloir mettre en œuvre de nouvelles approches, et la seule façon dans de telles situations est d'utiliser des processus existants, ce qui est dans la majorité des cas une approche prédictive [49]. En outre, les grandes organisations, avec un certain nombre d'unités organisationnelles impliquées dans des projets uniques, sont plus prêtes à utiliser l'approche prédictive [50] car cela l'approche met l'accent sur le contrôle du travail. Pour la même raison de contrôle, et le fait que l'importance du facteur humain n'est pas si accentuée dans l'approche prédictive, il est recommandé d'utiliser l'approche prédictive si les membres de l'équipe ne sont pas d'accord sur une approche différente, si les membres de l'équipe sont moins expérimentés, en cas de fluctuation des membres de l'équipe sont attendus au cours du projet, ou si les chefs de projet ne sont pas en contact quotidien avec les membres de l'équipe [62]. Enfin, il est recommandé d'utiliser l'approche prédictive si la criticité du système est l'une des caractéristiques clés du projet, lorsque les conséquences d'une défaillance du système peuvent être très graves [48]. Le tableau ci-dessous expliciter la différence entre les méthodes cycliques et les méthodes prédictives :

Description	Adapté au contrôle de:				Phases	Taille du projet (Petit, Moyen ou Grand)
	Cadre	Qualité	Délais	Coût		
Méthodes prédictives	Rational Unified Process	Y	Y	Y	Y	M, G
	PRINCE2	Y	Y	Y	Y	M, G
	System Development Life Cycle (SDLC)	Y	Y	N	*	P, M, G
	Solutions-based Project Methodology	Y	Y	N	N	P, M
	TenStep	Y	Y	Y	N	P, M
	Unicycle	Y	Y	Y	Y	P, M, G
	Code-and-fix Approach	N	N	N	N	P
	V-methodology	Y	Y	Y	Y	M, G
	Waterfall	Y	Y	Y	Y	M, G
	Spiral	Y	Y	N	N	M, G
Synchronize and Stabilize	Y	Y	N	N	M, G	

	Reverse Engineering Development	Y	Y	N	N	Y	M, G
	General Publication Methodology	Y	Y	N	*	Y	M, G
	Structured System Analysis & Design	Y	Y	N	N	Y	M, G

Tableau 1 : Comparaison de diverses méthodologies dans une perspective de gestion de projets [63]

Description		Adapté au contrôle de:				Phases	Taille du projet (Petit, Moyen ou Grand)
		Cadre	Qualité	Délais	Coût		
Méthodes agiles	Extreme Programming (XP)	N	Y	N	N	N	P, M
	Scrum	N	Y	N	N	N	P, M
	Crystal	N	Y	N	N	N	P, M
	Dynamic Sys. Development (DSDM)	Y	Y	Y	*	Y	P, M
	Rapid Applications Development (RAD)	Y	Y	Y	*	Y	P, G

Tableau 2 : Comparaison de diverses méthodologies dans une perspective de gestion de projets (Charvat J. , 2003)

En revanche, l'approche de gestion de projets agile s'adresse avant tout aux projets créatifs et innovants, comme les projets de recherche ou de développement de nouveaux produits innovants ou encore les projets d'amélioration des processus. Tous ces projets sont caractérisés par un niveau élevé d'incertitude, des objectifs de projet peu clairs ou des demandes incomplètes et imprévisibles, pour lesquelles on peut supposer qu'elles seront considérablement modifiées au cours du projet, mais d'un autre côté, avec un besoin et une vision métier clairs [64]. Encore une fois, en raison des demandes de changement constantes, les projets sont organisés de manière itérative, non linéaire, avec des modifications et mises à jour fréquentes du plan de projet et nécessitant une collaboration étroite et fréquente avec l'utilisateur final pendant le projet [47]. Cette approche itérative contribue également à une mise en œuvre rapide qui est nécessaire en raison de contraintes de temps serrées, et pour des raisons de meilleur suivi et contrôle du projet, les exigences sont organisées fonctionnellement (Boehm & Turner, 2005; Haas, 2007). Pour conclure, un projet agile

typique serait un projet de développement logiciel autonome plus petit, le plus souvent au sein d'une seule organisation, et généralement en mettant l'accent sur l'interface utilisateur.

Contrairement à l'approche prédictive, l'impact du facteur humain et surtout la communication entre les membres de l'équipe projet est accentué au point qu'il est recommandé que les membres de l'équipe projet soient très bons, sinon les meilleurs. Il est également recommandé que ces membres de l'équipe travaillent sur un site commun dans des équipes plus petites [65]. La conséquence est que les projets appropriés agiles ne mettent pas l'accent sur une documentation étendue, donc la connaissance du projet est principalement tacite.

En raison des différences significatives dans l'organisation du travail de projet par rapport à l'approche prédictive, l'environnement organisationnel a un impact significatif sur la mise en œuvre de l'approche de gestion de projets agile et l'organisation doit être prête à accepter les changements imposés par l'approche agile.

Ainsi, le tableau ci-dessous expliciter les différences fonctionnelles entre les méthodes prédictives et les méthodes agiles :

Caractéristiques	Approches prédictives	Approches agiles
Exigences	Exigences initiales claires; faible taux de changement	Créatif, innovant; exigences peu claires
Utilisateurs	Non impliqués (hors testing)	Collaboration étroite et fréquente
Documentation	Documentation officielle requise	Connaissances implicites
Taille du projet	Grands projets	Petits projets
Soutien organisationnel	utiliser les processus existants; grandes organisations	Prêt à adopter une approche agile
Membres de l'équipe projet	pas accentué; fluctuation attendue; équipe distribuée	Equipe co-localisée; petite équipe
Criticité du système	les conséquences d'une défaillance du système sont graves	Systemes moins critiques
Plan de projet	Liniaire	Complexe et itératif

Tableau 3 : Différence fonctionnelles entre l'approche prédictive et agile

I.4. Transformation digitale

I.4.1. Définition de la transformation digitale

La transformation digitale (ou numérique) marque une refonte radicale de la façon dont une organisation utilise la technologie, les personnes et les processus pour changer fondamentalement les performances de l'entreprise, déclare George Westerman, chercheur principal au MIT et auteur de *Leading Digital: Turning Technology into Business Transformation*.

Généralement dirigée par le PDG, en partenariat avec les DSI et d'autres hauts dirigeants, la transformation digitale nécessite une collaboration interdépartementale en associant des philosophies axées sur l'entreprise à des modèles de développement rapide d'applications.

Ces changements radicaux sont généralement entrepris à la recherche de nouveaux modèles commerciaux et de nouvelles sources de revenus, motivés par des changements dans les attentes des clients concernant les produits et services.

Comme exemple de la transformation digitale, il y'a plusieurs exemples dont on peut citer l'acquisition d'Assurance IQ par Prudential Financial en 2019 qui a indiqué un intérêt pour la prise en compte des préférences des consommateurs pour interagir avec les marques via les canaux numériques. L'assurance associe les acheteurs aux assurances vie, santé, assurance-maladie et automobile, leur offrant des options d'achat de produits en ligne ou avec l'aide d'un agent en direct.

Pour soutenir la stratégie numérique de Prudential, les ingénieurs de la société tirent parti des processus agiles et DevOps, collaborant avec des équipes interfonctionnelles comprenant des chefs de produits, des analystes et d'autres membres du personnel clé de l'entreprise, déclare Stacey Goodman, CIO. Le personnel met en place des capacités en libre-service pour aider à «tester et apprendre et échouer rapidement et créer des idées» lors de la création de nouvelles applications. Les ingénieurs s'appuient fortement sur Microsoft Teams et d'autres outils pour collaborer sur le code et d'autres aspects de leur travail.

Alors que le modèle hybride d'opérations de cloud et de centre de données de Prudential reste robuste face à la pandémie - les employés gèrent des environnements commerciaux depuis leur domicile - Goodman a pour objectif d'accélérer la mise sur le marché de nouveaux produits logiciels, ce qui, idéalement, améliorera les expériences des clients et des employés.

«Nous voulons être en phase avec cela pour être en mesure de répondre au marché», déclare Goodman, ajoutant qu'elle souhaite améliorer les capacités mobiles alors que Prudential continue de prendre en charge son modèle hybride permettant aux personnes de travailler depuis le bureau et à distance pendant la pandémie.

S'adapter au marché est un impératif essentiel pour Julie Averill depuis qu'elle a rejoint Lululemon en tant que CTO en 2017. Averill, qui a lancé plusieurs sites Web et applications mobiles internationaux depuis son arrivée chez le détaillant de vêtements, dit qu'elle migre la plupart des fonctions de calcul vers Microsoft Azure vers « offrir plus de flexibilité à nos clients. La mobilité est un objectif clé, car Lululemon s'efforce de permettre aux clients de rechercher son inventaire de produits à partir de leurs smartphones.

La transformation digitale se compose des composantes suivantes, voir la figure plus bas.

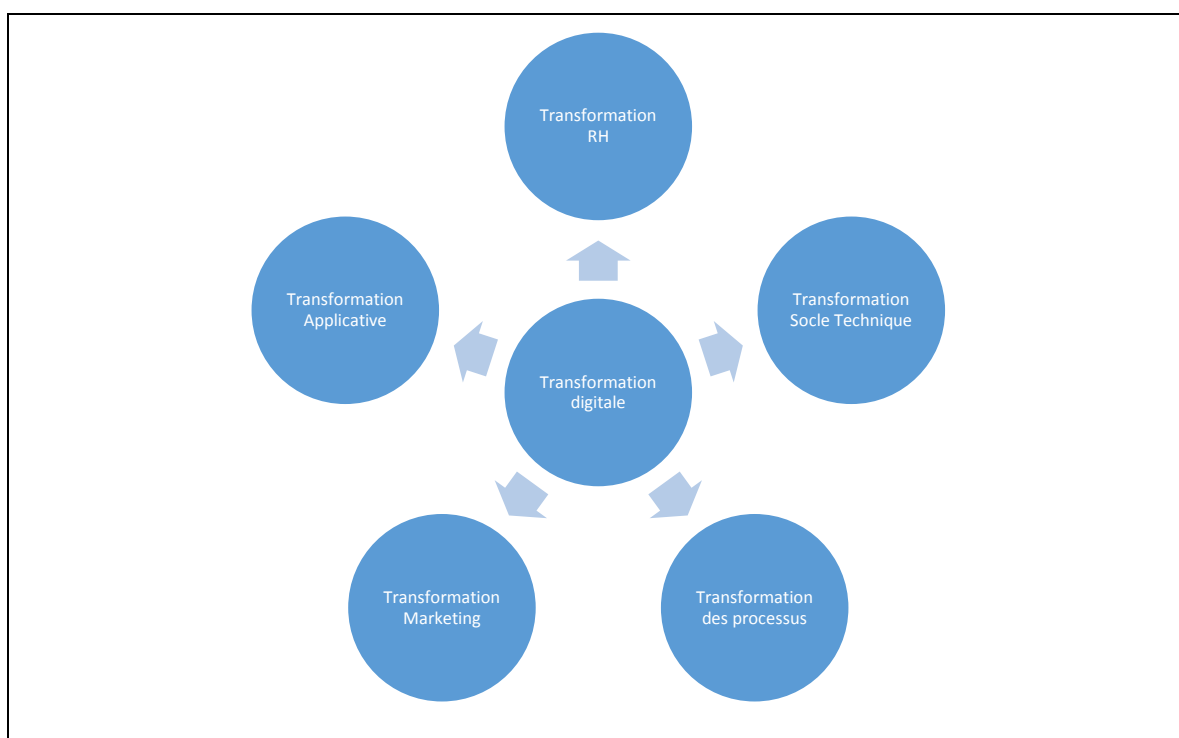


Figure 5 : Types de la transformation digitale

I.4.2. Relation avec la gestion de projets

Alors qu'une stratégie de transformation numérique commence dans la couche des dirigeants, l'exécution est à la main des chefs de projet. En tant que muscle exécutif de toute

organisation, les chefs de projet sont responsables de la mise en œuvre de la vision de la haute direction et du déploiement des nouvelles capacités numériques de l'entreprise.

Du point de vue du PM, la transformation numérique n'est pas quelque chose à craindre. Oui, cela prendra du temps, des efforts et un peu de perturbation. Mais cela vous permettra également en tant que chef de projet de mieux gérer les membres de l'équipe, de livrer plus efficacement les projets et, en fin de compte, de mieux servir les clients de votre entreprise.

Il existe quatre façons dont la transformation numérique change la gestion de projet :

1. Communication plus efficace et stratégique entre les équipes et les entreprises :

L'un des plus grands domaines dans lesquels la technologie numérique moderne redéfinit la gestion de projet est la communication entre équipes. Les outils traditionnels comme le courrier électronique sont encombrants en matière de collaboration, car ils ne sont pas conçus pour un dialogue en temps réel. Les informations importantes sont facilement enfouies dans des chaînes de messagerie sans fin et la productivité constante des surcharges de courrier électronique.

Le logiciel de gestion du travail collaboratif, quant à lui, permet aux membres de l'équipe et aux collègues de tous les services de s'engager, de se connecter et d'interagir en temps réel, réduisant considérablement l'encombrement des e-mails et économisant beaucoup de temps dans le processus. Mais plus qu'augmenter l'efficacité de la communication intra-professionnelle, ces technologies améliorent également son efficacité.

Lorsque les membres de l'équipe sont libérés du filtrage de centaines d'e-mails par jour juste pour suivre l'état d'un projet, ils peuvent passer plus de temps à discuter de la stratégie du projet, ce qui est précisément là où vous voulez que l'essentiel de la conversation de votre équipe se concentre.

2. Plus de collaboration au sein des équipes et d'appropriation par les membres de l'équipe

En plus de faciliter une communication plus efficace et axée sur la stratégie, les technologies modernes de gestion du travail facilitent la collaboration réelle des équipes. Avec la bonne plateforme en place, les cadres, les chefs de projet et les membres de l'équipe peuvent ajouter des commentaires, attribuer des tâches, organiser des tableaux de bord,

vérifier et approuver les actifs et gérer à peu près tout le reste du projet dans une seule solution pratique.

Ce niveau de collaboration profond conduit inévitablement à un plus grand sentiment d'appropriation partagée de la part des coéquipiers et contribue à favoriser un environnement coopératif et synergique. Et c'est là, comme on dit, que la magie opère. Il a été démontré que les travailleurs qui ont le sentiment de faire partie d'un effort de collaboration ont un plus grand engagement, une moindre fatigue et des taux de réussite plus élevés que ceux qui sont isolés de leurs pairs. En fait, la même étude a révélé que les collaborateurs persistaient à accomplir des tâches 64% plus longtemps!

3. Mettre davantage l'accent sur les résultats que sur le processus

Alors que la transformation numérique automatise les flux de travail et coordonne les tâches de gestion de projet traditionnelles telles que la planification, les PM ont plus de temps pour se concentrer sur l'optimisation de la stratégie et la livraison de projet. En fait, le PMI prédit que, alors que la transformation numérique continue de toucher les entreprises de tous les secteurs et de tous les secteurs, les chefs de projet seront davantage considérés comme des leaders stratégiques dans leurs organisations:

Avec davantage d'outils numériques et de processus automatisés à leur disposition, les chefs de projet recherchent les meilleures façons d'aligner chaque projet sur les stratégies et les objectifs de leur entreprise - et d'obtenir des résultats plus positifs dans le processus.

4. Davantage d'analyses pour améliorer les processus de gestion de projet, les résultats et le retour sur investissement

Enfin, la transformation numérique fournit aux PM la technologie analytique pour prendre des décisions basées sur les données, décomposer les modèles et les tendances, et finalement améliorer les résultats des projets et les taux de réussite. Cet accès aux données approfondies aide également les dirigeants et les gestionnaires à prendre des décisions mieux informées plus rapidement et plus facilement que jamais.

Des rapports analytiques robustes aident les gestionnaires à maintenir les projets sur la bonne voie et dans le budget avec des analyses en temps réel des coûts et de la main-d'œuvre. Des ensembles de données approfondis peuvent également être facilement décomposés pour les parties prenantes et les cadres, leur donnant un aperçu précis de l'impact commercial et du

retour sur investissement de chaque projet et les aidant à planifier les initiatives futures et à prendre des décisions stratégiques critiques.

I.5. Ingénierie des modèles

« Il n’y a pas une vérité, il y’a que des points de vues ». En d’autre terme nous, notre habilité à voir, à interpréter est une résultante d’une conception de la vie. Par conséquent nous ne pouvons que percevoir des modèles conçus et émettre des points de vue. L’être humain est donc adapté à modéliser et reproduire les éléments qui les entourent pour mieux les étudier et les optimiser, ainsi il a créé des programmes et des machines qui l’aide dans ses divers travaux.

L’émergence de l’informatique de nos jours, a contribué à sa banalisation démocratique qui la range peu à peu au rang des autres industries. Face à cette situation, les chercheurs et les industriels se sont mis d’accord sur le fait que la montée en puissance des modèles et la séparation entre le métier et la technologie sera un très bon atout pour développer ce domaine, aussi le fait d’utiliser la modélisation comme un outil de production massive plutôt qu’une simple tâche de documentation. Cette approche a permis la naissance de la notion d’Ingénierie Dirigée par les Modèles IDM ou (MDE Model Driven Engineering).

I.5.1. Définition

L’Ingénierie Dirigée par les Modèles (IDM) est une discipline récente du génie logiciel qui promeut les modèles en entités de première classe dans le développement logiciel. Elle fait l’objet d’un grand intérêt aussi bien de la part des équipes de recherche académiques (Action IDM. [47]) que des laboratoires industriels [66], [67], [68], [69], etc... C’est une forme d’ingénierie générative par laquelle tout ou partie d’une application informatique est générée à partir de modèles [70]. Dans cette nouvelle perspective, les modèles occupent une place de premier plan parmi les artefacts de développement des systèmes et doivent en contrepartie être suffisamment précis et riches afin de pouvoir être interprétés ou transformés par des machines. Le processus de développement des systèmes peut alors être vu comme une séquence de transformations de modèles partiellement ordonnées. Chaque transformation prend un ou des modèles en entrée et produit un ou des modèles en sortie, et ce, Jusqu’à l’obtention d’artefacts exécutables.

Dans la suite de cette section, nous présentons d'abord les définitions des concepts de base de la MDA & l'IDM. Par la suite, nous décrivons les approches de l'OMG [71] et aussi du Model Integrated Computing (MIC) [72] et les usines logicielles de Microsoft (Software factories) [73] qui sont aussi des manières d'aborder l'IDM.

I.5.2. Niveaux d'abstraction

Un modèle est une abstraction d'un système, décrit sous la forme d'un ensemble de faits construits dans une intention particulière. Un modèle doit pouvoir être utilisé pour répondre à des questions sur le système modélisé. Selon Bran Selic [74], pour que le modèle soit utile et efficace, il doit posséder les caractéristiques suivantes :

- Avoir un certain niveau d'abstraction qui le rend indépendant des technologies de mise en œuvre.
- Être exprimé dans une forme assez compréhensive pour qu'il soit compris de manière intuitive.
- Fournir une représentation précise et fidèle de caractéristiques réelles du système modélisé.
- Être moins coûteux et plus facile à développer que le système qu'il représente.

Pour que ces modèles possèdent ces caractéristiques, il faut disposer des techniques précises de mise en œuvre. Pour cela, il faut que les modèles soient conformes à un langage de modélisation clairement défini, appelé le méta-modèle.

Un **méta-modèle** est une abstraction d'un langage permettant de décrire des modèles. Il définit les éléments que l'on va retrouver dans le langage utilisé pour écrire un modèle ainsi que les relations qu'il existe entre ces éléments. Dans l'IDM, le principe de base est "tout est modèle" [75]. Par conséquence, le méta-modèle est un modèle. Il doit aussi être conforme à son méta-modèle, appelé le méta-méta-modèle.

Un **méta-méta-modèle** est un modèle d'un langage de modélisation permettant de décrire des méta-modèles. Ce méta-méta-modèle est aussi un modèle qui doit être conforme à un méta-modèle. Pour arrêter ce schémas infiniment récursif le méta-méta-modèle s'auto définit, c'est à dire qu'il est conforme à lui-même. MOF (Meta-Object Facility) [76]

standardisé par l'OMG est un tel méta-méta-modèle. Il définit des concepts de base comme classe et relation qui peuvent tout définir y compris eux-mêmes.

Dans l'IDM, deux relations fondamentales peuvent être identifiées entre les différents niveaux de modélisation [77]. D'après la définition d'un modèle, nous pouvons déduire la première relation qui lie le modèle et le système qu'il représente. Cette relation est appelée "Représente". La deuxième relation peut être déduite de la définition du méta-modèle. C'est une relation liant le modèle et le langage utilisé pour le construire, elle est appelée "Conforme".

I.5.3. Approche MDA

Après la technologie procédurale, la technologie objet et la technologie des composants, l'approche MDA (Model Driven Architecture) est un processus de l'ingénierie dirigée par les modèles (ou MDE pour Model Driven Engineering). Proposée par l'OMG (Object Management Group) en 2000, l'approche MDA est basée sur la séparation des préoccupations. Elle permet prendre en compte, séparément, aspect métier et aspect technique d'une application, grâce à la modélisation. Le code source de l'application est obtenu par génération automatique à partir des modèles de l'application. Les modèles ne sont plus seulement un élément visuel ou de communication, mais sont, dans l'approche MDA, un élément productif et le pivot du processus MDA.

I.5.4. Approche IDM

Face à la complexité croissante des applications, les modèles développés sont devenus de plus en plus difficiles à réaliser et à maintenir. De plus, la variété et l'évolution rapide des plates-formes d'exécution ont accentué le besoin de réutilisation et d'interopérabilité des modèles. En 2000, l'OMG a lancé son initiative IDM (ingénierie dirigée par les modèles) [78] afin de promulguer de bonnes pratiques de modélisation et d'exploiter pleinement les avantages des modèles. Cette approche vise à mettre en valeur les qualités intrinsèques des modèles, telles que la portabilité, l'interopérabilité, la maintenance et la réutilisation.

C'est à la fois une architecture et une démarche de développement. L'idée de base de l'IDM est de séparer les spécifications fonctionnelles d'un système des détails de son implémentation sur une plate-forme donnée. Pour cette raison, l'IDM définit une architecture

de spécification structurée en quatre types de modèles [79]: Les modèles indépendants de l'informatisation appelés CIM (Computational Independent Model), les modèles indépendants des plates-formes appelés PIM (Platform Independent Model), les modèles spécifiques à une plate-forme appelés PSM (Platform Specific Model), les modèles descriptifs de plateforme appelés PDM (Platform Description Model). Un CIM modélise les exigences d'un système, son but étant d'aider à la compréhension du problème mais aussi de fixer un vocabulaire commun pour un domaine donné. Dans UML, le diagramme des cas d'utilisation est un bon candidat pour représenter un CIM. Le PIM, connu aussi sous le nom de modèle d'analyse et de conception, est un modèle abstrait indépendant de toute plate-forme d'exécution. Il a pour but de décrire le savoir-faire ou la connaissance métier d'un domaine. Le PDM modélise la plate-forme Sur laquelle le système va être exécuté (ex. modèles de composants à différents niveaux d'abstraction PHP, EJB, etc.). Plus précisément, il définit les différentes fonctionnalités de la plate-forme et précise comment les utiliser.

Ayant isolé le savoir-faire métier dans des PIMs, on a besoin soit de transformer ces modèles en d'autres PIMs pour des besoins d'interopérabilité, soit de produire des modèles PSM ciblant une plate-forme d'exécution spécifique en se basant sur les PDMs dans le but d'améliorer la portabilité et augmenter la productivité. En se basant sur l'approche IDM, la réalisation du portage d'une application d'une plate-forme vers une autre se concrétise par une transformation de modèles qui prend comme entrée la description de l'application (PIM) sera guidée par le modèle de la nouvelle plate-forme cible (PDM). Ainsi, il suffit de garder le même modèle PIM et de varier les PDMs pour générer différents PSMs. La figure 2-5 illustre la description d'une telle approche.

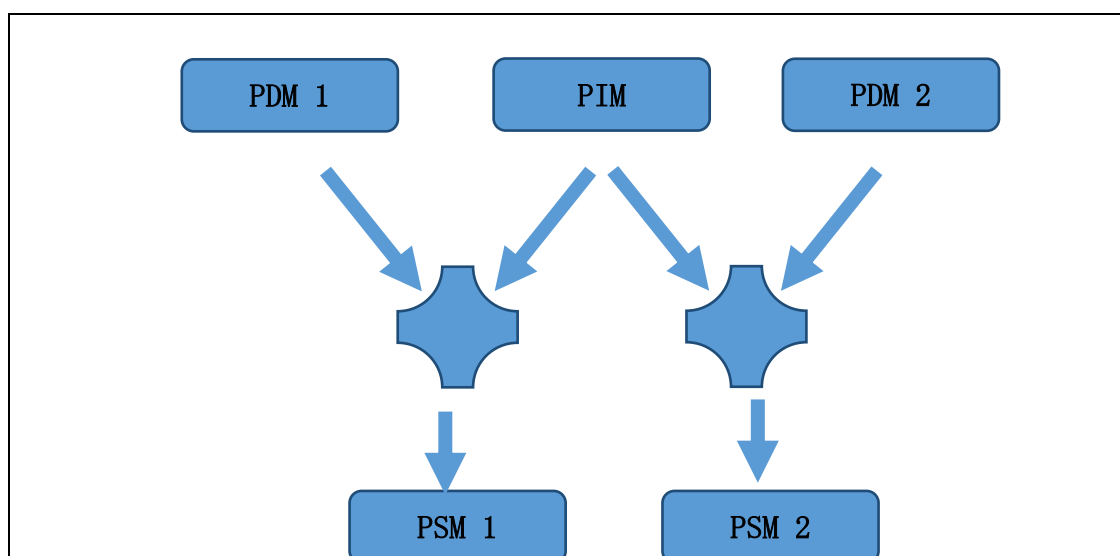




Figure 6 : Modèle de l'approche IDM

Cependant, comme l'approche IDM utilise le concept de plate-forme sans en donner une définition précise, cela ouvre la possibilité à plusieurs interprétations de la notion de description de la plate-forme (PDM). En effet, lors du passage du PIM vers PSM à l'aide d'une transformation, le PDM peut être décrit soit implicitement dans la transformation et donc le PDM fait partie intégrante de la transformation, soit il est décrit explicitement dans des modèles à part, et il sera alors passé en entrée à la transformation, avec le PIM.

I.5.1. Standards de l'OMG

L'OMG est un consortium regroupant des industriels, des fournisseurs d'outils, et des académiques dont l'objectif principal est de développer des standards pour normaliser les différentes approches d'un domaine. Parmi les standards OMG utilisés dans l'IDM, il y'a le Standard MOF (Meta-Object Facility) qui permet d'établir de nouveaux langages de modélisation ; le standard UML (Unified Modeling Language) qui permet la modélisation des systèmes logiciels; le standard OCL (Object Constraint Language), pour exprimer des contraintes (invariant, précondition, post-condition) sur des modèles ; le standard XMI (XML Metadata Interchange) qui permet de représenter les modèles sous forme de documents XML pour des besoins d'interopérabilité ; le Standard DI (Diagram Interchange) qui permet la représentation au format XML des parties graphiques d'un modèle ; le standard SPEM (Software & Systems Process Engineering Metamodel) dédié à la modélisation des processus logiciels et systèmes et l'approche IDM. Ces standards sur lesquels repose l'IDM sont centrés sur les notions de méta-modèles et de méta-méta-modèles, déjà définie, et se trouvent sur différents niveaux dans l'architecture à quatre niveaux de OMG. Dans cette architecture (voir Figure 2-4), le monde réel est représenté à la base de la pyramide (niveau MO). Les modèles représentant cette réalité constituent le niveau M1. Les méta-modèles permettant la définition de ces modèles (UML. SPEM. etc.) constituent le niveau M2. Enfin, le méta-méta-modèle (MOF), unique et méta-circulaire, est représenté au sommet de la pyramide (niveau M3). La figure suivante présente l'architecture expliquée :



Figure 7 : Architecture à 4 niveaux de l'approche IDM

I.5.2. Transformation de modèles

La définition de transformations a pour objectif de rendre les modèles opérationnels dans une approche IDM. Ce qui augmente la productivité des applications. La transformation de modèles est une opération très importante dans toute approche orientée modèle. En effet, les transformations assurent les opérations de passage d'un ou plusieurs modèles d'un niveau d'abstraction donné vers un ou plusieurs autres modèles du même niveau (transformation horizontale) ou d'un niveau différent (transformation verticale [80]). On peut citer comme exemple de transformation verticale, la transformation PIM vers PSM et comme exemple de transformation horizontale, la transformation PIM vers PIM ou PSM vers PSM. Le modèle transformé est appelé modèle source et le modèle résultant de la transformation est appelé modèle cible. Une classification discutée dans [81] distingue deux types de transformation une transformation modèle vers modèle et une transformation modèle vers code. En général la transformation modèle vers code peut être vue comme un cas particulier d'une transformation modèle vers modèle : il suffit juste de définir le méta-modèle du langage de programmation cible.

Parmi les approches de transformations modèles nous distinguons l'approche par programmation, l'approche par template et l'approche par modélisation ou par règles. Ces trois approches discutées dans [80] sont détaillées dans les sections qui suivent.

I.5.2.1. Approches de transformation de modèles

I.5.2.1.1. Approche par programmation

L'approche par programmation utilise les langages de programmation en général, et plus particulièrement les langages orientés objet. Cette approche décrit la transformation sous forme d'un programme à l'image de n'importe quelle application informatique. Cette approche reste très utilisée car elle réutilise l'expérience accumulée et l'outillage des langages existants. Cette approche est utilisée dans les outils tels que Softeam MDA Modeler et IBM Rational Software Modeler [81].

I.5.2.1.2. Approche par Template

L'approche par template consiste à définir des modèles cibles en y déclarant des paramètres. Ce modèle paramétré appelé aussi modèle template peut être dans un format graphique (template UMIL par exemple) ou textuel (template basés sur XMI ou un langage de programmation). L'exécution d'une transformation consiste à prendre un modèle template et à substituer ses paramètres par les informations contenues dans le modèle source. Cette approche par template est implémentée par exemple dans Softeam MDA Modeler. La notion de template est très utilisée dans le développement des sites web basés sur PHP ou JSP. En effet, dans ces langages des pages dynamiques sont créés à partir des canevas de pages (templates) qui contiennent des paramètres qui font référence par exemple aux données d'une base. L'approche par template est aussi supportée par les outils Softeam MDA Modeler et IBM Rational Software Modeler.

I.5.2.1.3. Approche par modélisation

L'approche par modélisation ou par règles consiste à appliquer les principes de l'ingénierie des modèles aux transformations. L'objectif est de modéliser les transformations de modèles et de rendre les modèles de transformation pérennes et productifs, en exprimant leur indépendance vis-à-vis des plates-formes d'exécution. Le standard MOF 2.0 QVT de l'OMG est une illustration parfaite de l'approche par modélisation et a pour but de fournir un moyen standardisé pour établir les transformations de modèles. L'approche par modélisation est aussi choisie par le groupe ATLAS pour le langage ATL que nous allons voir dans le prochain titre. Dans l'approche par modélisation, la transformation se fait par l'intermédiaire de règles de transformations qui décrivent la correspondance entre les entités du modèle source et celles du modèle cible. La transformation se situe en réalité, au niveau des méta-modèles source et cible qui décrivent la structure des modèles cible et source. La Figure plus

haut détaille un exemple d'application de l'approche par modélisation. Dans cet exemple, une transformation d'un modèle source est constituée de deux étapes. La première étape consiste à décrire le modèle de transformation appelé aussi règles de la transformation. Ces règles sont spécifiées sur la base des méta-modèles source et cible. La seconde étape consiste à générer automatiquement le modèle cible par l'exécution des règles de la transformation.

I.5.2.2. Langages de transformation de modèles

I.5.2.2.1. ATL

ATL est l'approche officielle pour les transformations de modèles dans Eclipse. Il s'agit d'un langage de transformation créé par l'équipe Atlas de l'Inria pour répondre à l'appel à propositions de l'OMG qui a lancé le processus d'élaboration de la norme QVT. Les différentes réponses reçues ont convergé vers un document unique, la norme QVT finale. Les auteurs d'ATL n'ont pas accepté tous les compromis faits dans le cadre de cette évolution et c'est pourquoi ATL n'a pas convergé vers le même résultat que QVT.

ATL représente les modèles d'entrée et de sortie par leurs arbres de syntaxe abstraite respectifs. Les transformations ATL sont écrites dans un langage textuel qui présente un système de règles dont la syntaxe abstraite est définie par un méta-modèle écrit en Ecore. ATL propose deux types de règles pour transformer un modèle :

- **Matched rules**, avec une syntaxe déclarative, correspondent aux règles du langage Relationnel de QVT, filtrent un nœud du modèle d'entrée et créent un élément correspondant dans le modèle de sortie.
- **Called rules**, avec une syntaxe impérative, correspondent aux règles du langage Opérationnel de QVT et sont appelées explicitement.

En plus des règles, ATL permet d'écrire des requêtes (queries) vers les nœuds du modèle et d'enrichir (par des **helpers**) les classes des nœuds des modèles à transformer.

Ci-dessous un schéma explicatif de la transformation ATL.

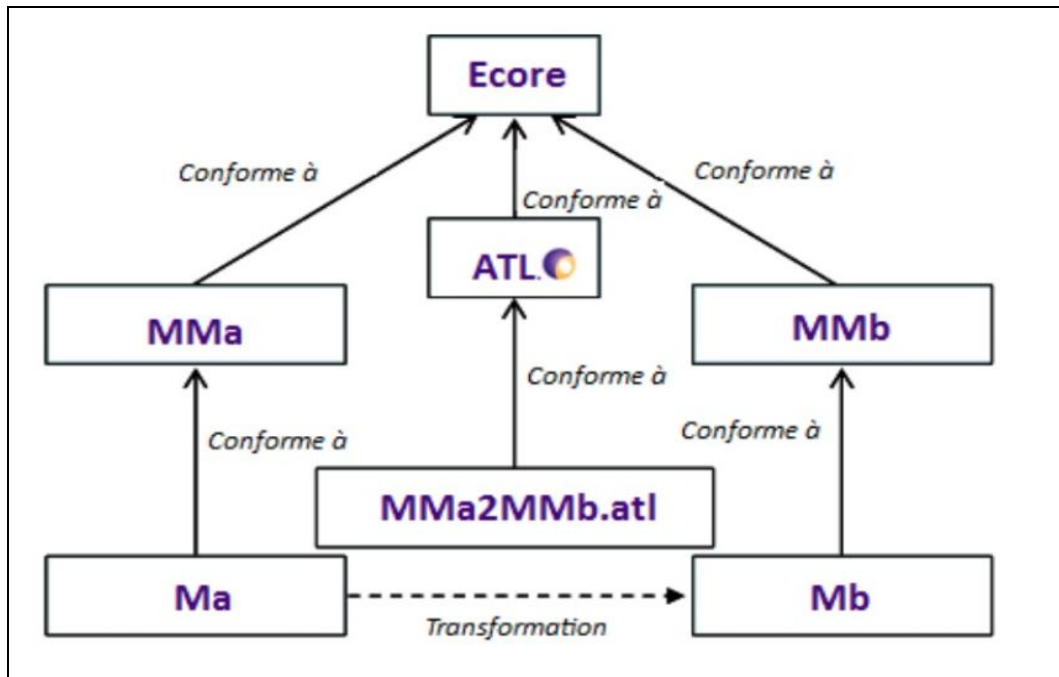


Figure 8 : Vue d'ensemble sur la transformation ATL [90]

ATL est parfaitement adapté lorsque les structures des modèles source et cible sont proches et que la transformation ne demande pas de parcours complexes ainsi par la suite nous avons opter pour utiliser le langage ATL.

I.5.2.2.2. QVT

Le QVT (Query, Views, Transformations) est la standardisation de la transformation de modèles et de spécification des vues et requêtes sur les modèles basé sur MOF défini par l'OMG [91] et [92].

Ce langage est défini sous trois niveaux :

- **QVT-Relation** : Langage déclaratif (Prolog), il s'agit langage déclaratif conçu pour permettre l'écriture de transformations de modèles unidirectionnelles et bidirectionnelles. Une transformation incarne une relation de cohérence sur des ensembles de modèles.
- **QVT-Core** : la sémantique des concepts déclaratifs (Pascal, C), il s'agit d'un langage déclaratif conçu pour être simple et servir de cible de traduction à partir de QVT-Relations.
- **QVT-Opérationnel** : langage hybride, il s'agit d'un langage impératif conçu pour écrire des transformations unidirectionnelles.

Il existe aussi le QVT-BlackBox, c'est un mécanisme pour invoquer des fonctionnalités de transformation exprimées dans d'autres langages (par exemple XSLT ou XQuery).

Ci-dessus l'architecture du standard QVT :

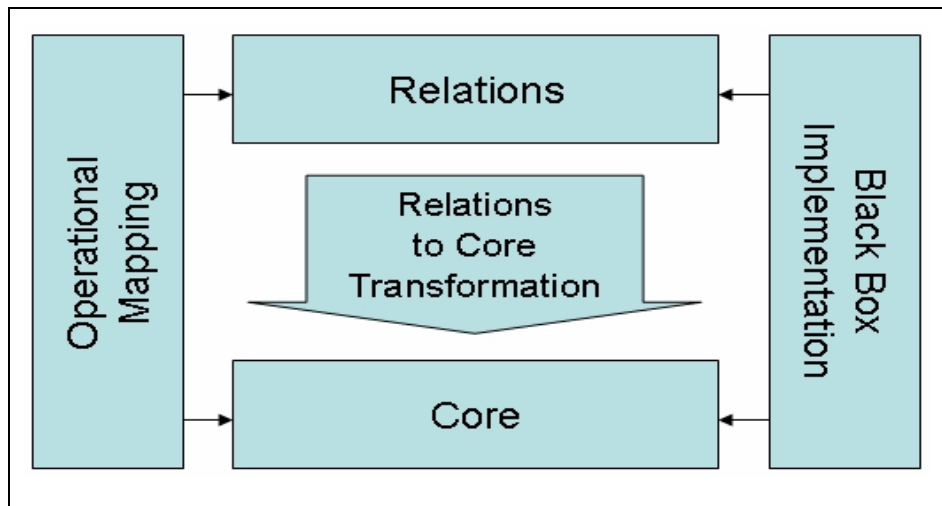


Figure 9 : Vue d'ensemble sur la transformation ATL

I.5.2.2.3. Xtext

Xtext [94 ; 96] est un environnement permettant la création des modèles à partir d'une définition de syntaxe abstraite décrite en EMF et une syntaxe concrète basée sur une grammaire textuelle. Il permet la création d'éditeurs de texte extrêmement puissants (avec complétion de code, vérification des erreurs et coloration de la syntaxe) à partir d'une définition améliorée de la grammaire en utilisant le formalisme EBNF. Inversement, Xtext permet aussi de génère un métamodèle à partir d'une définition de la syntaxe concrète d'un langage de modélisation. Dans les deux cas de figures, cet environnement génère automatiquement un analyseur syntaxique qui analyse des expressions saisies via l'éditeur du langage et crée une instance de modèle correspondante. Xtext profite de la puissance de l'écosystème des outils autour du projet EMP, puisqu'il utilise Eclipse EMF comme base pour ses méta-modèles. Ainsi, il peut être utilisé avec n'importe quel outil de génération de code et de transformation de modèle basé sur EMF (Xpand [97], ATL [84], etc.).

Xtext fournit des mécanismes d'extension et de composition de modélisation. La composition est accessible à travers la possibilité de référencer des concepts entre modèles à partir de la description de la syntaxe concrète. Alors que l'extension permet à un modèle d'hériter d'un autre et de remplacer ses règles de grammaires syntaxiques. Cependant, ce dernier mécanisme ne permet que de remplacer complètement les règles héritées par des

nouvelles règles et ne permet pas de les surcharger. En outre, ce mécanisme limite les connexions entre modèles à des liens unidirectionnels.

La réutilisation, l'extension et la composition des modèles dans Xtext restent donc assez limitées. Néanmoins, la complétion de code pour les références dans d'autres modèles ainsi que les contrôles de cohérence entre modèles et entre langages dans l'éditeur sont pris en charge immédiatement avec le moteur Xtext. En ce qui concerne la sémantique des modèles, Xtext permet de la définir en utilisant d'autres langages tels que Xtend [95 ; 98] que nous allons voir par la suite.

I.5.2.2.4. Xtend

Xtend est un langage de programmation de haut niveau à usage général pour la machine virtuelle Java. Sur le plan syntaxique et sémantique, Xtend a ses racines dans le langage de programmation Java mais se concentre sur une syntaxe plus concise et certaines fonctionnalités supplémentaires telles que l'inférence de type, les méthodes d'extension et la surcharge d'opérateurs. Étant principalement un langage orienté objet, il intègre également des fonctionnalités connues de la programmation fonctionnelle, par ex. expressions lambda. Xtend est typé statiquement et utilise le système de types de Java sans modifications. Il est compilé en code Java et s'intègre ainsi de manière transparente à toutes les bibliothèques Java existantes.

Le langage Xtend utilise principalement l'outil Xpand, il est intégré dans l'environnement EMF. Il permet la génération de code en partant d'un modèle. Ce dernier doit être conforme à un méta-modèle. Un fichier template Xpand (d'extension « .xpt ») comporte des instructions d'import qui servent à importer le méta-modèle, de zéro ou plusieurs extensions avec le langage Xtend et d'un ou plusieurs blocks DEFINE.

I.6. Conclusion

Dans ce chapitre, nous avons défini la gouvernance IT, ses principaux composants, sa relation avec la gouvernance entreprise et les référentiels qui tournent autour de cette notion. Aussi, nous avons effectué un état de l'art des différentes méthodes de gestion de projets, puis effectué une comparaison entre les méthodes et nous avons parlé de la transformation digitale et l'ingénierie des modèles. Dans le chapitre suivant, nous allons présenter une étude

comparative multicritère des différents référentiels de gouvernance IT avec un cas d'étude pour comprendre mieux l'utilité de notre comparaison.

Chapitre II : Proposition d'une méthode pour le choix d'un référentiel de la gouvernance IT

II.1. Introduction

Dans ce chapitre, nous réaliserons une étude comparative entre les différents référentiels en précisant leurs forces et leurs faiblesses puis nous mettrons en place un système de décision basé sur l'analyse multicritères des référentiels de la gouvernance IT que nous avons développé ensuite sous forme d'application aidant les gestionnaires dans la prise de décision. Ainsi, les entreprises peuvent donner les poids des critères et avoir le choix des référentiels à adopter en fonction de leurs propres besoins.

II.2. Analyse SWOT appliquée à la gouvernance IT

II.2.1. Définition de la méthode SWOT

L'analyse SWOT ou matrice SWOT est un outil stratégique permettant de déterminer les points forts ainsi que les points faibles d'une solution ou d'une option donnée. Le terme SWOT est un acronyme issu de l'anglais : **S**trengths (forces), **W**eaknesses (faiblesses), **O**pportunities (opportunités) and **T**hreats (menaces). La connaissance des forces et des faiblesses d'une solution permet de définir une stratégie claire de développement. À ce titre, l'analyse SWOT fait partie des outils contribuant à l'étude de la pertinence et de la cohérence d'une action future notamment pour notre cas la définition des forces et faiblesses des référentiels de la gouvernance IT.

II.2.2. Application de l'analyse SWOT

L'analyse SWOT est une méthode couramment utilisée pour analyser et positionner les ressources et l'environnement d'une organisation selon quatre régions : Forces, Faiblesses, Opportunités et Menaces [82]. Forces et les faiblesses sont des facteurs internes (contrôlables) qui soutiennent et empêchent les organisations de réaliser leur mission respectivement. Alors que les opportunités et les menaces sont les facteurs externes (incontrôlables) qui permettent et empêchent les organisations d'accomplir leur mission [83]. En identifiant le facteur dans ces quatre domaines, l'organisation peut reconnaître ses compétences pour la prise de décision, la planification et l'élaboration de sa stratégie.

Dans notre étude nous allons nous focaliser sur les facteurs internes, nous allons présenter les forces et les faiblesses des référentiels de la gouvernance IT sans présenter les opportunités et les menaces puisque ce sont des facteurs externes ; car nous avons besoin d'analyser ses référentiels et non l'environnement changeant qui nous entoure.

II.2.3. Matrice SWOT minimale des référentiels de la gouvernance IT

Dans cette section, nous présentons une analyse SWOT minimale afin de résumer les forces et les faiblesses des références :

CMMI	
Positifs	Améliorer la qualité du produit livré et la productivité du projet
	Augmenter la satisfaction client en répondant mieux à ses exigences
	Réduisez les coûts et respectez les délais
	Donner une meilleure visibilité à la direction et permettre une meilleure gestion des risques
Négatifs	Le manque de précision.
	Le niveau de maturité étant global, il peut masquer des domaines de l'organisation moins performants que d'autres et masquer des lacunes dans certains domaines de processus.

Tableau 4 : Matrice SWOT minimale du CMMI

COBIT	
Positifs	Répondre aux besoins des parties prenantes
	Couvrir toute l'entreprise de bout en bout
	Application d'un cadre unique
	Fournir une approche holistique de la prise de décision commerciale
	Séparer la gouvernance de la gestion
Négatifs	Difficulté de mise en œuvre
	Guide de gestion inconnu dans le framework

Tableau 5 : Matrice SWOT minimale de COBIT 5

TOGAF	
Positifs	Un langage commun au sein de l'entreprise
	Force du système d'information (comme la complexité croissante du SI)
	Maximisez la valeur informatique
	Utiliser un cadre commun pour faciliter la recherche de compétences
	Obtenez une meilleure qualité des produits
Négatifs	Ne couvre pas les processus de gestion

Tableau 6 : Matrice SWOT minimale du TOGAF

ISO 38500	
Positifs	Assurer à toutes les parties prenantes (y compris les clients, les actionnaires et les employés) que si la norme est appliquée, elles peuvent faire confiance à la gouvernance informatique de leur organisation
	Informé et guider les dirigeants pour orienter l'utilisation des ordinateurs dans

	leur organisation
	Fournir un cadre pour une évaluation objective de la gouvernance informatique de l'entreprise
Négatifs	Framework conçu pour le top management.
	Ne couvre pas les services de support, les services de construction et de mise en œuvre

Tableau 7 : Matrice SWOT minimale de l'ISO 38500

ITIL	
Positifs	Un gain de temps
	Réduction des coûts
	Définir des rôles et des responsabilités plus précis
	Meilleure satisfaction des utilisateurs
	Meilleure productivité / efficacité
	Des services informatiques de meilleure qualité
	Adaptation aux besoins des clients facilitée
Négatifs	Très peu connu du grand public
	Il y a très peu d'informations sur Internet
	Complexe (il faut être expert car il concerne le réseau, le système, l'application, le BD et avoir une vision globale du SI)

Tableau 8 : Matrice SWOT minimale du référentiel ITIL

PMBOK	
Positifs	Un guide de connaissances et de bonnes pratiques créé par des professionnels de la gestion de projets qui le mettent à jour régulièrement.
	Standardiser les pratiques de gestion de projets, ce qui signifie que chaque département fonctionne de la même manière.
	Retrouvez les mêmes pratiques d'une entreprise à l'autre.
Négatifs	Respecter tous les processus PMBOK pour assurer la réussite du projet
	Adapter la méthodologie à la taille et au secteur du projet

Tableau 9 : Matrice SWOT minimale de PMBOK

PRINCE 2	
Positifs	Gestion de projets rationnelle.

	Le formalisme permet de définir une logique et un vocabulaire commun facilitant les échanges
	Orientation vers l'apprentissage continu
Négatifs	La rationalisation systématique masque les réelles subtilités de la gestion de projets, dans la pratique; la gestion de projets est beaucoup plus complexe. Les réalités du terrain, les enjeux humains, les technologies immatures, les exigences de délais et de budgets ne seront pas résolus par l'obsession du formalisme

Tableau 10 : Matrice SWOT minimale de PRINCE 2

SCRUM	
Positifs	Augmente la fluidité de la libération et la vitesse de l'équipe
	Maîtrisez facilement les risques et les changements lors de la livraison
	Encourager l'orientation, la rigueur et l'énergie dans les équipes
	Augmentez la capacité et la qualité d'exécution par rapport aux exigences du client
	Surpasse considérablement la livraison des valeurs de priorité qui génèrent des résultats plus rapidement
Négatifs	Nécessite une équipe unie et motivée
	Difficulté à suivre le cycle de vie d'un développement
	Soutien nécessaire aux sponsors
	Phase d'adaptation essentielle (changements continus)

Tableau 11 : Matrice SWOT minimale de SCRUM

Il existe plusieurs méthodes possibles pour faire la comparaison entre les cadres de références, nous allons par la suite exposer une étude comparative dans le sous-titre qui suit.

II.3. Etude Comparative Multicritère des référentiels de la gouvernance IT

II.3.1. Méthodes d'analyse multicritère

Il existe plusieurs méthodes possibles pour faire la comparaison entre les cadres en utilisant un certain nombre de critères. Ces méthodes peuvent être divisées en trois familles principales :

- Agrégation complète (approche descendante) : agréger les n critères pour les réduire à un seul critère.

- Agrégation partielle (approche ascendante) : comparer les actions ou classements potentiels entre eux et établir entre eux des relations de sur-classement.
- Agrégation locale et itérative : En recherchant principalement une solution de départ, nous procédons ensuite à une recherche itérative pour trouver une meilleure solution.

Le tableau suivant présente les différentes méthodes multicritères existantes triées par famille :

Famille	Approche	Méthodes
Agrégation complète	approche descendante	Weighted Sum Method (WSM) TWO WAY ANOVA Weight Product Method (WPM) Analytic Hierachy Process (AHP) Multi Atribute Utility Theory (MAUT)
Agrégation partielle	approche en profondeur	ELECTRE Prométhée Melchior Qualifex Oreste Regim ...
Agrégation locale et itérative	Local et itératif	Improving Cones Method (ICM) GOAL Programming STEM Branch and Bound

Tableau 12 : Familles d'analyses multicritère [17]

II.3.1. Méthode Weighted Sum Method (WSM)

Nous avons choisi la méthode WSM (Weight Sum Method) pour notre analyse. En effet, cette méthode permet de trouver la meilleure approche possible en attribuant un poids à chaque critère de comparaison, elle permet de prendre en compte tous les critères en fonction de leur valeur et sans critère pénalisant les autres critères.

II.3.2. Critères de comparaison multicritères

Aujourd'hui, la gouvernance informatique telle que définie par ITGI et ISACA se résume aux cinq axes suivants :

- Alignement stratégique informatique
- Livraison de valeur informatique
- Gestion des risques informatiques
- Mesure du rendement
- Gestion des ressources informatiques

Ou selon l'ISO, il existe 6 principes directeurs pour la gouvernance d'entreprise qui s'appliquent à la plupart des organisations. L'ISO spécifie que chaque principe fait référence à ce qui devrait exister, mais ne décrit pas comment, quand ou par qui ces principes doivent être mis en œuvre. Néanmoins, les décideurs doivent exiger que ces principes soient appliqués.

- Principe 1 : responsabilité
- Principe 2 : Stratégie
- Principe 3 : Acquisition
- Principe 4 : Performance
- Principe 5 : Conformité
- Principe 6 : éthique

En se basant sur l'analyse précédente, nous allons maintenant développer une analyse multicritère entre ces référentiels. Une analyse de décision multicritères, ou MCDA, est un outil précieux qui peut être appliqué à de nombreuses décisions complexes. Il peut résoudre des problèmes complexes qui incluent des aspects qualitatifs et / ou quantitatifs dans un processus décisionnel. Nous visons à aider les décideurs en facilitant le choix du meilleur cadre à utiliser selon les critères souhaités et leur importance.

Le score d'un cadre est calculé sur la base d'un certain nombre de critères. Jusqu'à présent, nous avons identifié dix critères en se basant sur l'analyse SWOT présentée plus haut qui met en évidence l'ensemble des principes de gouvernance informatique pour l'ITGI et ISO 38500, ci-dessous les critères que nous avons constitués :

- **C1** : Alignement stratégique de l'IT : une approche pour aligner la stratégie du système d'information avec la stratégie de l'entreprise.
- **C2** : Livraison de valeur informatique : La création de valeur fait référence à la finalité financière des entreprises pour leur informatique. La valeur est créée lorsque les investissements sont basés sur de l'IT dont le taux de rendement est supérieur aux coûts qui sont réalisés.
- **C3** : Gestion des risques informatiques : Risque opérationnel et métier par rapport à l'informatique, il s'agit de mettre en place des processus de gestion des risques informatiques.
- **C4** : Mesure de la performance informatique : la mise en œuvre des KPI pour atteindre les objectifs. En plus des indicateurs, c'est aussi la bonne définition des objectifs à mettre en place.
- **C5** : Acquisition IT : La gestion des prestataires et des interventions externes permet de réaliser les objectifs de l'informatique ainsi un bon pilotage de l'activité informatique.
- **C6** : Gestion des ressources informatiques : Le principal atout de la division informatique est les ressources humaines ainsi que les matériaux utilisés, ainsi une bonne gestion des ressources permet d'obtenir une bonne gouvernance du système informatique.
- **C7** : Responsabilité informatique : La définition des rôles et des responsabilités est un facteur clé de réussite des processus, c'est donc un pilier pour mener à bien les processus adoptés.
- **C8** : Conformité informatique : la réglementation informatique a beaucoup évolué au cours des dernières décennies, avec les lois de protection des utilisateurs, un bon référentiel permettant une bonne application des réglementations informatiques permettra une meilleure gouvernance.
- **C9** : Comportement humain informatique : Selon la CNIL, le respect de la liberté est un axe important pour une bonne gouvernance informatique ; plusieurs articles décrivent en détail ce pilier.

- **C10** : Contrôle informatique : mettre en place un contrôle permanent / récurrent sur l'ensemble de processus métier / informatique avec des KPI afin de contrôler l'exécution et les performances des processus.

II.3.3. Poids des critères

Les critères identifiés ont la même importance pour nous, ainsi le poids WSM accordé sera le même et égal à "1" pour l'ensemble des critères. Cependant, comme nous le verrons plus loin dans cet article le poids de chaque critère changera en fonction de chaque entreprise, par la suite nous allons exposer une vision d'une entreprise réelle qui travaillera avec cette comparaison pour l'aider à déterminer le choix du référentiel à implémenter.

II.3.4. Matrice de choix multicritères

La méthode WSM commence par remplir la matrice de choix multicritères, ses colonnes matricielles contiennent les référentiels à comparer et ses lignes contiennent les critères avec le poids attribué à chaque critère, nous convenons que ça sera «1» pour l'ensemble des critères car tous les critères ont la même importance , et dans les cellules il y a le score attribué à chaque critère basé sur l'étude comparative détaillée de chaque cadre que nous avons étudié dans l'analyse SWOT [84], [85], [86].

Concernant le score, nous utiliserons le modèle de maturité des processus, qui se compose en cinq niveaux de maturité pour pondérer le critère sur chaque cadre, chaque niveau donnera un score, par exemple « niveau 1 » donnera un score de « 1 », le « niveau 2 » donnera un score de « 2 », et ainsi de suite jusqu'au 5.

Ci-dessous la définition des 5 niveaux en l'appliquant à notre cas.

- **Niveau 1** : Il n'y a pas de méthode formelle, ni de cohérence, ni de standard, sur la base desquels le critère serait construit. L'élaboration du critère n'est pas décrit dans le référentiel.
- **Niveau 2** : Il existe un consensus dans le référentiel sur la façon dont le critère doit être géré, mais cela n'a pas été formalisé ou décrit.
- **Niveau 3** : Le processus d'élaboration du critère est formalisé, documenté et appliqué.

- **Niveau 4** : Le référentiel a institué un processus formel de collecte d'informations pour suivre et gérer l'élaboration du critère ainsi que le système qui en résulte. Les indicateurs contrôlent le bon déroulement des processus et le respect des objectifs de qualité du critère.
- **Niveau 5** : Le référentiel utilise des mesures pour optimiser en continu le processus d'élaboration du critère. Il décrit en détail un processus de correction des aspects qui seraient jugés insuffisants, en lisant les indicateurs permettant de gérer le critère.

Le tableau plus bas représente la matrice de choix multicritères résultante en fonction du score de chaque critère.

Référentiel / Critères	CMMI	COBIT	TOGAF	ISO 38500	ITIL	PMBOK	PRINCE2	SCRUM
C1	4	4	3	3	3	1	3	3
C2	3	4	4	4	4	3	3	3
C3	3	4	4	2	3	5	5	4
C4	4	5	3	4	3	4	4	4
C5	1	2	2	1	1	1	1	1
C6	1	1	2	1	2	4	2	2
C7	4	3	1	3	1	1	1	1
C8	3	2	2	1	1	1	1	1
C9	3	3	3	3	3	2	2	2
C10	4	4	2	2	3	1	5	1

Tableau 13 : Matrice de choix multicritère.

Nous convertissons le tableau en un graphique araignée à des fins visuelles ; on remarque qu'il n'y a pas de référence complète ; cependant, COBIT est le plus complet selon nos investigations, voir la figure plus bas :

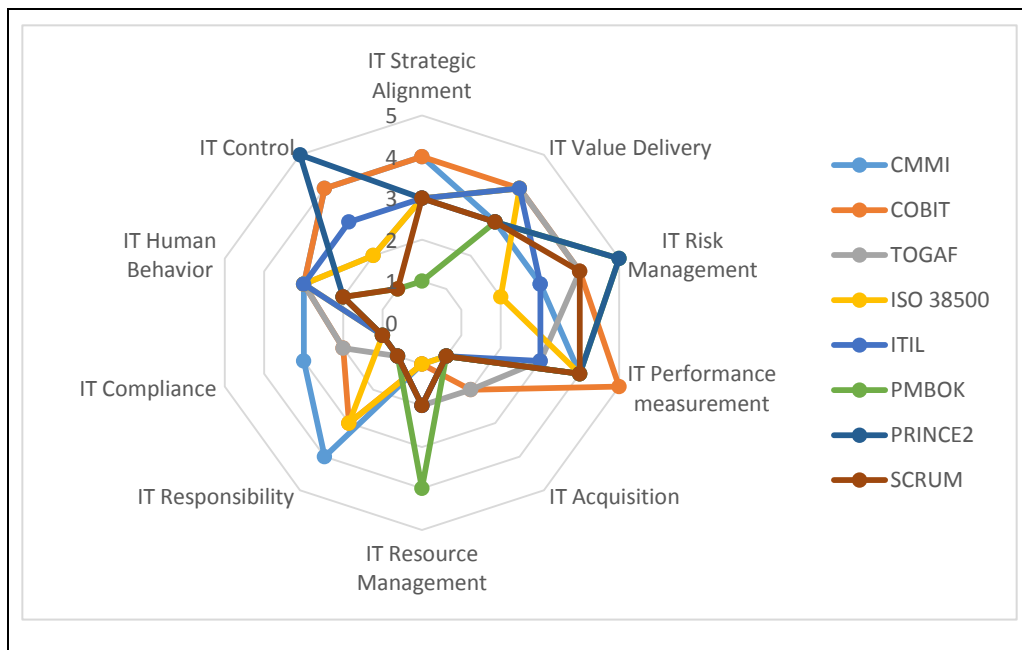


Figure 10 : Radar Multicritère des différents référentiels

II.3.5. Discussion

Toutes les composantes de la bonne gouvernance ne sont pas applicables sur tous les référentiels, en analysant les résultats, on constate que CMMI couvre 62% des composantes de la gouvernance informatique alors que COBIT couvre 65% des fondements d'une bonne gouvernance, d'autre part PMBOK présente le moindre pourcentage d'application avec seulement un taux de 43% ce qui est normal car il ne contient pas de méthodologie de gestion de projets mais c'est un ensemble de bonnes pratique en management de projets regroupé dans un seul référentiel.

Nous analyserons les composants et décortiquerons les valeurs clés de chacun des composants, en indiquant quels référentiels les appliquent le mieux, nous avons constaté que COBIT et CMMI répondent mieux à l'alignement stratégique étant donné leur engagement envers les valeurs de l'entreprise avec une définition claire des processus, ils permettent donc une vision compréhensible par la gestion de ce qui est fait par l'entreprise. En raison de l'approche processus et de la conformité, une attribution claire de la propriété et des responsabilités traite des exigences de contrôle de l'environnement informatique.

La création de valeur est présente sur presque tous les référentiels car c'est l'essence même de la mise en place de l'informatique dans l'entreprise quel que soit le référentiel choisi.

La gestion des risques est un élément important de la gouvernance informatique ; il permet d'identifier les risques de manière claire et structurée. Ayant une connaissance claire de tous les risques auxquels une entreprise est exposée, elle peut les hiérarchiser et prendre les mesures appropriées pour réduire les pertes et réduire le coût total du risque, la plupart des référentiels étudiés reposent sur des stratégies de gestion des risques selon les couche d'intervention, qu'elle soit opérationnelle ou organisationnelle.

Les processus de gouvernance informatique sont évalués, dirigés et contrôlés, ainsi les indicateurs sont un mécanisme de suivi permettant d'atteindre les objectifs commerciaux et informatiques.

En conséquence, il n'y a pas de référentiel complet de la gouvernance informatique, mais la référence la plus complète est COBIT compte tenu des positions qu'elle prend sur chacun des critères.

II.4. Développement d'un outil de choix d'un référentiel de la gouvernance IT

Suite à l'étude comparative qui a été menée, nous avons développé une application à base de notre matrice de référence qui se base sur la pondération des utilisateurs et leur restitue le bon référentiel en se basant sur la matrice de référence.

Ci-dessous l'interface de notre application :

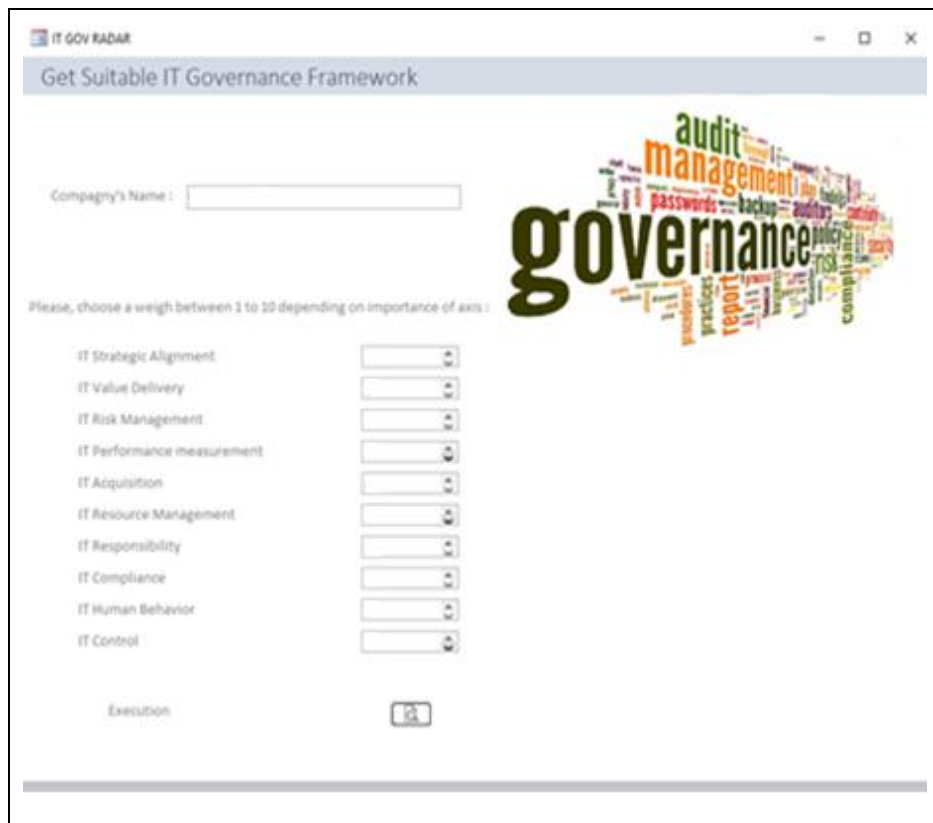


Figure 11 : Interface de l'application de choix d'un référentiel de la gouvernance IT

Cette application a été développée sous Microsoft Access vu l'aspect portabilité et facilité d'utilisation.

Par la suite nous avons donné l'application à une entreprise marocaine afin de tester leurs appréciations et afin de qualifier le résultat obtenu.

II.5. Etude de cas

Nous avons consulté la société USA HOME, une société spécialisée dans l'installation de caméras, la maintenance de logiciels de pointage et l'installation de bureaux. Compte tenu de la concurrence sur le marché marocain, le PDG souhaite optimiser ses ressources informatiques et appliquer une gouvernance informatique à son service informatique.

II.5.1. Pondération des critères suivant l'étude de cas

Sur la base d'une étude, nous avons construit un formulaire de demande afin d'extraire les intérêts de l'entreprise, ci-dessous le résultat obtenu de notre cas d'étude (entreprise USA HOME) :

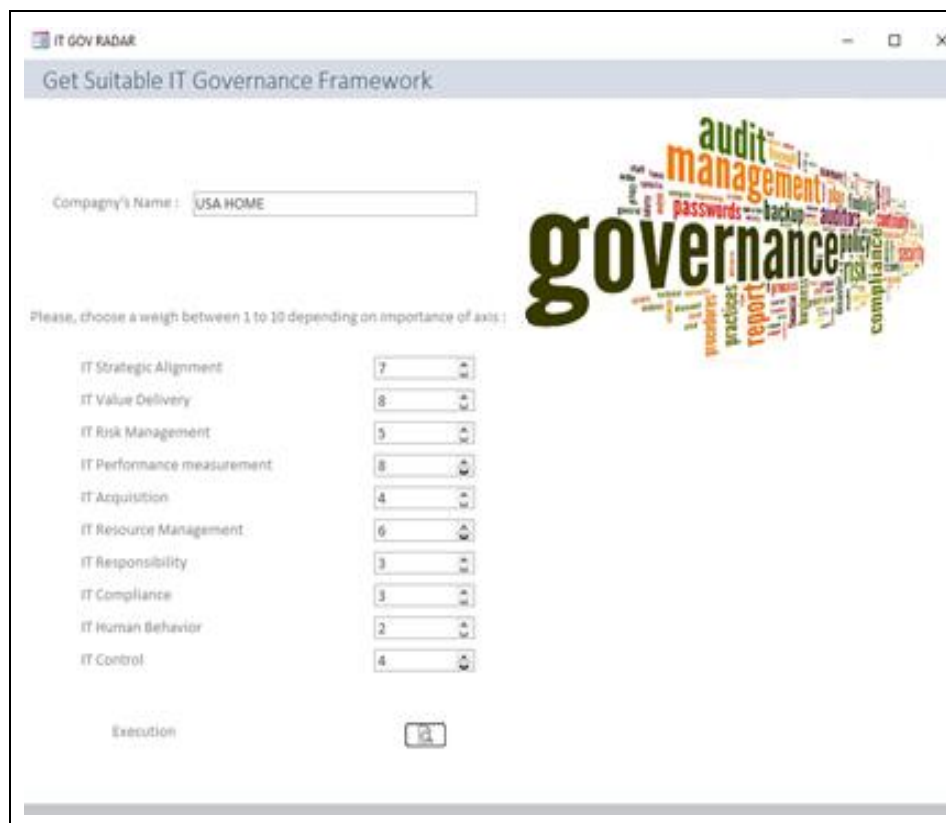


Figure 12 : Ecran de pondération remis à la société USA HOME

Ainsi la matrice de choix multicritères a changé, le score de chaque référentiel va évoluer selon la formule (1) ci-dessous :

$$P' = (5 * a * P) / S \quad (1)$$

P' Nouveau score suivant les préférences de l'entreprise

P Score initial produit

a Poids de l'entreprise

S Somme des poids de tous les critères fournis par l'entreprise

Ainsi la matrice a changé selon le poids fourni par l'entreprise, voir le tableau plus bas :

Références / Critères	CMMI	COBIT	TOGAF	ISO 38500	ITIL	PMBOK	PRINCE2	SCRUM
C1	0.93	0.88	0.81	0.88	0.88	0.30	0.78	0.95
C2	0.80	1.00	1.23	1.33	1.33	1.04	0.89	1.09
C3	0.50	0.63	0.77	0.42	0.63	1.09	0.93	0.91

Selon le résultat obtenu, PMBOK sera une bonne référence en fonction des exigences de l'entreprise USA HOME. Il permettra à l'entreprise de valoriser le potentiel de créativité, assurant la réalisation et la coordination des acteurs du projet et des ressources disponibles selon le plan défini, c'est aussi un moyen efficace pour une entreprise très concernée par les projets.

Ce choix est aussi justifié par la taille de l'entreprise qui veut gagner en maturité et s'élargir dans son portefeuille d'activité avec le minimum de risque possible.

II.6. Conclusion

Le choix d'un cadre de gouvernance informatique est une tâche très importante après la mise en œuvre de la stratégie, un bon choix conduira à de meilleurs résultats [87]. Ce chapitre nous a permis de déceler les points forts et les points d'efforts sur les systèmes de gouvernances IT puis de faire une contribution aux professionnels pour les aider à choisir entre différents cadres existants, et ce en fonction de leurs besoins qui comptent le plus pour eux. Dans le chapitre suivant, nous allons présenter une proposition d'approche de génération d'un méta-modèle en appliquant les concepts IDM puis nous allons appliquer cette approche sur la gestion de projets.

Chapitre III : Proposition d'une approche générique de construction de méta- modèles & application sur la gestion de projets

III.1. Introduction

Dans ce chapitre, nous allons proposer une approche pour la génération d'un méta-modèle unique à partir de plusieurs modèles en se basant sur plusieurs concepts qui tournent autour de l'ingénierie dirigée par les modèles (IDM). Le but est d'agréger plusieurs modèles pour construire un seul méta-modèle global. Nous allons nous appliquer notre approche sur la gestion de projet comme composante de la gouvernance IT dans la suite de ce rapport.

III.2. Proposition d'une approche générique de construction de méta-modèles

En se basant sur l'état de l'art effectué sur l'ingénierie dirigé par les modèles. Premièrement, nous proposons de produire un cadre de référence d'un sujet donné, il contient des éléments importants qui devraient être inclus dans le méta-modèle. Ensuite, une deuxième étape consiste à rassembler les modèles existants du sujet, notez que plus nous rassemblons de modèles, mieux c'est pour la qualité de notre méta-modèle. Ensuite, nous analysons les concepts en utilisant des techniques de correspondance ou de matching. Enfin, nous optimisons les concepts remontés dans un méta-modèle contenant une agrégation de tous les modèles. Ainsi, nous proposons une approche est basée sur quatre étapes (cf. figure plus bas) :

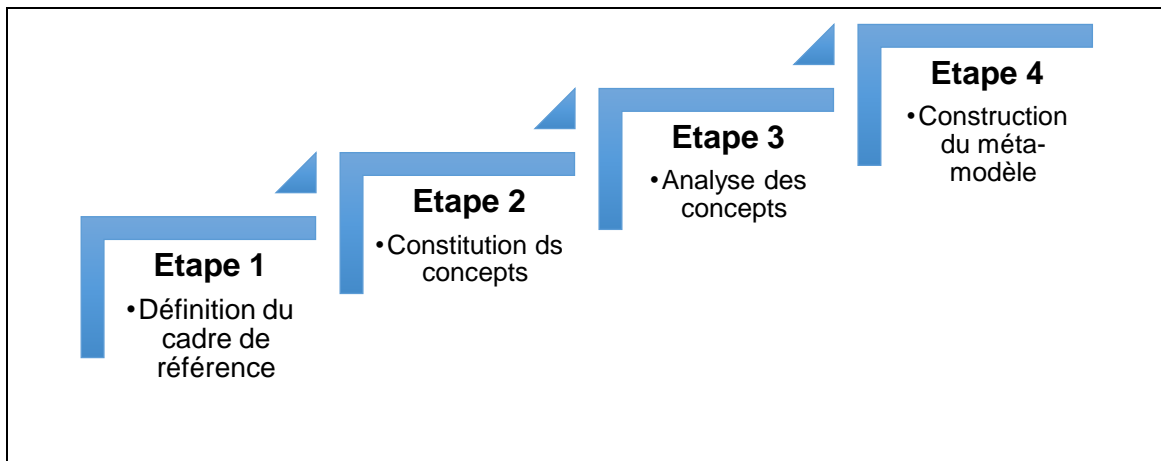


Figure 14 : Approche de construction d'un méta-modèle

III.2.1. Définition du cadre de référence

La définition d'un cadre référentiel (cf. figure plus bas) est une étape essentielle pour délimiter le champ d'une recherche, il s'agit d'un ensemble de règles et de concepts interconnectés de façon à effectuer des recherches et trouver une solution pour une problématique donnée.

Afin d'établir un méta-modèle d'un système cible, le cadre de référence nous permet de délimiter notre système selon des plusieurs facettes, le but étant un bon cadrage des exigences du système, ainsi il nous permet la génération d'un ensemble de concepts et de relation qui doit figurer dans le méta-modèle, il s'agit des abstractions importantes pour la constitution d'une première couche d'un méta-modèle.

Un cadre de référence peut être modélisé comme suit :

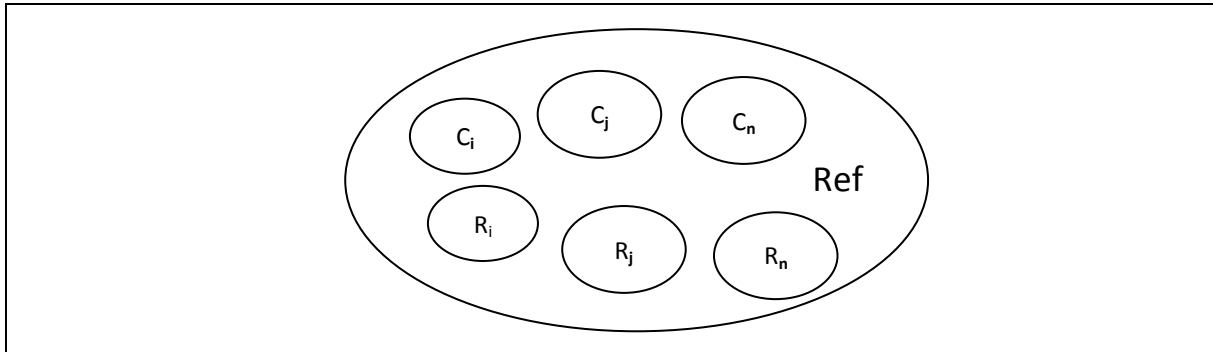


Figure 15 : Modélisation du cadre de référence

Ou selon la formule mathématique ci-dessous :

$$\text{Ref} = \{C_j, R_j, \forall j \in \mathbb{N}, C_j \in \text{Ref}, R_j \in \text{Ref}\} / C_j, R_j \in \text{PIMG} \quad (1)$$

Où **PIMG** : le méta-modèle du système.

R : une relation inter-concept.

C : un concept appartenant au cadre de référence.

III.2.2. Constitution des concepts

Pour construire un méta-modèle d'un système nous disposons d'un cadre de référence qui délimite le champ de notre vision, il se constitue par les éléments que nous jugeons important dans le système et qui doivent idéalement figurer dans le méta-modèle, par la suite nous allons considérer qu'un cadre de référence est une modélisation idéale du système $\text{PIM}_{\text{idéal}}$ (Platform Independent Model) qui figure complètement dans le méta-modèle, seulement il existe d'autres concepts qui sont à prendre en compte, ne figurant pas dans notre champs de vision, voir la figure plus bas :

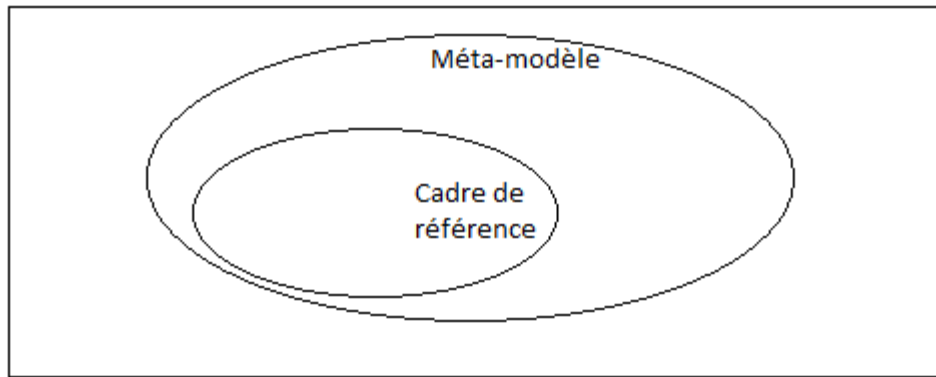


Figure 16 : Inclusion du cadre de référence dans le PIM

Si nous appelons le méta-modèle PIMG; nous pouvons conclure la formule ci-dessous :

$$PIM_{\text{idéal}} = \{C_j ; \forall j \in \mathbb{N} \quad C_j \in \text{Cadre Référence} \} \subset \text{PIMG}$$

Aussi, nous disposons de plusieurs modèles du même système que nous voulons construire, chaque modèle appelé aussi PIM (Platform Independent Model) à ses propres spécificités que chacun juge important lors de son implémentation. Le modèle idéal aura plusieurs intersections avec tous les PIMs que le système représente, par contre nous pouvons trouver qu'il y'a plusieurs intersections entre les différents PIMs indépendamment du modèle idéal que nous avons construit. Ainsi, pour obtenir un méta-modèle fiable permettant de générer tous les PIMs qu'il représente, il faut inclure dans le méta-modèle tous les intersections entre les différentes modèles PIMs que le système peut représenter.

III.2.3. Analyse des concepts remontés

Le fait de factoriser les concepts trouvés dans chaque PIM donnera naissance à des concepts redondants qu'il faudra analyser et supprimer par la suite. Il faudra définir des règles qu'il faut suivre pour filtrer les concepts remontés dans le méta-modèle.

En tant que technique d'analyse des concepts, nous pouvons nous appuyer sur l'une des approches suivantes [3] :

- **Approche générative** : basée sur le nom des concepts réassemblés.
- **Approche interprétative** : Interprétation fonctionnelle des concepts.
- **Approche hybride** : une combinaison d'approche générative et interprétative.

- **Analyse factorielle des concepts** (Factorial Analysis of Concepts AFC) : consiste à factoriser des concepts selon les axes définis dans le référentiel.
- **Analyse relationnelle des concepts** (Relational Analysis of Concepts ARC) : consiste à factoriser les concepts selon le niveau hiérarchique de leurs relations.

III.2.4. Construction du méta-modèle

Le méta-modèle est une intersection de tous les modèles (cf. figure plus bas). Où Idéal PIM représente le cadre de référence et PIM1, PIM2 et PIM3 sont des modèles du même système.

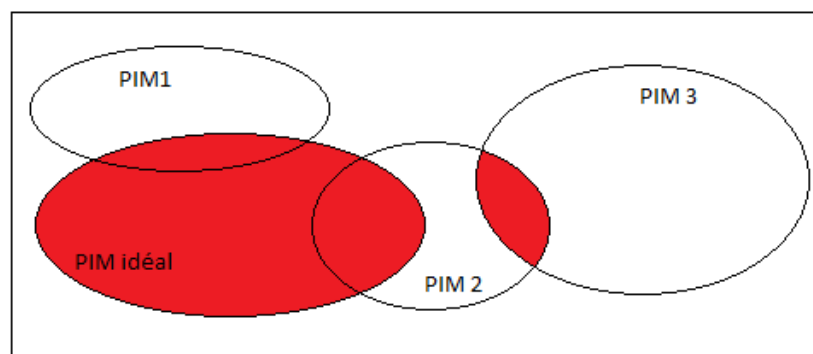


Figure 17 : Construction d'un méta-modèle

Nous pouvons conclure que le méta-modèle est un modèle constitué de l'intersection maximale de tous les modèles. Voir la figure plus haut où PIM idéal représente le cadre de référence et PIM1, PIM2 et PIM3 sont des modèles du même système.

Par conséquent, le méta-modèle sera l'union de tous les concepts analysés, voir la formule plus bas :

$$PIMG = \bigcup_i^j PIM_i \cap PIM_j \quad / \quad i \neq j \quad (2)$$

III.3. Proposition d'un méta-modèle global pour la gestion de projets

III.3.1. Gestion de projets

Dans l'organisation, tous les processus peuvent être sujets à des perturbations. Nous parlons dans ce rapport d'un aléa ou un besoin de changement pour caractériser tous types de

perturbations. Un aléa est défini dans le dictionnaire Larousse comme "un tour imprévisible et le plus défavorable pris par les événements et lié à une activité, action". Il s'agit d'événements qui affectent toutes les perspectives de l'approche processus présentées dans le chapitre précédent. Chacune de ces perspectives permet de décrire un aspect important des modèles du processus et peut être un objet du changement tout au long de son cycle de vie [89]. Dans le travail de [90], selon une étude qu'il a réalisée sur les travaux de [91] et [92], la gestion de l'incertitude et la réponse aux changements est devenue une préoccupation primordiale des organisations. Si la nécessité de gérer des changements n'est pas nouvelle pour l'entreprise, ce qui l'est davantage en revanche, c'est l'augmentation quasi-constante de la fréquence avec laquelle ces changements doivent être menés. Ceci explique l'intérêt croissant pour les recherches qui portent sur l'agilité. En effet, dans le but de répondre aux changements, l'agilité avec ses deux dimensions (i.e organisationnelle et technique) permet d'agir sur toutes les perspectives du processus : (i) l'agilité technique agit sur la perspective informationnelle avec la résolution des différences syntaxiques et sémantiques et garantir l'interopérabilité d'entreprise, (ii) l'agilité organisationnelle permet d'agir au niveau de la perspective organisationnelle avec une mobilisation et restructuration dynamique de ressources et (iii) les deux dimensions (i.e organisationnelle et technique) agissent sur la perspective fonctionnelle, opérationnelle et comportementale avec une reconfiguration dynamique du processus.

Avant la publication du manifeste agile en 2001, quelques méthodes agiles étaient déjà appliquées en conception logicielle. Nous allons présenter les plus fréquemment citées [93] [94]. Ces méthodes sont dites différenciatrices et présentent donc des caractéristiques différentes, mais répondent toutes à l'objectif principal de l'agilité : la satisfaction client. Les pratiques constituant ces méthodes peuvent en théorie être utilisées partout, individuellement ou couplées à d'autres [95] et proposent donc un caractère malléable.

La méthode Rapid Application Development ou RAD est la première méthode agile de gestion de projets en rupture totale avec le cycle auparavant classique de développement en cascade. Ce nouveau cycle de développement est qualifié d'itératif, incrémental et adaptatif. Cette méthode a été publiée par James Martin [96]. La méthode RAD préconise des phases d'évaluations individuelles puis collectives du code produit avant de l'intégrer rapidement dans le projet. Nous trouvons également des phases de programmation en binôme pour les parties les plus stratégiques, permettant à la fois d'augmenter la capacité à déceler des erreurs, mais également d'augmenter la créativité.

Par ailleurs, la méthode RAD introduit le rôle de facilitateur. Cette personne, externe à l'équipe de développement, permet d'avoir un recul sur le projet. En fonction de l'évolution du projet, que ce soit sur le plan technique, économique ou stratégique, ce représentant du client peut prendre des décisions afin de réorienter le projet de manière objective. Le facilitateur a également pour mission de proposer des solutions aux problèmes que lui rapporterait l'équipe de conception.

La méthode RAD propose dans sa version 2 une plus grande souplesse quant à la taille des groupes de conception et ce afin d'optimiser les ressources humaines en fonction de la nature du projet. Elle apporte également une plus grande considération à la communication entre les acteurs de la conception en mettant en place des groupes d'animation et de rapport. Ces groupes sont chargés de l'évaluation et de la validation. Enfin, la version 2 offre la possibilité de proposer des livrables de fonctionnalités réduites afin d'écourter encore plus la durée des cycles de développement.

Extreme programming ou XP, est une méthode agile de gestion de projets orientée production logicielle, inventée par Kent Beck, Ward Cunningham et Ron Jeffries, utilisée dès 1996, et documentée en 1999 [97]. Elle repose sur cinq valeurs :

- Communication entre tous les acteurs
- Simplicité
- Nombreuses rétrospectives
- Courage
- Respect

Les pratiques de cette méthode sont l'application de cycles très rapides, avec peu de fonctionnalités à faire. Dès que les tâches sont terminées, elles sont implémentées dans le logiciel, créant ainsi une intégration continue. Nous trouvons aussi une très forte implication du client ou de son représentant qui participe activement aux prises de décisions. L'équipe de conception souhaite parfois avoir le client ou son représentant directement sur site [98]. La méthode XP préconise également l'utilisation du Planning Poker, un jeu de cartes d'assistance à la communication et à la planification [99] [100]. Ce jeu de cartes permet d'estimer la complexité et la durée des tâches de conception à réaliser tout en mettant en confrontation les joueurs face aux estimations des autres. Concernant l'équipe de conception, elle doit être de

petite taille, c'est à dire composée de moins de 20 membres. Ils ne doivent jamais faire d'heures supplémentaires pour être tout le temps efficace et peuvent pratiquer la programmation en binôme. Tout comme dans la méthode RAD, cette pratique consiste à positionner deux concepteurs l'un à côté de l'autre et à les faire écrire le code en collaboration. Dans la méthode XP, il est conseillé de pratiquer la programmation en binôme le plus souvent possible.

La méthode Scrum est une méthode agile de gestion de projets associant un processus de travail composé de courtes itérations appelées « sprints » au bout desquelles des livrables sont proposés ainsi que des pratiques appliquées tout au long de ces sprints. Cette méthode a été initialement théorisée par Takeuchi et Nonaka en [101] sous le nom de « rugby approach » par analogie avec une équipe de rugby qui avancerait unie pour faire avancer la balle. La méthode rugby approach était initialement destinée au secteur industriel, et proposait une production itérative et multidisciplinaire [102]. C'est en 2001 que Schwaber et Beedle décrivent la méthode appelée Scrum encore utilisée aujourd'hui [103].

La méthode consiste à écrire des scénarios d'usage, appelés user stories, dans lesquels les concepteurs se projettent comme des utilisateurs potentiels. La méthode Scrum (mêlée en anglais) est une méthode agile de gestion de projets associant un processus de travail composé de courtes itérations appelées « sprints » au bout desquelles des livrables sont proposés ainsi que des pratiques appliquées tout au long de ces sprints. Cette méthode a été initialement théorisée par Takeuchi [104] sous le nom de « rugby approach » par analogie avec une équipe de rugby qui avancerait unie pour faire avancer la balle. La méthode consiste à écrire des scénarios d'usage, appelés user stories, dans lesquels les concepteurs se projettent comme des utilisateurs potentiels. Durant ce sprint, l'équipe travaillera à la réalisation de tâches pour répondre aux usages des user stories. Il s'agit donc de transformer les besoins du client en intentions conceptuelles puis en tâches de conception. Tandis qu'un sprint dure en général d'une à quatre semaines, chaque jour l'équipe de conception doit pratiquer une réunion, appelée daily scrum. Durant cette réunion, les concepteurs échangent rapidement sur leur état d'avancement et sur les problèmes éventuels. Lorsque toutes les user stories sont terminées ou que le sprint touche à sa fin, les concepteurs organisent une réunion avec les clients, ou leur représentant, appelée sprint review, dans laquelle ils présentent des livrables montrant les user stories complétées. Cela peut être une démonstration, un visuel, un tableau ou n'importe quel élément visualisable. Le product backlog est mis à jour, tandis qu'un énième sprint est prêt à

recommencer. Les concepteurs réalisent une autre réunion, appelée la sprint retrospective dans laquelle ils énumèrent les points positifs et les points négatifs du sprint tout juste réalisé, dans le but d'améliorer le sprint à venir. Chaque sprint, donc chaque itération permet d'incrémenter le projet de fonctionnalités, et les nombreuses réunions permettent de proposer une forte adaptabilité, tout en améliorant la cohésion de groupe, la conscience mutuelle et le partage de connaissances [105] [106] [107].

III.3.2. Application de l'approche proposé sur la gestion de projet

Dans la suite nous allons appliquer notre approche sur la gestion de projet :

III.3.2.1. Définition du cadre de référence

Comme il existe de nombreuses facettes pour voir un problème, il existe plusieurs types de cadre de référence. Il vise à clarifier le sujet de recherche, à présenter une vision complète et à donner une forme structurée aux réponses de tout sujet.

Dans ce qui suit, nous utiliserons le « cadre des quatre mondes » introduit par Rolland en 1998 [83].

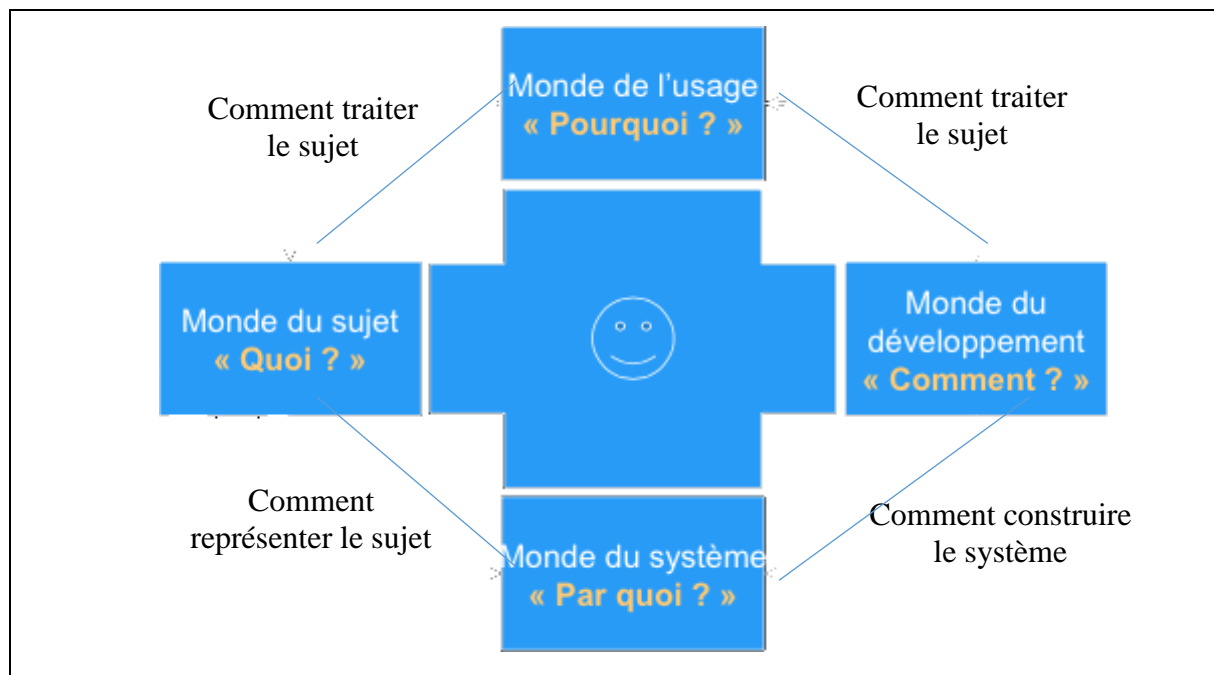


Figure 18 : Modélisation du cadre de référence

Un cadre référentiel a pour but de clarifier le sujet de notre recherche, présenter une vision complète d'un problème et de donner une forme structurée aux réponses des problématiques posées.

Le cadre du système d'origine (cf. figure plus haut) contient quatre axes, à savoir :

- **Le monde du sujet** : contient la connaissance du domaine sur lequel le SI proposé doit fournir des informations. Il contient des objets du monde réel, qui deviennent le sujet de la modélisation du système. Contient la réponse à Quoi que la gestion de projets (ensemble de processus, rôles et indicateurs pour diriger le projet).
- **Le monde du système** : comprend des spécifications à différents niveaux de détail de ce que fait le système. Il contient les entités modélisées, les événements, les processus, etc. du monde sujet ainsi que la mise en correspondance avec les spécifications de conception et les implémentations.
- **Le monde de l'utilisation** : décrit l'environnement organisationnel du système d'information, c'est-à-dire l'activité des agents et la manière dont le système réalise le travail, y compris les parties prenantes qui sont propriétaires et utilisateurs du système. Le monde d'utilisation traite des aspects intentionnels du SI à construire tandis que le monde sujet se réfère au domaine qu'il représentera.
- **Le monde du développement** : se concentre sur les entités et les activités qui découlent du processus d'ingénierie lui-même. Il contient les processus qui créent le système d'information, c'est-à-dire les processus qui impliquent l'analyse et la compréhension des connaissances contenues dans les autres mondes et la représentation de ces connaissances.

En appliquant le même cadre à la gestion de projets appliquée au contexte de gestion de projets, nous obtenons ce qui suit (voir la Figure 5) :

- **Le monde du sujet** contient la réponse à Qu'est-ce que la gestion de projets (ensemble de processus, rôles et indicateurs pour diriger le projet).
- **Le monde du système** comprend les spécifications des projets Avec Quoi nous gérons les projets (à l'aide de métriques et de modèles).

- **Le monde de l'utilisation** décrit l'environnement organisationnel, il répond à Pourquoi nous utilisons la gestion de projets (pour créer de la valeur, gérer le temps et améliorer la qualité).
- **Le monde du développement** contient les activités, c'est plus comme Comment la gestion de projets obtiendra des résultats (en utilisant des activités et des phases pour produire des produits de travail).

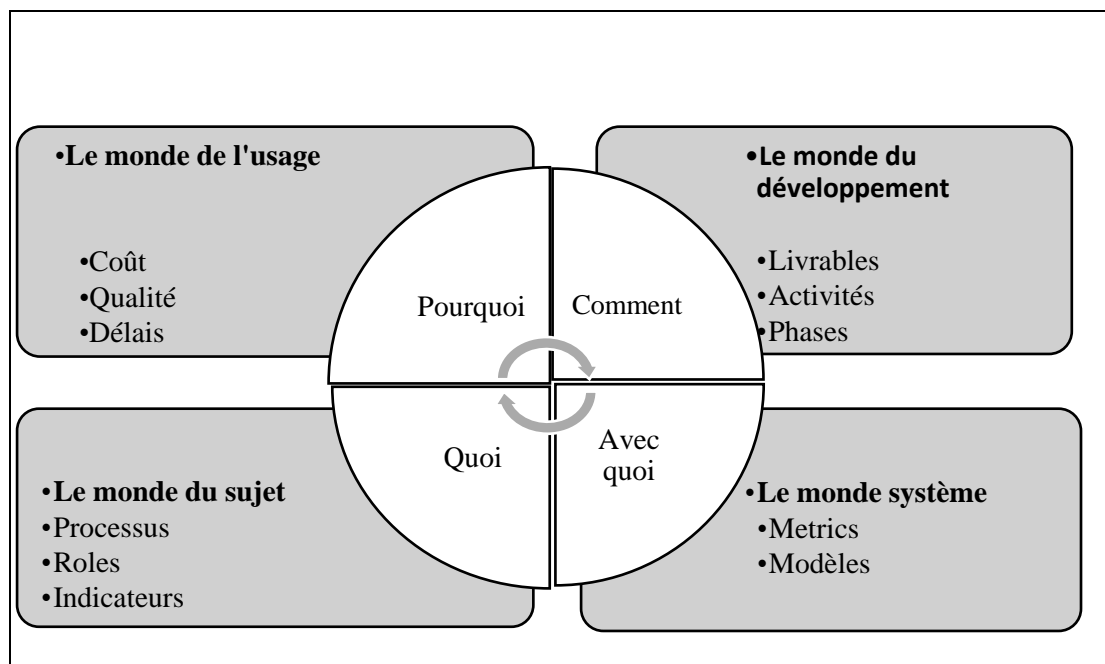


Figure 19 : Cadre de référence pour la gestion de projets

Ce cadre de référence est un outil très important qui nous donne une compréhension claire de gestion de projets ; c'est aussi la première étape pour modéliser le système.

III.3.2.2. Constitution des concepts

En se basant sur le cadre de référence nous pouvons extraire plusieurs concepts clés de gestion de projets (voir le tableau ci-dessous) :

Concepts	Définition
Coût	La gestion des coûts est le processus d'estimation, d'allocation et de contrôle des coûts d'un projet. Cela permet à une entreprise de prévoir les dépenses à venir afin de réduire les chances qu'elle dépasse son budget.
Qualité	Répondre aux exigences du client. En d'autres termes, en gardant la définition liée au client ou à la partie prenante, la gestion de la qualité peut avoir une orientation plus étroite, ce qui signifie qu'elle a plus de chances d'atteindre ses objectifs.
Délais	Temps alloué au projet, en excluant la partie de maintenance.
Livrables	Un produit ou bien un service donné au client
Activités	Une étape d'un plan de gestion de projets
Phases	Une partie du plan de gestion de projets
Processus	Une série d'actions ou de mesures prises pour atteindre une fin particulière
Rôles	Une fonction assignée à une personne afin d'accomplir une ou plusieurs tâche
Indicateurs	Il s'agit d'une information qui vous aide à évaluer une situation et à prendre une décision adaptée. Un indicateur doit être réaliste, mesurable et défini dans le temps.
Métriques	Tout type de mesure utilisée pour mesurer un certain composant quantifiable de performance.
Modèles	Démarches de matérialisation des processus

Tableau 15 : Liste des concepts remonté dans le cadre de référence.

Ci-dessous la définition d'autres concepts remontés en se basant sur les méthodes de gestion de projet Agile et Prédictive :

Concepts (FR)	Concepts (EN)	Description
Processus	Process	Éléments de méthodologie transformant les entrées en sorties. La relation entre un ou plusieurs processus éléments génère le processus.
Cycle	Cycle	Éléments de méthodologie transformant les entrées en sorties. La relation entre un ou plusieurs processus éléments génère le processus.
Activité	Activity	Un élément de processus abstrait de base qui modélise une

		opération fonctionnellement cohérente qui peut être effectuée par un ou plusieurs producteurs
Tâche	Task	La plus petite unité de travail gérable.
Technique	Technique	Des moyens spécifiques pour atteindre un objectif.
Mesure	Measurement	Des moyens spécifiques pour effectuer des mesures sur les processus
Risque	Risk	Un élément essentiel en ce qui concerne les méthodes Prédictives
Producteur	Producer	Quelqu'un ou quelque chose qui exécute une unité de travail soit directement ou indirectement (c.-à-d. crée, évalue, itère ou maintient).
Etape	Stage	Représente des intervalles de temps gérés ou des points temps, au sein d'un projet, c'est-à-dire l'aspect temporel du processus.
Jalon	Milestone	Il s'agit des jalons des projets
Produit	Product	Tout artefact développé et produit pendant le projet. Cet élément équivaut au "Travail Élément produit".
Activité	Activity	Un livrable est un élément de produit qui fournit un description et définition de l'emballage d'autres éléments de produits, et peuvent être livrés à un ou partie externe.
Documents	Documents	Une ressource identifiable fournie par l'équipe de développement qui prend en charge le processus global de développement logiciel.
Artefact	Artefact	Une ressource identifiable fournie par l'équipe de développement qui prend en charge le processus global de développement logiciel.
Résultat	Outcome	Le résultat est un élément de produit qui fournit une description et une définition des éléments de produit non tangibles. Une différence clé entre les résultats et artefacts est que les résultats ne sont pas candidats pour récolter comme actifs réutilisables.
Ressource	Ressource	Éléments méthodologiques utilisés dans la réalisation du projet.
Langage	Language	Vocabulaire (c'est-à-dire un ensemble de termes) combiné avec un ensemble de règles grammaticales utilisées pour produire un ou plusieurs éléments de produits.
Ligne directrice	Guideline	Règles et directives sur l'utilisation appropriée de un élément méthodologique donné.

Tableau 16 : List des concepts remontés.

III.3.2.3. Analyse des concepts remontés

Nous nous sommes basé sur une analyse fonctionnelle entre les différents concepts remontés afin de faire un matching entre les deux modèles prédictive et agile. Puis, nous nous sommes basé sur une analyse interprétative basée sur les noms des concepts rassemblés ainsi que leurs définitions pour identifier les concepts qui se ressemblent.

Nous avons pu conclure par la suite un matching entre les différents concepts, ci-dessous le résultat du matching manuel que nous avons effectué :

Meta-Modèle Prédictive	Méta-Modèle Agile
Process	Cycle
Activity	Activity
Task	Task
Technique	Technique
Measurement	Measurement
Risk	Managing Product Delivery
Producer	Producer
Step	Stage
Milestone	Release
Product Element	Product Element
Activity	Activity
Documents	Artefact
Outcome	Outcome
Resource	Resource
Language	Language
Guideline	Guideline

Tableau 17 : Résultat du matching interprétative des concepts.

III.3.2.4. Construction du méta-modèle

En se basant sur les principes de l'ingénierie de modèle, nous avons analyser les concepts remonté dans le tableau précédent, ainsi la gestion de projets se base sur des phases que ça soit des sprints ou bien des séquences qui sont liés entre eux, ils se composent de

plusieurs activités que ça soit des activités d'initialisation de phase, de revue ou bien de clôture de phase.

Ainsi nous proposons le méta-modèle générique plus bas :

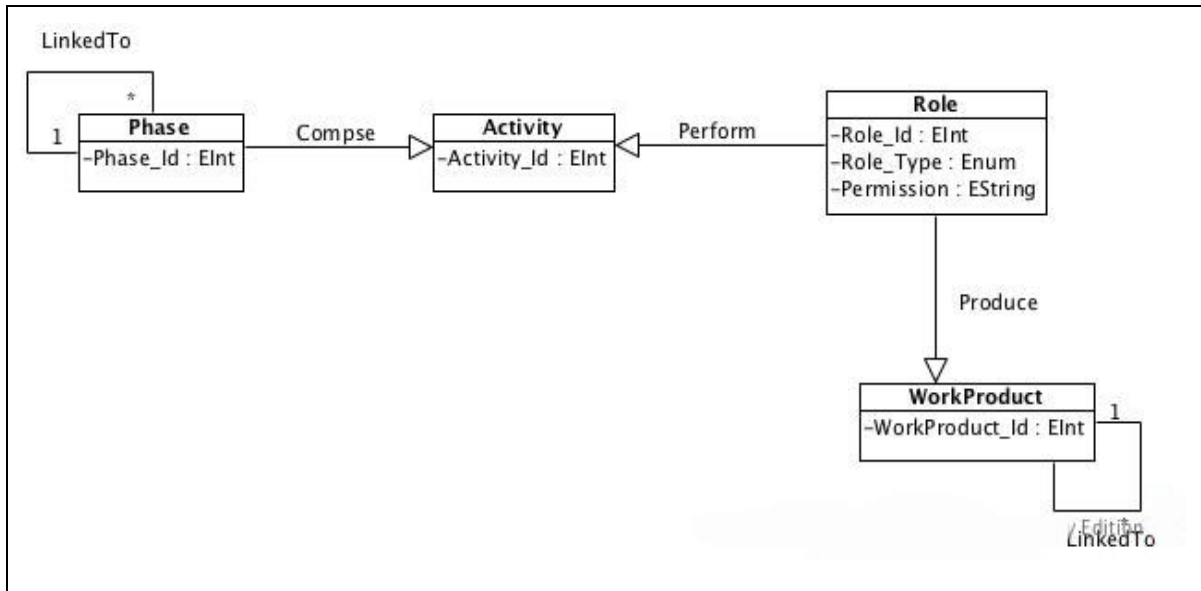


Figure 20 : Méta-modèle générique de Gestion de projets [88]

En incluant les concepts remontés dans le paragraphe d'analyse des concepts, nous pouvons conclure le méta-modèle détaillé plus bas :

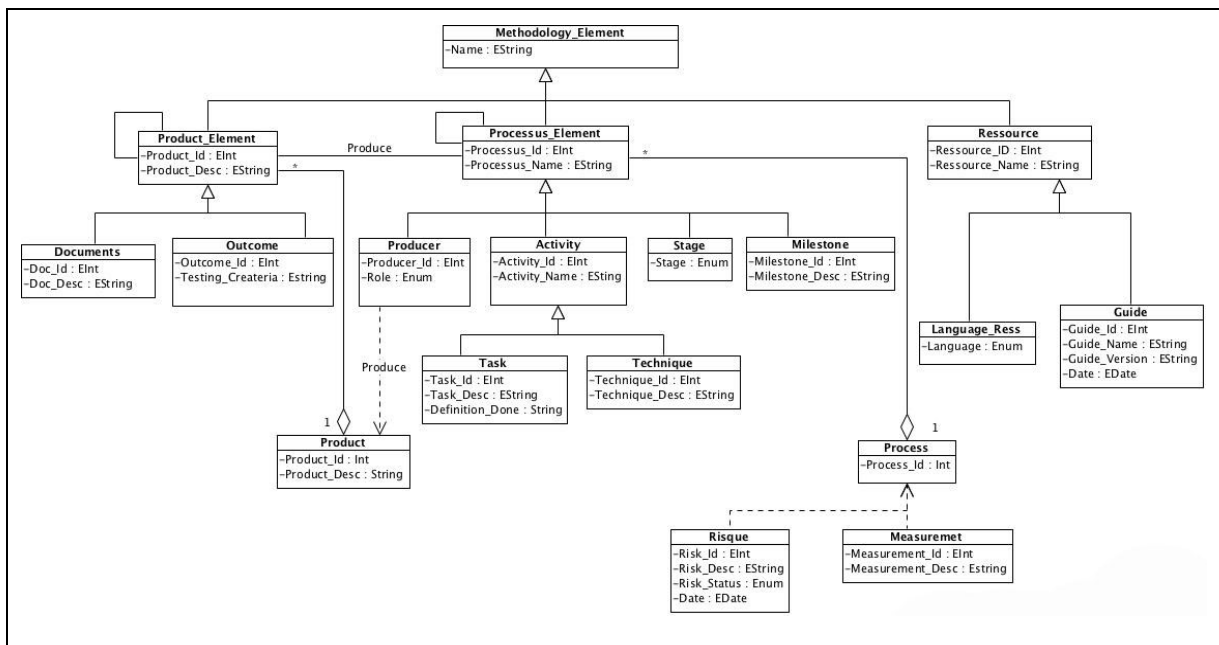


Figure 21 : Méta-modèle détaillé de Gestion de projets

III.4. Conclusion

Dans ce chapitre, nous avons présenté en détail chaque étape de notre approche, en mettant en œuvre pour chaque étape les outils à utiliser pour la constitution du méta-modèle global. Ensuite, nous avons appliqué cette approche à la gestion de projets, en utilisant des mécanismes pour rendre les méta-modèles fiables. Ce travail nous a finalement permis d'établir un méta-modèle de gestion de projets générique et détaillé.

L'arbre généalogique des méthodes de gestion de projets se limite à deux branches. La première, dans l'ordre d'apparition chronologique, est fondée sur un cycle de levée du risque par la validation d'une succession de nombreuses étapes. L'objectif étant de tenter de réduire le risque du changement en cours de développement. Les méthodes issues de cette branche sont représentées par la métaphore d'une « cascade » ou bien les méthodes prédictives. La seconde branche, plus récente, se base sur la mesure objective du changement afin de l'accepter. Elle se structure selon un cycle dit « itératif, incrémental et adaptatif ». Les méthodes issues de la branche « cascade » se réfèrent du « prédictif » et les autres de l'« adaptatif » ou bien l'agile.

Par la suite, nous allons appliquer notre approche sur deux méthodes de la gestion de projets. Nous allons travailler avec deux méthodes concurrentes les plus utilisés dans la dernière décennie Agile et Prédictive, abstraction faite des méthodologies choisies que ça soit SCRUM, XP, Water Fall ou bien PRINCE2.

Chapitre IV : Proposition d'un méta-modèle pour les méthodes prédictives de la gestion de projets

IV.1. Introduction

En suivant l'ordre chronologique de l'apparition des méthodes de gestion de projets, nous allons expliciter dans ce chapitre les différents concepts des méthodes Prédictives, les rôles, les processus, les thèmes, les principes et les livrables de la méthodologie. Nous allons prendre comme exemple la méthode PRINCE 2 puis nous allons proposer un méta-modèle des méthodes Prédictives.

Le but est d'analyser par la suite le méta-modèle proposé et le comparer avec un méta-modèle des méthodologie Agile en utilisant l'approche proposée dans le chapitre précédent et construire une transformation entre les deux modèles.

IV.2. Principes des méthodes Prédicatives

La méthodologie prédictive se concentre sur la planification et l'analyse approfondie de l'avenir projeté pour les risques anticipés. Aussi connue sous le nom de méthode Prédicative, cette méthode repose sur une analyse de phase précoce et une répartition détaillée des fonctionnalités et des tâches pour l'ensemble du processus de développement de projet. Dans ce modèle traditionnel, nous définissons des délais standard pour effectuer des tâches et des dépendances entre les tâches. Cela détermine également le chemin critique et une date d'achèvement prévue. Il devient assez difficile de changer la direction du projet si quelque chose ne va pas. Les équipes prédictives n'apprécient pas le changement et des tampons sont souvent ajoutés aux calendriers pour s'adapter à contrecœur aux changements mineurs.

Ci-dessous, un bref aperçu de la méthodologie de planification des projets gérés en prédictive :

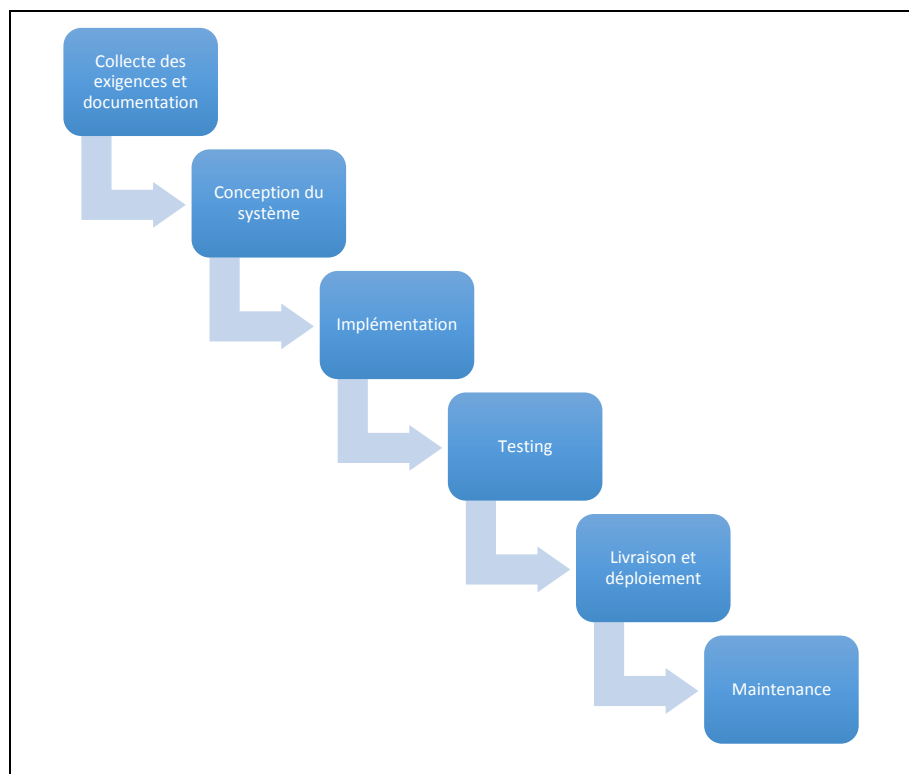
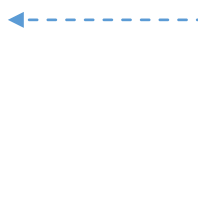


Figure 22 : Aperçu d'un exemple de cadre de projet prédictif

IV.2.1. Collecte et documentation des exigences



À cette étape, vous devez rassembler des informations complètes sur les besoins de ce projet. Vous pouvez collecter ces informations de différentes manières, des entretiens aux questionnaires en passant par le brainstorming interactif. À la fin de cette phase, les exigences du projet doivent être claires et vous devez disposer d'un document d'exigences qui a été distribué à votre équipe.

En utilisant les exigences établies, votre équipe conçoit le système. Aucun codage n'a lieu pendant cette phase, mais l'équipe établit des spécifications telles que le langage de programmation ou les exigences matérielles.

IV.2.2.Implémentation du projet

Pour les projets informatiques, le codage a lieu dans cette phase. Les programmeurs prennent les informations de l'étape précédente et créent un produit fonctionnel. Ils implémentent généralement le code en petits morceaux, qui sont intégrés à la fin de cette phase ou au début de la suivante.

IV.2.3.Test du projet

Une fois que tout le codage est terminé, les tests du produit peuvent commencer. Les testeurs trouvent et signalent méthodiquement tout problème. Si des problèmes graves surviennent, votre projet devra peut-être revenir à la première phase pour une réévaluation.

IV.2.4.Livraison et déploiement du projet

Dans cette phase, le produit est terminé et votre équipe soumet les livrables à déployer ou à publier.

IV.2.5.Maintenance

Le produit a été livré au client et est en cours d'utilisation. Lorsque des problèmes surviennent, votre équipe peut avoir besoin de créer des correctifs et des mises à jour peuvent les résoudre. Encore une fois, de gros problèmes peuvent nécessiter un retour à la première phase.

IV.3. Proposition d'un cadre de référence

En se basant sur l'explication des méthodes Prédictives, nous avons mis les concepts sur la cadre de référence de Rolland en 1998 [83], voir les quatre mondes du modèle ci-dessous :

- **Le monde du sujet** contient la réponse à **Qu'est-ce** que la gestion de projets prédictive : Un ensemble de processus à respecter pour mener à terme un projet, il s'agit d'un ensemble de processus, rôles et indicateurs pour diriger le projet.
- **Le monde du système** comprend les spécifications des projets **Avec Quoi** nous gérons les projets : un ensemble de techniques et des livrables pour le suivi, en utilisant des métriques et produire plusieurs types de documents.
- **Le monde de l'usage** décrit l'environnement organisationnel, il répond à **Pourquoi** nous utilisons la gestion de projets : Il s'agit des principes des méthodes Prédictives ; certains se basent sur les contraintes, d'autres sur les délais. Il s'agit d'un ensemble de principes pour créer de la valeur, gérer le temps et améliorer la qualité des projets en se basant sur les principes des méthodes Prédictives.
- **Le monde du développement** contient les activités, c'est plus comme **Comment** la gestion de projets obtiendra des résultats : Processus à respecter en utilisant des activités et des phases pour produire des livrables.

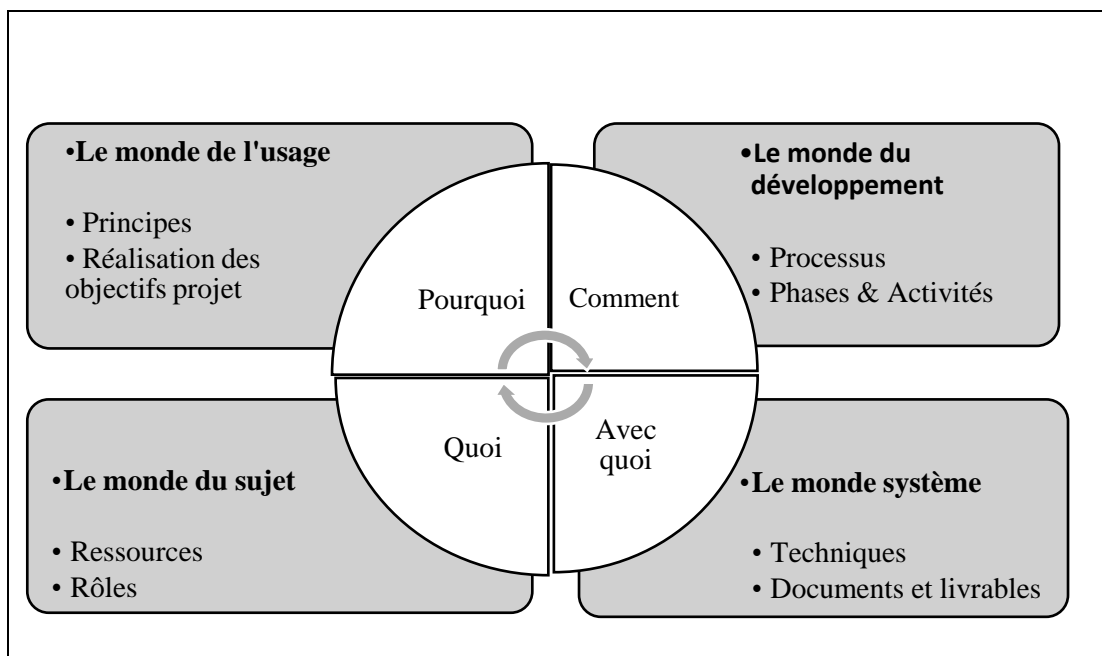


Figure 23 : Cadre de référence des méthodes Prédictives

IV.4. Constitution des concepts

L'approche prédictive ou prédictive se concentre sur l'identification précoce de la **portée**, des **coûts** et de la **durée** du projet. Les changements sont activement évités et étroitement gérés. L'équipe de projet suit le principe central - **planifier avant d'agir**.

L'approche prédictive permet de développer un projet de manière systématique. Passer de la conception du projet à la planification, puis à l'exécution et à la clôture. Cette méthode est préférée lorsque le produit à livrer est clairement compris. Aussi, lorsqu'un produit n'a de valeur que lorsqu'il est livré dans son intégralité.

Le Guide PMBOK ne décrit pas une méthodologie de gestion de projets. C'est un ensemble de bonnes pratiques et de processus. Dans la plupart des cas, ils couvrent bien plus que n'importe quel projet. Il existe une annexe A1 dans le guide PMBOK. Il décrit une norme de gestion de projets. En général, nous pouvons considérer comme une méthodologie basée sur un plan.

Néanmoins, vous pouvez utiliser tous les processus et outils de PMBOK Guide dans d'autres méthodologies comme PRINCE 2. Ou vous pouvez combiner certaines des pratiques dans votre propre méthodologie.

Avantages :

- Parfait pour les projets critiques et de grande valeur. Les approches basées sur un plan vous obligent à effectuer une planification complète dès le départ. Il comprend la collecte des exigences, la définition de la portée, l'élaboration des budgets et l'identification des risques. A chaque étape d'un projet, vous aurez une meilleure compréhension de vos chances de réussite.
- Les bonnes pratiques couvrent tous les aspects d'un projet. Selon la nature, la complexité et la taille, vous pouvez choisir un ensemble approprié de processus, d'outils et de techniques. Le guide PMBOK contient toutes les informations nécessaires.
- Les méthodologies prédictives sont évolutives. Avec le même ensemble de processus, vous pouvez gérer des projets, des programmes et des portefeuilles de projets petits, moyens et grands. Les concepts et le principe sont les mêmes.

Difficultés :

- Les changements sont coûteux. En général, les changements sont activement évités. Cela ne signifie pas que vous ne pouvez pas apporter de modifications. Vous avez juste besoin d'un processus complet pour les gérer afin de rester dans les limites des contraintes. Néanmoins, vous pouvez facilement appliquer une approche de planification par vagues glissantes. Gérez simplement une partie peu claire et ambiguë d'un projet à un niveau supérieur.
- La compétence est requise. Vous n'avez pas besoin de beaucoup de connaissances et d'expérience pour adopter une méthodologie établie. Cependant, les méthodologies axées sur les plans nécessitent généralement des activités de surveillance et de contrôle. Ce qui à son tour peut nécessiter des changements de méthodologie. Sans maîtrise des approches planifiées, il sera difficile de prendre des décisions éclairées. Heureusement, vous pouvez trouver beaucoup de formation sur les approches PMI et PRINCE 2. Et ils les fournissent à différents niveaux.

IV.5. Méthode prédictive PRINCE 2

Projects In Controlled Environment est une méthode orientée processus développée par le gouvernement britannique et accréditée par Axelos, basée sur les meilleures pratiques en gestion de projets. Cette méthodologie flexible s'applique à toutes les entreprises et uniquement pour un projet à portée définie. « Construit sur sept principes, thèmes et processus, PRINCE2 peut être adapté pour répondre à des exigences spécifiques » [2]. Dans cette méthodologie (voir Figure 19), chaque processus est déterminé en fonction des éléments d'entrée, des résultats, des objectifs et des activités.

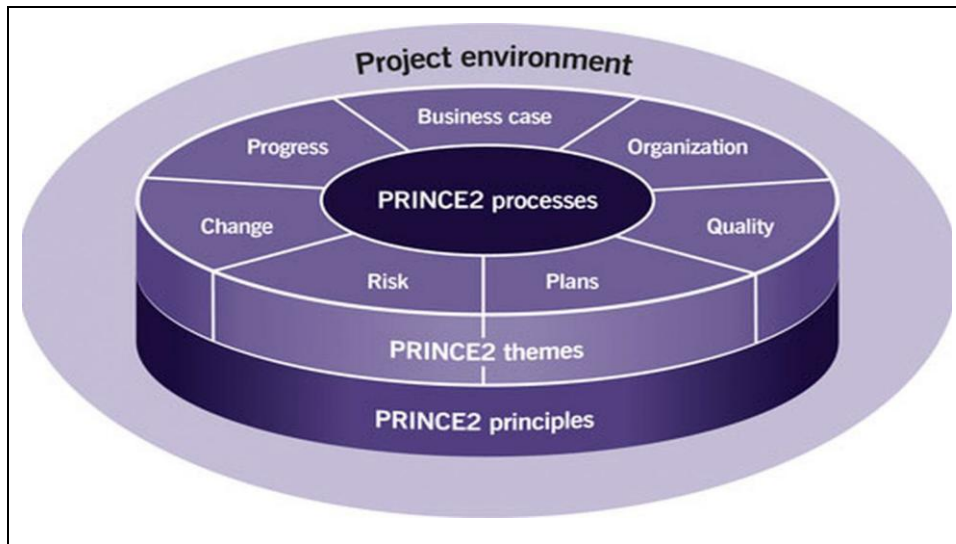


Figure 24 : Méta-modèle PRINCE 2 [25]

La méthode PRINCE 2 se base sur sept principes, sept thèmes et sept processus, ci-dessous l'explication des concepts situés dans ce référentiel :

IV.5.1. Rôles

Prince 2 se base sur 7 niveaux de rôles majeurs à savoir :

- **Le client ou l'exécutif** : C'est la personne ou l'organisation qui a commandé le projet et qui paiera pour cela. Le client sera le propriétaire de l'analyse de rentabilisation et aura la responsabilité ultime de s'assurer que le projet tient sa promesse. Le client sera le président du comité de projet, qui comprend également l'utilisateur principal et le fournisseur principal.
- **L'utilisateur principale** : Il s'agit du représentant de l'organisation ou des personnes qui utiliseront le produit ou service fourni par le projet PRINCE2 : ceux qui en seront directement impactés. Il se peut que le client et l'utilisateur soient la même personne. L'utilisateur senior est responsable de s'assurer que le comité de projet respecte les besoins de l'utilisateur et que ceux-ci sont spécifiés dans le projet.
- **Le fournisseur principal** : Il s'agit de l'organisation ou de la personne qui fournit l'expertise requise pour la réussite du projet PRINCE2. Il ou elle aura les connaissances, l'expérience et l'expertise nécessaires pour informer le comité de

projet de ce qui est impliqué dans la production du livrable fini et s'assurer que les ressources appropriées sont fournies pour l'achèvement du projet.

- **Le chef de projet :** Le chef de projet organise et contrôle le projet, choisit l'équipe et s'assure que le projet est achevé dans les délais, le budget et les normes de qualité requises. Le travail essentiel de planification de projet est du ressort du chef de projet.
- **Le chef d'équipe :** Il peut y avoir un ou plusieurs chefs d'équipe, selon la taille du projet. Il n'est pas rare que le chef de projet assume le rôle de chef d'équipe sur des projets plus petits. Le chef d'équipe gère le travail opérationnel quotidien du projet PRINCE2, en s'assurant que le produit est fabriqué dans la tolérance convenue de temps, de coût et de qualité.
- **L'autorité de changement :** Il s'agit d'un individu ou d'un groupe qui examinera toute demande de modification des spécifications du projet. Un budget de modification sera fourni et l'autorité de modification a le pouvoir d'accepter ou de refuser les demandes de modification. Parfois, le chef de projet assumera cette responsabilité.
- **L'assurance projet :** L'assurance du projet fournit une vue indépendante de l'avancement du projet à des intervalles convenus. Ce rôle doit être indépendant du chef de projet et est souvent assumé par le comité de projet. Cependant, une équipe indépendante chargée de l'assurance du projet peut également être nommée par le comité de projet. Il incombe à cette équipe de s'assurer que le projet reste sur la bonne voie dans les trois domaines de l'assurance commerciale, de l'assurance des utilisateurs et de l'assurance des fournisseurs.

IV.5.2.Principes

La méthode PRINCE 2 se base sur 7 principes qui constituent le fondement de la méthodologie :

- Justification continue du business case (le cas d'affaire)
- Apprendre des expériences
- Définir les rôles et les responsabilités

- Découper le projet en étapes gérables
- Gérer par exception
- Se focaliser sur les produits finaux
- Adopter la méthode à l'environnement

IV.5.3. Thèmes

Les thèmes PRINCE2 sont des domaines de gestion de projets qui doivent être abordés en permanence tout au long du projet. Il existe 7 thèmes PRINCE2 qui sont basés sur les principes et ils sont appliqués tout au long du projet lors de l'application des processus :

- Business Case
- Organisation
- Qualité
- Planification
- Gestion des risques
- Gestion des changements
- Progression / Avancement

IV.5.4. Processus

Un processus est un ensemble structuré d'activités conçues pour atteindre un objectif spécifique. PRINCE2 a des activités pour démarrer, planifier, exécuter, contrôler et fermer un projet. Il regroupe ces activités en processus :

- Diriger le projet (DP)
- Elaboration projet (EP)
- Initialiser le projet (IP)
- Gérer une Limite de Séquence (LS)
- Contrôler une Séquence (CS)

- Gérer la Livraison des Produits (LP)
- Clore le Projet (CP)

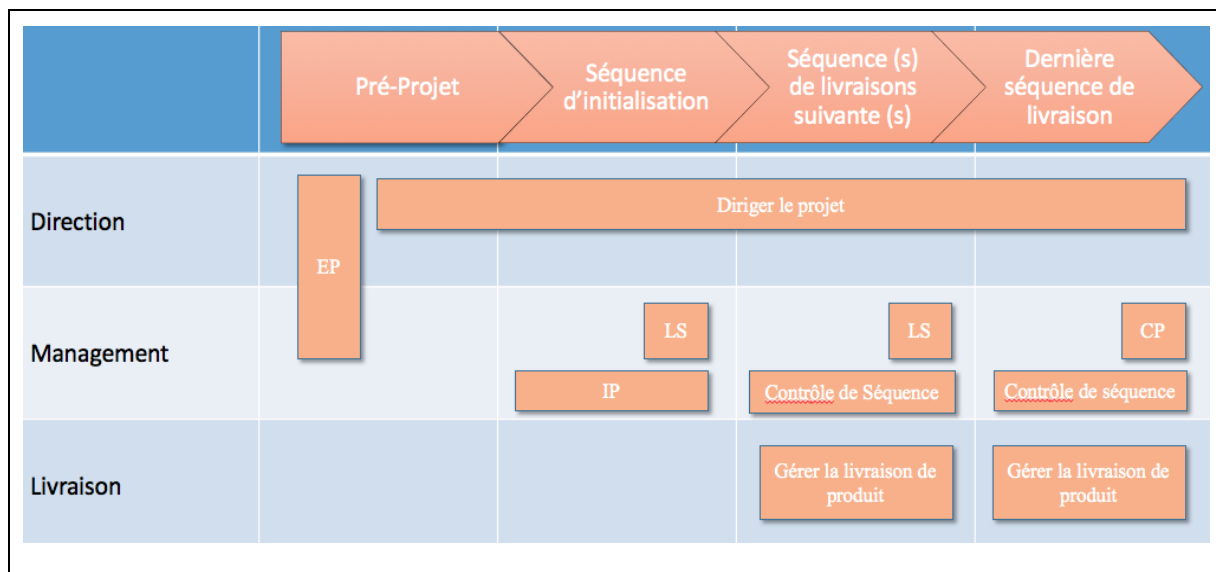


Figure 25 : Processus de la méthode PRINCE 2.

IV.5.5.Livrables

Prince 2 comporte 26 produits de gestion, ci-dessous les principaux livrables :

- A1 Plan de Revue des Bénéfices
- A2 Cas d'Affaire
- A4 Stratégie de Communication
- A5 Enregistrement de configuration
- A6 Stratégie de configuration
- A16 Plan (couvre les Plans au niveau Projet, Séquence et équipe)
- A17 Description(s) de Produit
- A18 Rapport d'état du produit
- A19 Exposé du Projet
- A20 Documentation d'initialisation de projet
- A21 (La)Description de Produit du Projet

- A22 Stratégie qualité
- A23 Registre qualité
- A24 Stratégie des Risques
- Mandat
- Cas d'affaire (Business case)
- Plan Projet
- PAQ : Plan assurance qualité
- Documentation d'initiation du projet
- Documentation de clôture de projet

IV.5.6. Proposition d'un méta-modèle pour PRINCE 2

En analysant les concepts PRINCE 2, nous remarquons que les différents rôles sont liés avec les différentes phases avec des relations soit de type gérer, et que les activités sont aussi liées avec les produits de travail (workproduct) avec des relations de type produire.

Ainsi, ci-dessous une proposition de méta-modèle PRINCE 2 :

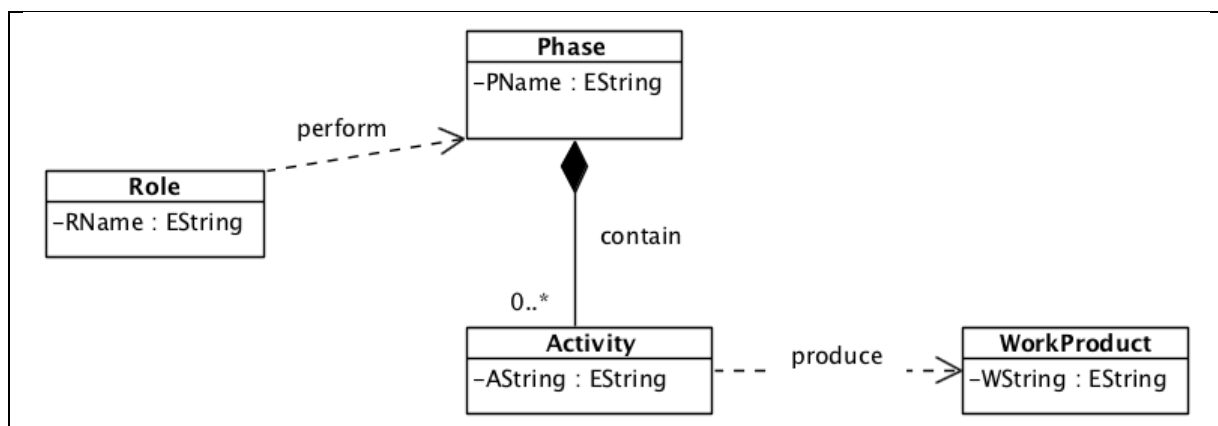


Figure 26 : Méta-modèle PRINCE 2.

Le tableau ci-dessous redéfinit les composants du méta-modèle :

Entité	Type	Description
Phase	Meta-Class	Une phase a un nom

		Une phase contient des activités et des livrables Una phase a un identifiant unique
Activité	Meta-Class	Une activité a un nom Une activité contient plusieurs livrables
Rôle	Meta-Class	Un rôle a un nom Un rôle peut produire plusieurs livrables
Livrable	Meta-Class	Un livrable a un nom

Tableau 18 : Structure de la méthode PRINCE 2.

IV.6. Proposition d'un méta-modèle générique pour les méthodes Prédictives

En se basant sur les concepts évoqués plus haut, nous avons proposé ce méta-modèle qui résume les relations entre les différents concepts des méthodes prédictives :

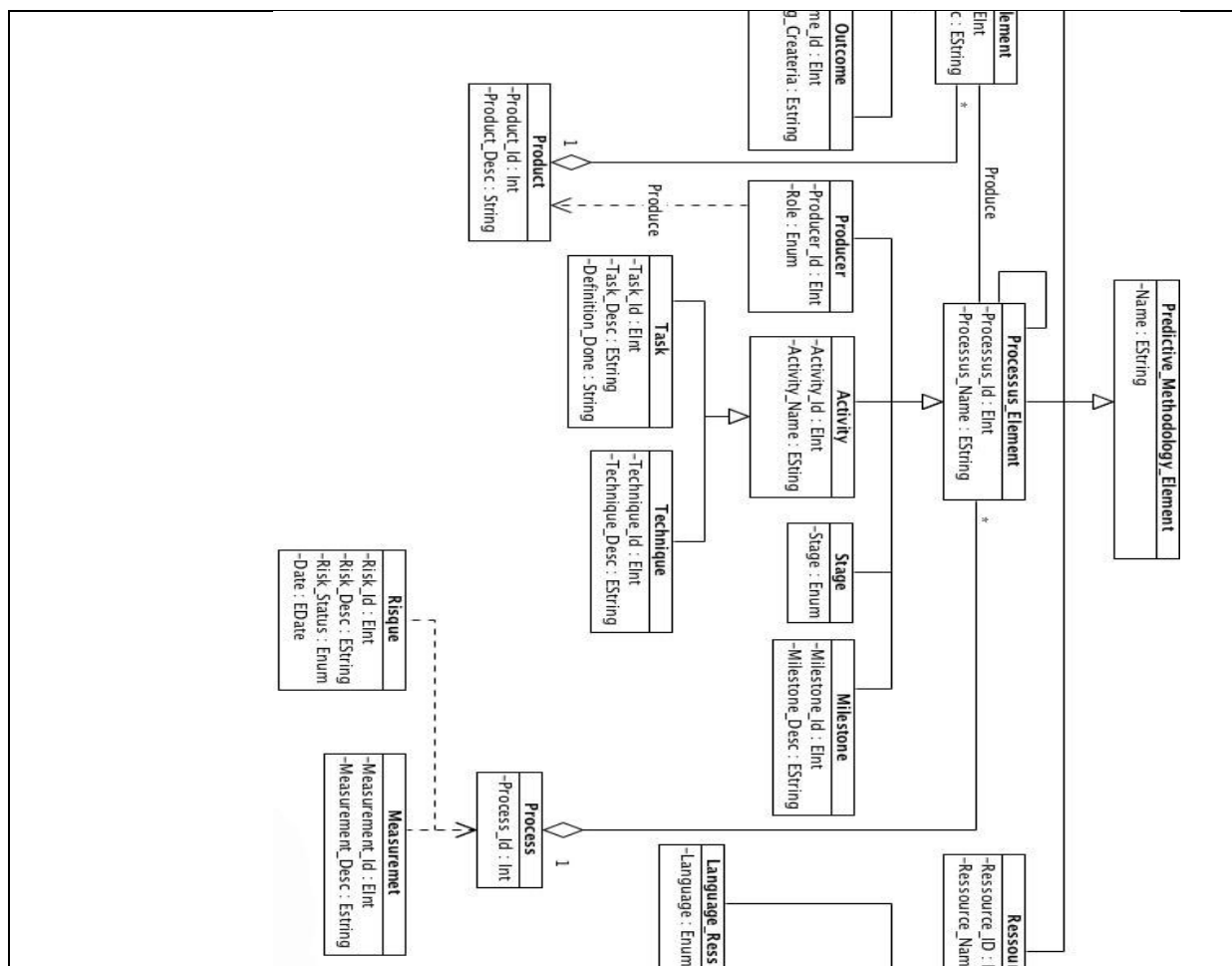


Figure 27 : Méta-modèle des méthodes Prédicatives.

Le tableau ci-dessous redéfinit les composants du PIM multidimensionnel (dimention Produit, Processus et Ressource) :

Classe	Description	Instances possibles
1- Process Element	Éléments de méthodologie transformant les entrées en sorties. La relation entre un ou plusieurs processus éléments génère le processus.	Pratiques de gestion, étapes du Performance du processus
1.1- Activity	Un élément de processus abstrait de base qui modélise une opération fonctionnellement cohérente qui peut être effectuée par un ou plusieurs producteurs	Pratique (programmation par paires, Refactoring ...), Activité, Tâche, Technique
1.1.1- Task	La plus petite unité de travail gérable.	Rédiger des tests unitaires, des cas d'utilisation de modèles, Ecrire le code
1.1.2- Technique	Des moyens spécifiques pour atteindre un objectif.	Entretien, réflexion ateliers

1.1.3- Measurement	Des moyens spécifiques pour effectuer des mesures sur les processus	Diagramme de Gant, diagramme de performance ..
1.1.4- Risk	Un élément essentiel en ce qui concerne les méthodes Prédicatives	Niveau du risque, statut du risque ...
1.2- Producer	Quelqu'un ou quelque chose qui exécute une unité de travail soit directement ou indirectement (c.-à-d. crée, évalue, itère ou maintient).	Chef de projet, équipe, testeur, organisation, outil
1.3- Stage	Représente des intervalles de temps gérés ou des points temps, au sein d'un projet, c'est-à-dire l'aspect temporel du processus.	Phase
1.4- Milestone	Il s'agit des jalons des projets	Jalon, point de décision, ...
2- Product Element	Tout artefact développé et produit pendant le projet. Cet élément équivaut au "Travail Élément produit".	Version, modèles UML, technique documents
2.1- Activity	Un livrable est un élément de produit qui fournit un description et définition de l'emballage d'autres éléments de produits, et peuvent être livrés à un ou partie externe.	Activité, rapport technique
2.2- Documents	Une ressource identifiable fournie par l'équipe de développement qui prend en charge le processus global de développement logiciel.	Document de cadrage, d'architecture logicielle, Modèle de cas d'utilisation ...
2.3- Outcome	Le résultat est un élément de produit qui fournit une description et une définition des éléments de produit non tangibles. Une différence clé entre les résultats et artefacts est que les résultats ne sont pas candidats pour récolter comme actifs réutilisables.	Plan de cadrage, livrable produit ..
3- Resource	Éléments méthodologiques utilisés dans la réalisation du projet.	Rapports standards, documentation d'expertise
3.1- Language	Vocabulaire (c'est-à-dire un ensemble de termes) combiné avec un ensemble de règles grammaticales utilisées pour produire un ou plusieurs éléments de produits.	Langage de modélisation, Langage d'implémentation, base de données Langue
3.1- Guideline	Règles et directives sur l'utilisation appropriée de un élément méthodologique donné.	Norme, directives de la méthodologique

Tableau 19 : Structure du méta-modèle des méthodes Prédicatives.

IV.7. Conclusion

Dans ce chapitre, nous avons présenté les différents concepts des méthodes Prédictives, puis nous avons cité comme exemple la méthode PRINCE 2, nous avons remonté plusieurs concepts qu'il faudra analyser et les comparer avec d'autres méthodologies pour avec des concepts uniques. Le modèle généré renferme plusieurs types de relation qu'il faudra analyser dans les chapitres suivant avant d'appliquer notre approche.

Dans le chapitre suivant nous allons étudier une autre famille de méthodologie de gestion de projets qui a une philosophie différente, il s'agit des méthodes Agile.

Chapitre V : Proposition d'un méta-modèle pour les méthodes agiles de la gestion de projets

V.1. Introduction

Les méthodes Agiles partagent une grande partie la même philosophie globale, ainsi que plusieurs des mêmes caractéristiques et pratiques. Hors, du point de vue de la mise en œuvre, chacun a son propre mélange unique de pratiques, de terminologies et de tactiques (Agile Scrum Methodology, Lean Software Development, Kanban, Extreme Programming (XP), Crystal, Dynamic Systems Development Method (DSDM), Feature Driven Development (FDD), SCRUM, etc).

Dans ce chapitre, nous allons expliciter les différents concepts des méthodologies Agile, les principes, les rôles, les évènements et les livrables de la méthodologie, puis nous allons proposer un méta-modèle des méthodologies Agile.

V.2. Principes des méthodes Agiles

Les méthodes Agiles forment « un cadre dans lequel les gens peuvent résoudre des problèmes adaptatifs complexes, tout en fournissant de manière productive et créative des produits de la plus haute valeur possible » [89].

Beaucoup de choses ont changé depuis 2001 où il y'a eu une révolution agile. Les appareils mobiles que nous utilisons sont de plus en plus puissants et les attentes des utilisateurs quant à ce que ces appareils peuvent offrir ne cessent de croître. En conséquence, le rythme du changement pour les développeurs a augmenté de façon exponentielle, et leur travail ne peut vraiment être accompli qu'en adoptant des méthodes Agiles.

Même l'agilité elle-même a changé : différentes approches pour appliquer l'agilité à grande échelle continuent de croître et d'évoluer dont le référentiel SAFE. La méthodologie s'est même étendue au-delà du service informatique, les praticiens agiles changeant les services juridiques, opérationnels ou marketing de leur organisation.

Quelle que soit l'évolution des pratiques et des techniques agiles, la base des méthodes Agiles reste enracinée dans des pratiques positionnées pour permettre des environnements collaboratifs où diverses équipes peuvent continuellement apprendre, s'améliorer, grandir et produire.

En tant que tel, j'ai voulu revoir les 12 principes de la méthodologie Agile qui aident à former le fondement Agile de la productivité collaborative. Ces 12 principes - inspirés et soutenant le Manifeste Agile - sont des pratiques directrices qui aident les équipes à mettre en œuvre et à exécuter avec agilité la production de développement logiciel et bien plus encore.

- **Satisfaire le client** : Notre priorité absolue est de satisfaire le client grâce à une livraison rapide et continue de logiciels précieux.
- **Bienvenue au changement** : Accueillez les besoins changeants, même tard dans le développement. Les processus agiles exploitent le changement pour l'avantage concurrentiel du client. Il est difficile de ne pas avoir une vague de désespoir en pensant aux demandes de changement, mais le changement est bon si vous pouvez y réagir assez rapidement. Le changement signifie que vous vous rapprochez des besoins des clients et c'est une bonne chose.

- **Livrer fréquemment :** Fournissez fréquemment des logiciels fonctionnels, de quelques semaines à quelques mois, avec une préférence pour les délais plus courts. Plus tôt vous livrez un logiciel incrémentiel, plus le retour d'information est rapide et vous pouvez identifier rapidement un mauvais virage ou une mauvaise communication avec le client.
- **Travailler ensemble :** Les professionnels et les développeurs doivent travailler ensemble au quotidien tout au long du projet. Il est logique que le client fasse partie de l'équipe. Après tout, les développeurs et les clients ont le même objectif.
- **Construire des projets :** Construisez des projets autour d'individus motivés. Donnez-leur l'environnement et le soutien dont ils ont besoin, et faites-leur confiance pour faire le travail. Les projets agiles mettent l'accent sur des équipes auto-organisées qui sont instinctivement capables de gérer à la fois elles-mêmes et le travail. La microgestion de projets n'est plus nécessaire ni bénéfique.
- **Temps en face à face :** La méthode la plus efficace et la plus efficace pour transmettre des informations à et au sein d'une équipe de développement est la conversation en face à face. La colocation est l'idéal. La communication osmotique - où vous absorbez une partie de la conversation pour savoir si vous faites partie de la conversation réelle ou non - est un avantage très réel de la colocalisation. Malheureusement, les équipes de développement sont souvent réparties. Si les équipes ne sont pas colocalisées, tous les efforts doivent être faits pour communiquer souvent et pour accroître l'utilisation des techniques de communication technique.
- **Mesure du progrès :** Le logiciel de travail est la principale mesure du progrès. Lorsque vous vous concentrez sur le suivi du plan, vous êtes généralement trop impliqué dans la mise à jour de la documentation - en vous détournant de l'objectif du projet. Lorsque vous faites du logiciel fonctionnel la principale mesure des progrès, vous le promouvez au cœur du projet.
- **Le développement durable :** Les processus agiles favorisent le développement durable. Les sponsors, les développeurs et les utilisateurs devraient être en mesure de maintenir un rythme constant indéfiniment. Pensez à la conciliation

travail-vie personnelle. Vous vous souvenez de la fin du tableau de bord du projet? Tout le monde a travaillé presque 24 heures sur 24 pour terminer le projet. Peu importe l'impact sur la qualité, qu'en est-il de l'impact sur l'équipe ? Agile s'efforce de maintenir un niveau d'activité constant qui se traduit par une vitesse constante. Un résultat important est une meilleure capacité de prévision.

- **Attention continue :** Une attention continue à l'excellence technique et à une bonne conception améliore l'agilité. Bien qu'un design élégant ait du sens, une solution qui traversera l'épreuve du temps est encore plus précieuse. Peut-être encore plus important une solution qui a la capacité d'être mise à jour pour la maintenir à jour. A quoi bon un design élégant s'il ne peut pas maintenir sa valeur grâce aux cycles de mise à jour et de maintenance.
- **Rester simple :** L'art de maximiser la quantité de travail non effectué - essentiel. Bien sûr, les fonctionnalités les plus fiables sont celles qui ne sont pas encore construites ... elles ne peuvent pas échouer. Mais cela mis à part, près de 30% des fonctionnalités que nous développons sont rarement ou jamais utilisées. Agile est impitoyable sur la réduction des fonctionnalités qui ne prêtent pas de valeur.
- **Équipes organisées :** Les meilleures architectures, exigences et conceptions émergent d'équipes auto-organisées. Des équipes auto-organisées qui sont également interfonctionnelles. Qui mieux reconnaître les problèmes avant qu'ils ne deviennent de véritables obstacles ? bien sûr, les personnes les plus proches de la solution.
- **Réfléchir pour l'efficacité :** À intervalles réguliers, l'équipe réfléchit à la façon de devenir plus efficace, puis ajuste et ajuste son comportement en conséquence. Nous avons tous été sur des projets qui se terminent par un AAR, After Action Review. Ces évaluations aident le prochain projet, mais pas celui que vous venez de terminer. Agile définit plusieurs cérémonies et parmi celles-ci, la rétrospective est importante. Généralement organisé à la fin de chaque Sprint / Itération, c'est un moyen pour les équipes de détecter et d'améliorer les comportements avant qu'ils n'aient un impact énorme et néfaste sur le projet.

Pour créer des équipes hautement performantes, les méthodes Agiles valorisent les individus et les interactions plutôt que les processus et les outils. Toutes les pratiques Agile, y compris les 12 principes Agile, cherchent à accroître la communication et la collaboration grâce à des cycles d'inspection et d'adaptation fréquents et à la cohérence des échanges collaboratifs enracinés dans une base de vérité Agile - transparence, confiance, respect et engagement.

Nous proposons d'examiner les résultats des travaux empiriques qui ont été consacré à clarifier les différences entre les méthodologies de gestion de projets Agile, nous allons prendre comme exemple trois méthodologies les plus connus :

	Scrum	XP	Lean
Principes	Transparence, suivi et adaptation	Communication, simplicité, feedback, respect et courage	Elimination du gaspillage, livraison rapide, qualité intrinsèque, gestion des compétences, responsabilité des personnes et optimisation de l'ensemble.
Pratiques managériales	Réunions quotidiennes, réunions de planification de l'itération, revue de l'itération et rétrospective de l'itération.	Conception simple Tests unitaires et fonctionnels Remaniement des codes Déploiement quotidien Intégration continue. Planning game, stand-up meeting, programmation en paire, appropriation collective des codes, client sur le site.	Système pull, développement piloté par les tests, développement itératif, sessions kaizen.
Artfacts	Product-backlog, sprint- backlog, burndown charts.	Story-cards, story-board, user- stories	L'andon, value stream mapping, pareto des causes, 5 Pourquoi, A3 problem solving.

Tableau 20 : Différences entre les méthodologies de gestion de projets Agile.

V.3. Proposition d'un cadre de référence

En se basant sur l'explication des méthodes Agiles, nous avons mis les concepts sur la cadre de référence de Rolland en 1998 [5], voir les quatre mondes du modèle ci-dessous :

- **Le monde du sujet** contient la réponse à **Qu'est-ce** que la gestion de projets en mode : Un ensemble de processus à respecter pour mener à terme un projet, il s'agit d'un ensemble de rôles et indicateurs pour diriger le projet.
- **Le monde du système** comprend les spécifications des projets **Avec Quoi** nous gérons les projets : un ensemble de techniques et des livrables pour le suivi, en utilisant des métriques.
- **Le monde de l'usage** décrit l'environnement organisationnel, il répond à **Pourquoi** nous utilisons la gestion de projets : Il s'agit des principes des méthodes Agiles; certains se basent sur les contraintes, d'autres sur les délais. Il s'agit d'un ensemble de principes pour créer de la valeur, gérer le temps et améliorer la qualité des projets.
- **Le monde du développement** contient les activités, c'est plus **Comment** la gestion de projets obtiendra des résultats : Thèmes à respecter en utilisant des cycles et des activités pour produire les artefacts.

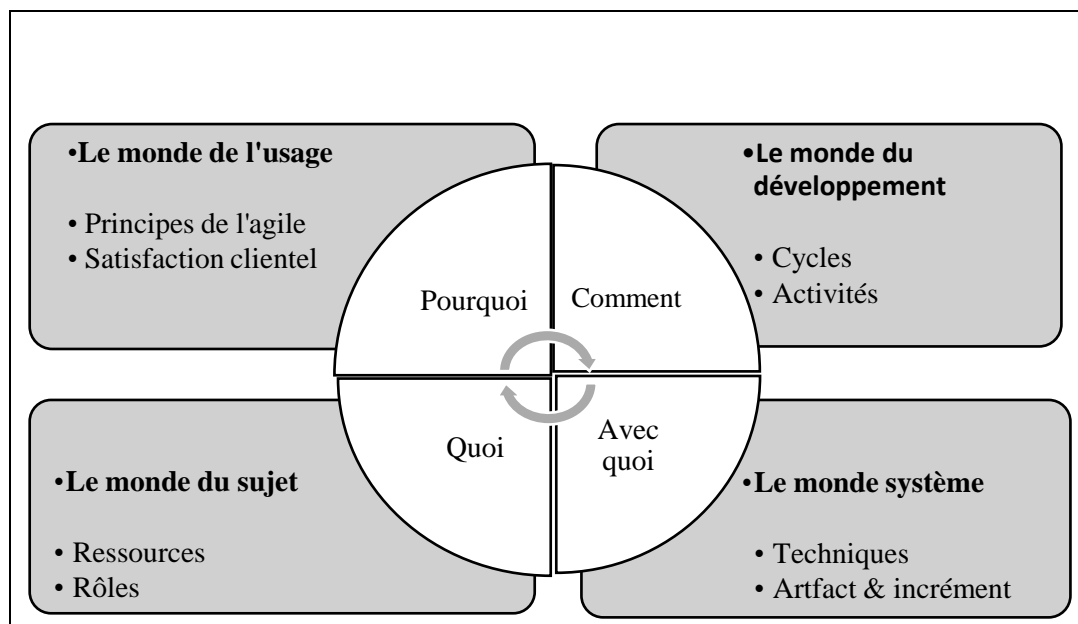


Figure 28 : Cadre de référence des méthodes Agiles

V.4. Constitution des concepts

Adaptive Project Framework (APF), également connu sous le nom de Adaptive Project Management (APM), tient compte des facteurs inconnus qui peuvent surgir au cours d'un

projet. Il prépare les équipes à anticiper les imprévus et à réagir. Considérez son principe de base comme « l'apprentissage par l'action ».

En abordant les projets en sachant que les composants clés sont constamment en mouvement, les équipes peuvent adopter un état d'esprit flexible pour apprendre continuellement en réévaluant les résultats et les décisions tout au long d'un projet.

Cela nécessite une communication régulière avec les parties prenantes à tous les niveaux pour que l'équipe s'adapte efficacement, voir la figure plus bas :

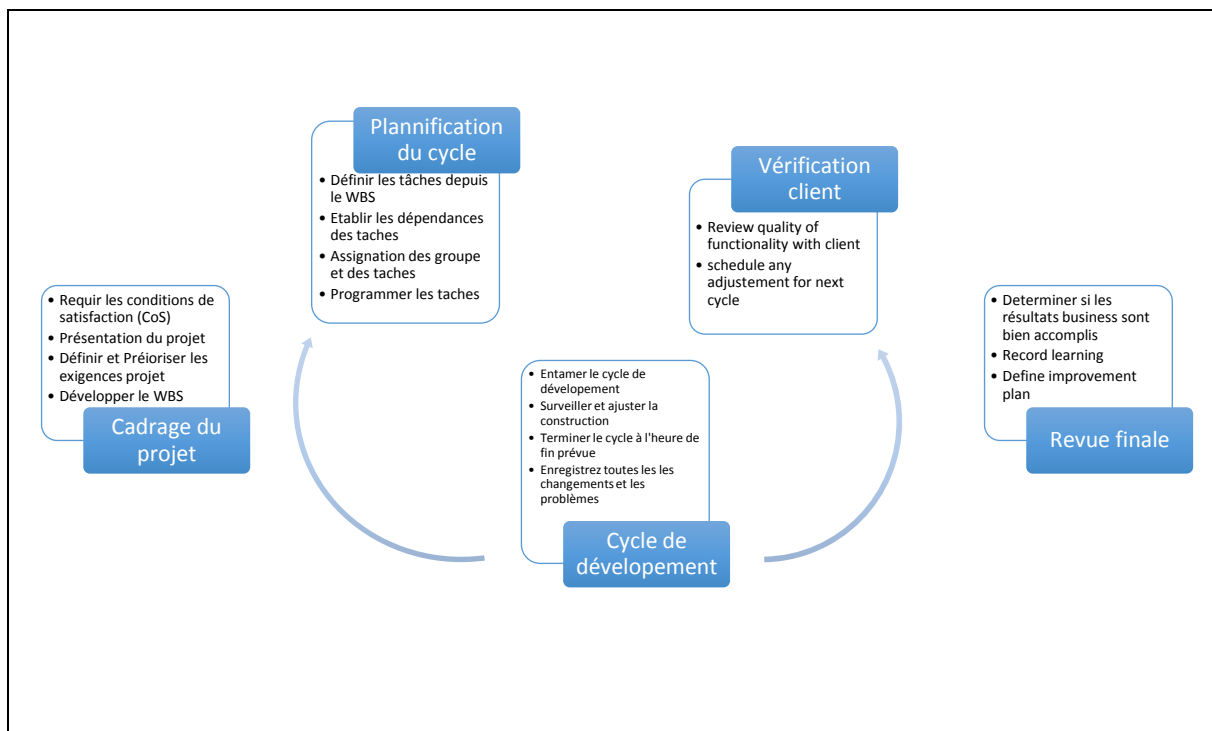


Figure 29 : Aperçu du cadre de projet Agile

V.4.1. Cadrage du projet

La première étape de la phase de cadrage consiste à identifier les conditions de satisfaction du projet. En d'autres termes, les parties prenantes doivent définir quels sont les objectifs du projet et à quoi ressemble un résultat réussi.

L'énoncé de présentation du projet est le produit livrable du CdS. Ce document décrit la CdS finale approuvée signée par toutes les parties prenantes.

Reportez-vous au PoS tout au long du projet pour évaluer l'efficacité de vos processus au fur et à mesure. Bien que le PoS soit approuvé au début, il peut servir de document évolutif dans le cadre de l'APF.

Si la portée ou les objectifs du projet évoluent, ajustez le PoS pour s'adapter à l'évolution du CoS. N'oubliez pas de faire participer toutes les parties prenantes à l'itération pour vous assurer que tout le monde est sur la même longueur d'onde.

La hiérarchisation des exigences du projet définit en outre la portée du projet et désigne également l'ordre de mise en œuvre.

Il existe plusieurs façons de hiérarchiser les exigences, mais l'approche de base consiste à collaborer avec les parties prenantes pour organiser les exigences en désignations pondérées (élevée, moyenne, faible) avec des valeurs numériques.

Cependant, gardez à l'esprit que si les parties prenantes doivent être impliquées dans ce processus, l'analyste et le chef de projet doivent guider la discussion pour s'assurer que les priorités sont attribuées de manière réaliste.

Un WBS décompose les composants et les processus du projet en sections gérables. Il fournit à la fois le cadre pour l'estimation et le contrôle des coûts ainsi que des conseils pour l'élaboration du calendrier.

Créer un WBS clair peut sembler intimidant, mais un outil comme Lucidchart fait la plupart des démarches pour vous. Des fonctionnalités intuitives telles que la fonctionnalité glisser-déposer, les options de formatage, la collaboration en temps réel, l'historique des révisions et des liens vers des documents externes facilitent la collaboration avec votre équipe pour créer et mettre en œuvre un organigramme WBS efficace.

La dernière étape de la phase de cadrage consiste à évaluer et à prioriser le « triangle de portée ». Le triangle de portée est un modèle des contraintes de qualité de votre projet: coût, calendrier et étendue. Pour hiérarchiser les contraintes de votre projet, classez les limites comme « inflexibles », « adaptables » ou « peuvent concéder ».

V.4.2. Planification du cycle

Une fois votre projet défini, la phase suivante est la planification. Ce cycle comporte quatre étapes de base :

- Définir des tâches à partir de WBS
- Établir des dépendances
- Regrouper et attribuer des tâches
- Planifier le travail

L'objectif au cours de cette phase est de mieux définir et planifier les tâches que vous allez mettre en œuvre pendant la phase de développement.

Cette étape comprend la décomposition des tâches individuelles de la WBS, l'identification des dépendances des tâches (ou l'ordre dans lequel les tâches doivent être terminées), l'attribution du travail aux membres de l'équipe et la planification des délais et des délais pour chaque partie du projet.

V.4.3. Cycle de développement du projet

Une fois votre projet défini et planifié, vous êtes prêt à vous lancer et à commencer le développement. Cette phase comprend plusieurs éléments clés :

- Début des travaux
- Surveillance et ajustement de la construction du cycle
- Terminer le cycle à l'heure de fin prévue
- Planification des fonctionnalités incomplètes pour le prochain cycle
- Enregistrement de toutes les demandes de changement / idées d'amélioration
- Enregistrement et suivi de tous les problèmes

La principale différence entre la méthodologie de développement Agile et la gestion de projets prédictive (TPM) est que le calendrier défini pendant les cycles de cadrage et de planification est fixe.

Dans un projet TPM, la date d'achèvement planifiée peut être repoussée pour répondre à un livrable. Mais dans la méthodologie de développement agile, si un délai expire, le livrable est mis de côté et redéfini pour le cycle suivant.

V.4.4. Vérification Client

La phase de point de contrôle client est une partie cruciale du cadre de projet agile. C'est le moment de vérifier avec le client pour examiner la qualité de la fonctionnalité fournie dans le cycle de construction.

Sur la base de cette évaluation, le client et le chef de projet travailleront ensemble pour planifier les ajustements ou les corrections de cap nécessaires pour la prochaine itération.

À ce stade, le processus se répète jusqu'à ce que le budget du projet soit épuisé. En d'autres termes, l'équipe revient au cycle de planification à travers les phases de construction et de point de contrôle jusqu'à ce que le projet soit terminé.

V.4.5. Revue finale

À la fin d'un projet APF, les gestionnaires et les parties prenantes se rencontrent pour évaluer le succès du projet, noter ce qui a été appris et définir les améliorations futures du processus.

V.5. Méthode Agile SCRUM

Scrum est une méthodologie ou un cadre de gestion de projets agile utilisé principalement pour les projets de développement de logiciels dans le but de fournir de nouvelles capacités logicielles toutes les 2 à 4 semaines. C'est l'une des approches qui ont influencé le **Manifeste Agile**, qui articule un ensemble de valeurs et de principes pour guider les décisions sur la manière de développer plus rapidement des logiciels de meilleure qualité.

La méthode SCRUM implique que le projet progresse à travers la mise en place de séries de "sprints". Chaque fois qu'un sprint est lancé, une réunion de planification est organisée afin que chaque membre de l'équipe puisse s'engager sur le nombre de tâches qu'il pourra effectuer, ainsi que la création du "backlog sprint", qui est la liste globale des tâches à faire pendant un sprint.

Chaque jour du sprint, tous les membres de l'équipe (ainsi que le chef de produit et le maître SCRUM) doivent assister à la réunion SCRUM quotidienne. Cela ne devrait pas durer plus de 15 minutes et permet aux membres de l'équipe de partager avec les autres ce qu'ils ont fait la veille, ce sur quoi ils travaillent le même jour et d'identifier les problèmes qui

pourraient entraver le bon déroulement du sprint. Cette réunion vous permet de synchroniser tous les membres de l'équipe.

La fin d'un sprint est marquée par une session de débriefing pour présenter le travail accompli au chef de produit et partager des informations pouvant influencer le prochain sprint.

La méthode propose donc une approche très itérative de la gestion de projets. Le succès de cette méthode repose sur le strict respect des rôles de chacun, ainsi que sur des cycles de travail courts, rigoureux et flexibles. Le respect de ces règles accorde à la fois une grande autonomie et liberté à toute l'équipe. Compte tenu de la complexité croissante des projets innovants, la méthode SCRUM semble être la meilleure solution pour répondre aux exigences d'exécution de ces derniers, ce qui explique le succès qu'elle rencontre aujourd'hui.

V.5.1. Rôles

Il existe trois rôles essentiels dans le framework SCRUM ; l'équipe SCRUM (developpement team), Product Owner et le Scrum Master. Dans ce qui suivra nous allons exposer les différents rôles.

- **Equipe SCRUM :**

L'équipe SCRUM doit avoir deux caractéristiques essentiels ;

- **Auto organisé :** Pas de manager, pas de hiérarchie et l'équipe est solidaire et soudé.
- **Multidisciplinaire :** l'équipe de développement possède toutes les compétences pour produire un produit livrable et il n'y a pas de titre pour chaque membre de l'équipe.

La taille de l'équipe SCRUM doit être entre 3 et 9 personnes, rassemblé dans un même espace de travail pour plus de synergie et de communication. D'après l'effet Ringelmann (...) les équipes de petites taille sont plus productives que les équipes larges.

SCRUM peut être étendu à plusieurs équipes et là on parle de SCALED SCRUM ou bien USING SCRUM AT SCALE.

- **Product Owner**

Le Product Owner (PO) décide dans les domaines qui sont ceux d'un chef de projet traditionnel, par exemple la date de livraison du produit, il clarifie la vision produit aux membres de l'équipe, par conséquent il décide du choix stratégique du produit.

Aussi, un PO présent et impliqué fera des choix tactiques comme par exemple l'apparence et le contenu d'un formulaire pour un site marchand. Il est responsable de définir l'objectif du produit et de le partager avec l'équipe qui le développe. Ci-dessous les missions du PO :

- Le PO fournit une vision partagée du produit. Notamment, l'énoncé du problème que le produit veut résoudre, une description du produit qui soit claire pour tout le monde ainsi qu'une liste des fonctionnalités essentielles.
- Le PO définit le contenu du produit. Pour cela, il identifie les fonctionnalités requises et les collecte dans une liste appelée le carnet du produit (Product Backlog)
- C'est le PO qui définit l'ordre dans lequel les parties du produit seront développées en fonction de la valeur qu'elles apportent.

A noter que le PO n'a pas d'autorité formelle sur des personnes de l'équipe développement, le PO a une grande influence sur le produit réalisé.

- **Scrum Master**

Le SCRUM Master a pour responsabilité essentielle d'aider l'équipe à appliquer le framework Scrum et à l'adapter au contexte. Il a une grande influence sur la façon de travailler, sur le processus, comme le Product Owner en a une sur le produit. Ce qui pourrait conduire à qualifier le Scrum Master de Process Owner par équivalence. Ci-dessous les missions du SCRUM Master :

- Un coach facilitateur en ce sens qu'il a la responsabilité de la résolution des problèmes et la gestion des conflits entre les membres de l'équipe de développement.
- Il accompagne et stimule l'équipe de développement.

- Il assure la formation de l'équipe afin de les rendre le plus pluridisciplinaire possible.

V.5.2. Evénements

Les événements SCRUM sont à durée fixe, ce que nous appelons ; des événements Time Boxed :

- **Sprint**

Chaque projet SCRUM est divisé en un ensemble de sprints ou bien itérations (1 à 4 semaines), l'objectif du Sprint est de créer une version utilisable par l'utilisateur final et potentiellement déployable (potentially releaseable).

- **Planification du Sprint**

Chaque Sprint doit être planifié et les tâches arrêtées par l'équipe de développement. Ainsi, la réunion de planification du sprint est une étape importante d'un projet Scrum, elle sert à planifier le travail à réaliser au cours du sprint.

La durée maximum de la planification de sprint est de 8 heures pour un sprint de 4 semaines. Cette durée est proportionnellement inférieure si la durée du sprint est plus courte.

- **Daily SCRUM (Daily Stand Up)**

Réunion de 15 min pour coordonner les réalisations et dépasser les obstacles pour les prochains 24h. Le Daily Scrum a lieu tous les jours du Sprint pour optimiser la collaboration et les performances de l'équipe en inspectant le travail depuis le dernier Daily Scrum et en prévoyant le travail à venir du Sprint. Le Daily Scrum a lieu à la même heure et au même endroit chaque jour pour réduire la complexité.

- **Revue de Sprint ou bien la réunion de revue**

A la fin de chaque Sprint, l'équipe de développement présente au client le livrable et reçoit le feedback, la réunion de revue de sprint est l'avant-dernière événement du sprint et devrait avoir lieu à chaque sprint. Seule la rétrospective suit (juste après) la revue du sprint, et après la rétrospective, un nouveau sprint commence.

- **Rétrospective Sprint**

A la fin de chaque sprint, et avant de commencer le Sprint suivant, l'équipe de développement tient une réunion pour capitaliser sur le sprint terminé et améliorer le processus (leçons apprises).

Il s'agit au maximum d'une réunion de trois heures pour des Sprints d'un mois. Pour les sprints plus courts, l'événement est généralement plus court. Le Scrum Master s'assure que l'événement a lieu et que les participants comprennent son objectif. C'est l'occasion pour l'équipe Scrum de s'améliorer et tous les membres doivent être présents.

Au cours de la rétrospective Sprint, l'équipe discute :

- Ce qui s'est bien passé dans le Sprint
- Ce qui pourrait être amélioré
- Que nous engagerons-nous à améliorer au prochain Sprint

V.5.3. Artefacts

Le terme « artefact » fait référence à un objet qui a été fabriqué par un humain. Les racines latines du mot artefact se traduisent grossièrement par « œuvre d'art ». Ainsi, un artefact est quelque chose que nous fabriquons, soit un outil qui résout un problème, soit une œuvre d'art qui nous inspire. Scrum décrit trois artefacts principaux : le **Backlog Produit**, le **Backlog Sprint** et l'**Incrément Produit** dont l'explication du contenu est décrite plus bas :

- **Product backlog**

C'est un document partagé par l'équipe de développement qui est élaboré progressivement et qui n'est pas figé. Il contient une liste d'histoire utilisateur (User Stories US), chaque story est une description d'un cas d'utilisation simple avec un test d'acceptation (Condition Of Satisfaction COS).

Dans sa définition la plus simple, le Scrum Product Backlog est simplement une liste de tout ce qui doit être fait dans le projet. Il remplace les artefacts traditionnels de spécification des exigences. Ces éléments peuvent avoir une nature technique ou être centrés sur l'utilisateur, par exemple. Sous forme de user stories. Le propriétaire du Backlog produit Scrum est le Product Owner Scrum. Le Scrum Master, l'équipe Scrum et d'autres parties prenantes y contribuent pour avoir une liste de tâches large et complète.

Travailler avec un backlog de produit Scrum ne signifie pas que l'équipe Scrum n'est pas autorisée à créer et utiliser d'autres artefacts. Des exemples d'artefacts supplémentaires peuvent être un résumé des différents rôles d'utilisateur, des descriptions de flux de travail, des directives d'interface utilisateur, des storyboards ou des prototypes d'interface utilisateur. Cependant, ces artefacts ne remplacent pas le Scrum Product Backlog mais complètent et détaillent son contenu.

ID	US	Estimation	Priorité
7	En tant qu'utilisateur non autorisé, je souhaite créer un nouveau compte	3	1
5	En tant qu'utilisateur autorisé, je souhaite me connecter	1	2
10	En tant qu'utilisateur non autorisé, je souhaite me connecter	1	3
3	Créer un script pour purger la base de données	2	4
4	En tant qu'utilisateur autorisé, je veux voir la liste des éléments afin de pouvoir en sélectionner un	1	5
6	En tant qu'utilisateur autorisé, je souhaite ajouter un nouvel élément pour qu'il apparaisse dans la liste	3	6
8	En tant qu'utilisateur autorisé, je souhaite supprimer un élément sélectionné	2	7
9	En tant qu'utilisateur autorisé, je souhaite mettre à jour un élément sélectionné	2	8
1	En tant qu'administrateur, je veux voir la liste des comptes lors de la connexion	3	9
Total		18	

Tableau 21 : Exemple du backlog Produit

- **Sprint Backlog**

Il s'agit d'un ensemble des items à réaliser non pris en charge, il contient les éléments du Backlog produit rassemblés dans un seul Sprint, ainsi que le plan du travail nécessaire pour fournir un incrément de produit utilisable. Il s'agit d'un artefact transparent qui montre tout le travail que l'équipe de développement juge nécessaire pour atteindre l'objectif de sprint convenu. Une mise à jour récente du Guide Scrum introduit également au moins une amélioration de processus hautement prioritaire dans le cadre d'un Sprint Backlog.

Nous pouvons distinguer 4 types de demandes au niveau du sprint Backlog :

- Features (fonctionnalités)

- Bugs
- Travaux techniques
- Acquisition de connaissances

Le type d'items existants dépend des équipes et de leur choix dans la gestion du backlog.

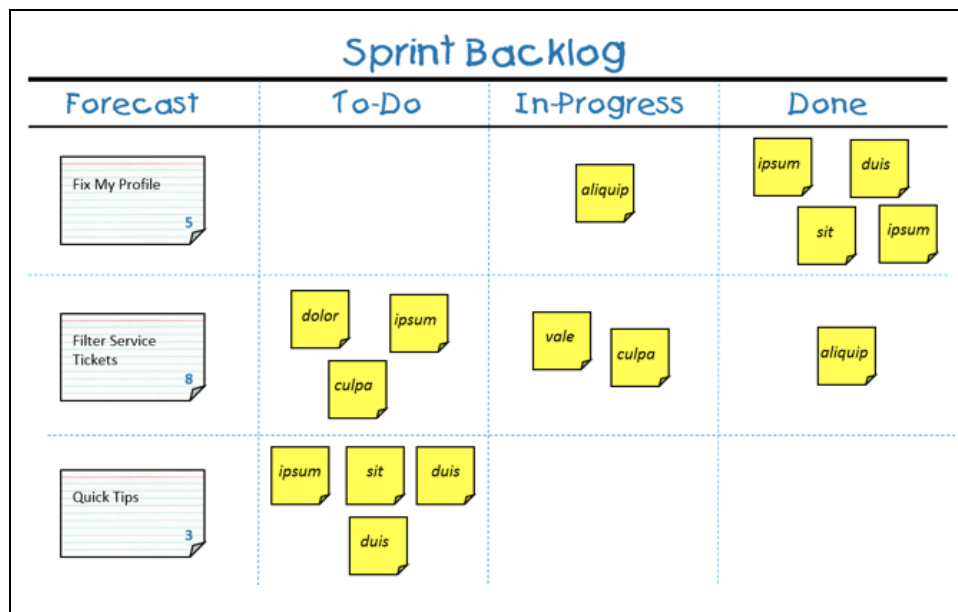


Figure 30 : Exemple d'un Sprint Backlog.

- **Incrément**

C'est le livrable de chaque Sprint, il s'agit d'un fragment de la solution potentiellement livrable. A la fin d'un sprint l'équipe a réalisé un produit partiel contenant de nouvelles stories et le produit est potentiellement livrable. A la fin de la release, le produit est vraiment livré. Scrum Master pousse à finir une story à la fin d'un sprint selon la définition de fini pour chaque story (US).

Le suivi de la progression se fait à tous les niveaux ; Sprint, Release et Projet. Le suivi de la release peut se faire selon les diagrammes :

- Burn-Down Chart
- Burn-Down Bar Chart
- Diagramme de vélocité
- Escaped Defect Chart

Par contre le suivi du Sprint se fait par :

- Burn-Down Chart
- Tableau du Sprint (Sprint Task Board)
- Niko-Niko Diagramme

V.5.4. Proposition d'un méta-modèle pour SCRUM

En analysant les concepts décrits dans SCRUM, nous remarquons que les différents rôles sont liés avec les différents livrables avec des relations soit de type produire, contrôler ou bien gérer, et que les événements sont aussi liés avec les rôles avec des relations de type planifier, revoir, ou bien gérer.

Ainsi, ci-dessous une proposition de méta-modèle SCRUM :

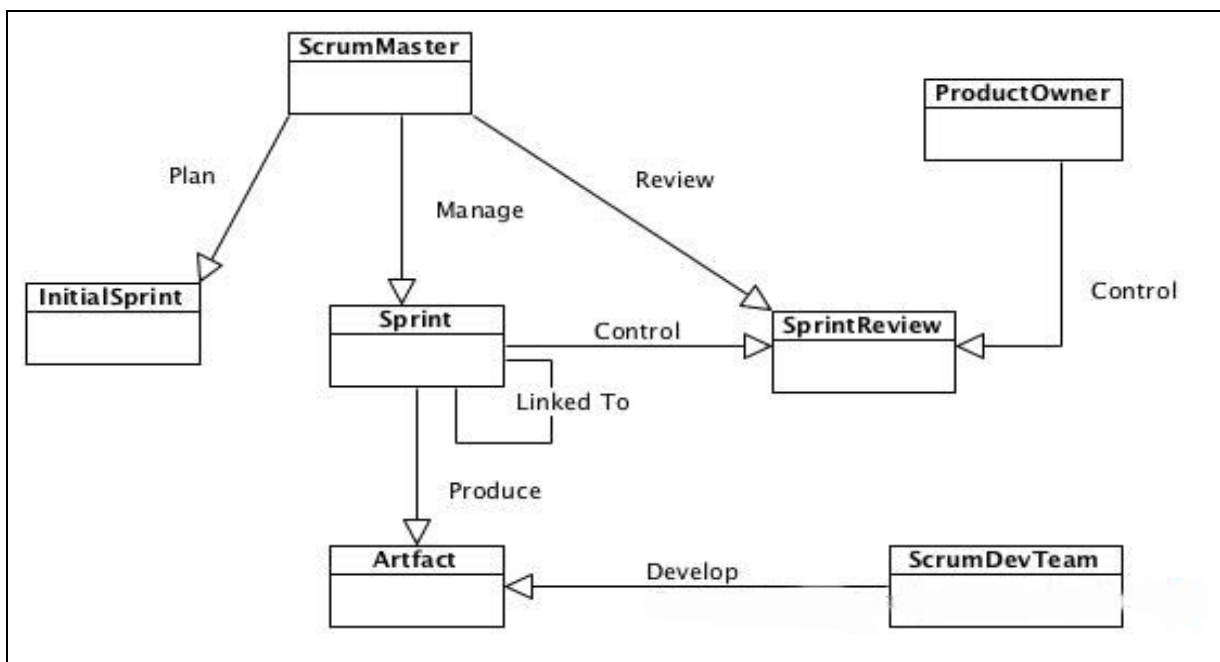


Figure 31 : Méta-modèle SCRUM.

Le tableau ci-dessous redéfinit les composants du méta-modèle :

Entité	Type	Description
ScrumMaster	Meta-Class	Le Scrum Master est le garant de la méthodologie
ProductOwner	Meta-Class	Le rôle du Product Owner

InitialSprint	Meta-Class	Il s'agit du premier sprint
Sprint	Meta-Class	Il s'agit des cycle du projet scrum
SprintReview	Meta-Class	La revue du sprint
Artifact	Meta-Class	Les produits de chaque sprint
ScrumDevTeam	Meta-Class	L'équipe de développement SCRUM

Tableau 22 : Structure de la méthode SCRUM.

V.6. Proposition d'un méta-modèle générique pour les méthodes Agiles

En se basant sur les concepts évoqués plus haut, nous avons proposé ce méta-modèle qui résume les relations entre les différents concepts des méthodes agiles :

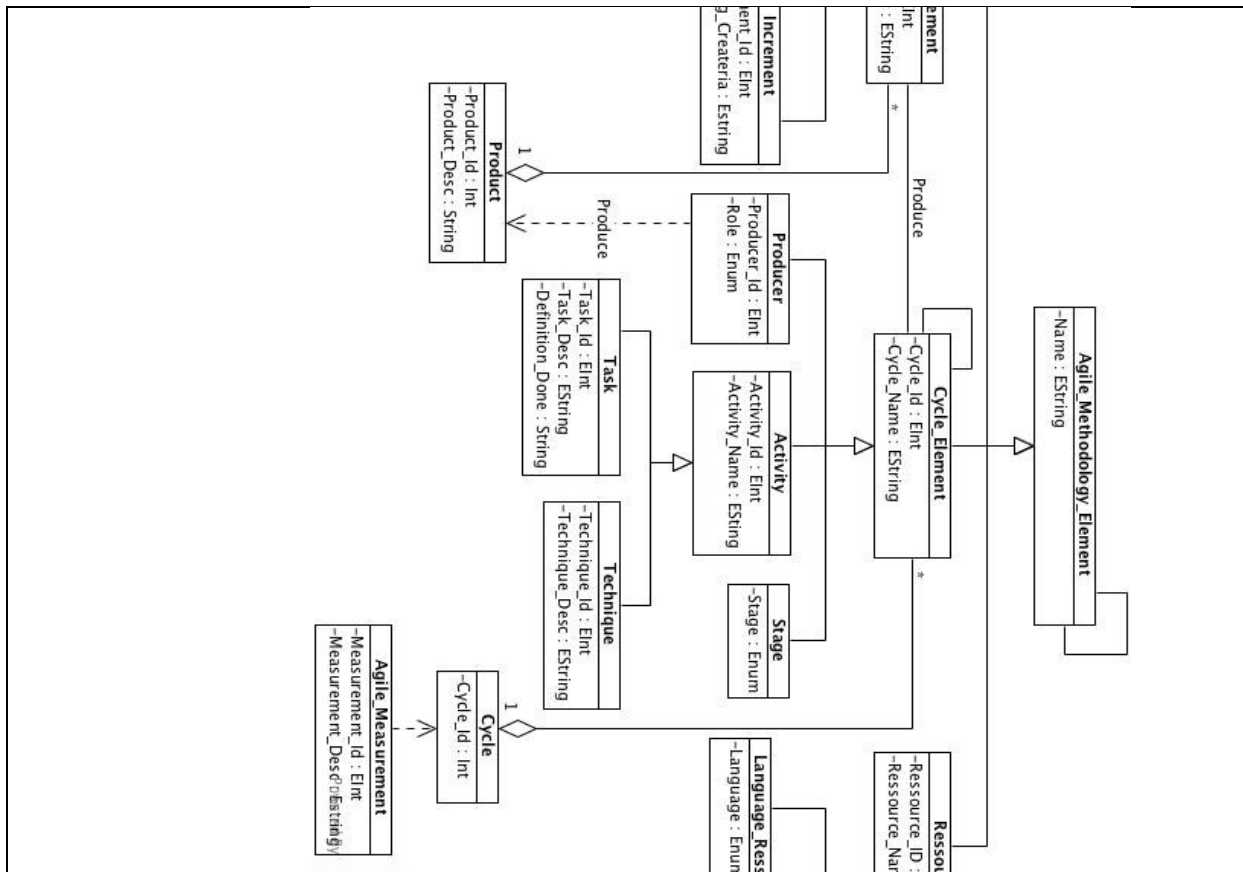


Figure 32 : Méta-modèle des méthodes Agiles.

Le tableau ci-dessous redéfinit les composants du PIM multidimensionnel (dimension Produit, Cycle et Ressource) :

Classe	Description	Instances possibles
--------	-------------	---------------------

1- Cycle Element	Éléments de méthodologie transformant les entrées en sorties. La relation entre un ou plusieurs processus éléments génère le processus.	Pratiques de gestion, étapes du Performance du processus...
1.1- Activity	Un élément de processus abstrait de base qui modélise une opération fonctionnellement cohérente qui peut être effectuée par un ou plusieurs producteurs	Pratique (programmation par paires, Refactoring ...), Activité, Tâche, Technique ...
1.1.1- Task	La plus petite unité de travail gérable.	Rédiger des tests unitaires, des cas d'utilisation de modèles, Ecrire le code ...
1.1.2- Technique	Des moyens spécifiques pour atteindre un objectif.	Entretien, réflexion ateliers ...
1.1.3- Measurement	Des moyens spécifiques pour effectuer des mesures sur les processus	Burn Down Chart, Diagramme de vitesse ...
1.2- Producer	Quelqu'un ou quelque chose qui exécute une unité de travail soit directement ou indirectement (c.-à-d. crée, évalue, itère ou maintient).	Scrum master, équipe, testeur, organisation, outil
1.3- Stage	Représente des intervalles de temps gérés ou des points temps, au sein d'un projet, c'est-à-dire l'aspect temporel du processus.	Itération, sprint, phase, cycle ...
2- Product Element	Tout artefact développé et produit pendant le projet. Cet élément équivaut au "Travail Élément produit".	Version, modèles UML, technique Documents ...
2.1- Activity	Un livrable est un élément de produit qui fournit une description et définition de l'emballage d'autres éléments de produits, et peuvent être livrés à un ou partie externe.	Libération, rapport technique ...
2.2- Artefact	Une ressource identifiable fournie par l'équipe de développement qui prend en charge le processus global de développement logiciel.	Document d'architecture logicielle, Modèle de cas d'utilisation ...
2.3- Outcome	Le résultat est un élément de produit qui fournit une description et une définition des éléments de produit non tangibles. Une	Pratique, discipline, domaine, plan ...

	différence clé entre les résultats et artefacts est que les résultats ne sont pas candidats pour récolter comme actifs réutilisables.	
3- Ressource	Éléments méthodologiques utilisés dans la réalisation du projet.	Rapports standards, documentation d'expertise ...
3.1- Langage	Vocabulaire (c'est-à-dire un ensemble de termes) combiné avec un ensemble de règles grammaticales utilisées pour produire un ou plusieurs éléments de produits.	Langage de modélisation, Langage d'implémentation, base de données Langue ...
3.1- Guideline	Règles et directives sur l'utilisation appropriée de un élément méthodologique donné.	Norme, directives méthodologiques ...

Tableau 23 : Structure du méta-modèle des méthodologies Agile.

V.7. Conclusion

Dans ce chapitre, nous avons présenté les différents concepts des méthodologies Agile, puis nous avons cité comme exemple la méthode SCRUM, et nous avons proposé un méta-modèle générique de la méthodologie qui relie ses différents concepts, le but est d'appliquer l'approche proposée pour générer un méta-modèle unique de la gestion de projets.

Dans la section suivante, nous présentons l'implémentation de notre approche, en définissant un cadre de référence et analysant les concepts remontés selon plusieurs méthodes.

Chapitre VI : Validation de l'approche proposée

VI.1. Introduction

Dans ce chapitre, nous mettons en avant une étude de cas d'un projet de compagnie Société Générale qui avait été traité selon la méthode PRINCE 2. Le projet souffre d'un effet tunnel de 4 mois après quoi le management se rend compte qu'il n'y a pas de livrable, la problématique consiste à changer de méthodologie en cours de route, à travers notre approche nous allons expliquer comment ça pouvait être traité en SCRUM.

Dans un premier temps, nous allons procéder à la transformation en se basant sur le langage ATL [90] pour transformer le projet qui se gère en prédictive vers de la gestion Agile.

VI.1. Transformation des modèles prédictives en modèles agiles

Une transformation de modèle est le processus de conversion d'un modèle source en modèle cible selon un ensemble de règles de transformation. Les règles sont définies avec un langage de transformation de modèle [91]. Une règle de transformation se compose de deux parties: un côté gauche qui accède au modèle source; et un côté droit qui se développe sur le modèle cible. Le modèle source est conforme au méta-modèle source et le modèle cible est conforme au méta-modèle cible. Pendant la transformation, le modèle source reste inchangé. La figure ci-dessous montre un modèle de transformation de modèle adapté à l'IDM [71].

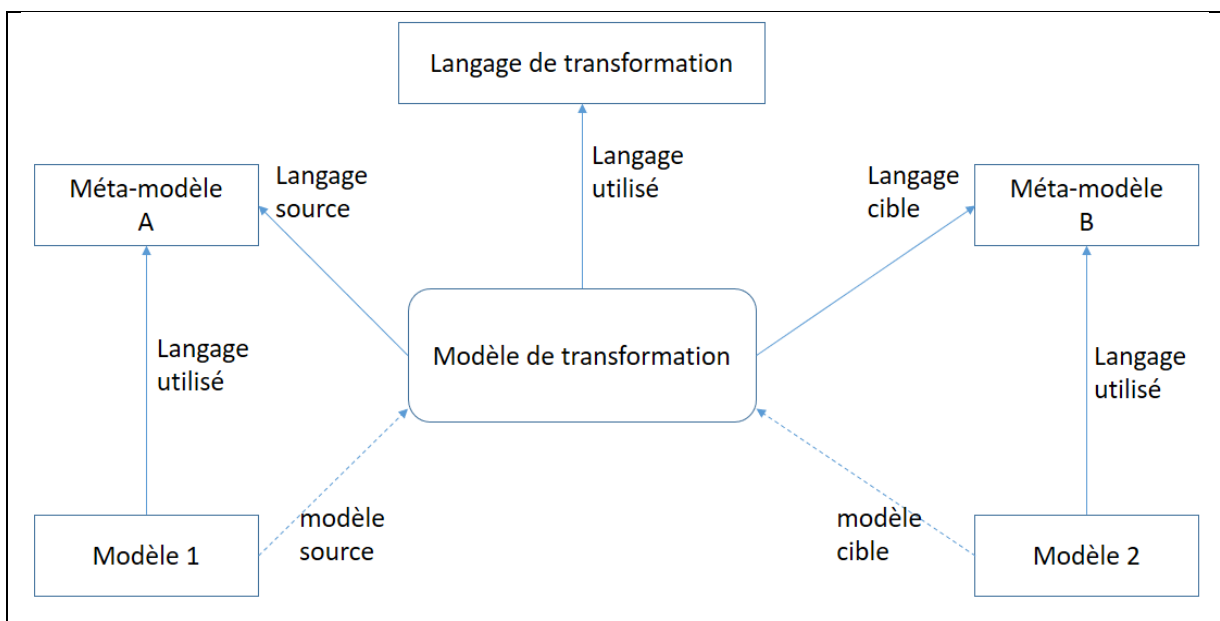


Figure 33 : Modèle de transformation de modèle dans l'ingénierie des modèles

Si les méta-modèles source et cible sont identiques, la transformation est dite endogène; sinon, il est appelé exogène [91]. Si le niveau d'abstraction ne change pas, la transformation est appelée transformation horizontale. Si le niveau d'abstraction change, la transformation est appelée transformation verticale. Selon le type et la complexité du processus, la transformation de modèle peut être nommée comme transformation de modèle en modèle (M2M), transformation de modèle en texte (M2T), fusion de modèle, liaison de modèle, synthèse de modèle, mappage de modèle ou transformation de modèle en code (génération de code).

Dans une approche MDD, il existe généralement une chaîne de plusieurs transformations M2M et une transformation finale de modèle en code. Les transformations

M2M sont utilisées avec les différents modèles intermédiaires, nécessaires pour exprimer différentes vues d'un système, à des fins d'optimisation, de réglage ou de débogage [92].

Comme son nom l'indique, une transformation M2M convertit un modèle source en un modèle cible, qui peut être des instances de méta-modèles identiques ou différents. Cependant, une transformation M2T convertit un modèle source en un fichier texte. La transformation M2T est généralement utilisée pour la génération de code final ou la création de documents de support. Bien qu'il existe différentes solutions pour transformer des modèles en texte, ce n'est pas le cas pour transformer des modèles en modèles [92]. La transformation M2M étant un domaine relativement jeune, des capacités de transformation très limitées et souvent ad hoc sont offertes par les environnements de modélisation disponibles sur le marché.

Les langages de transformation M2M bien connus sont ATL (ATLAS Transformation Language) et QVT (Query / View / Transformation). La spécification MOF 2.0 QVT est un ensemble de langages de transformation de modèle définis par l'OMG (OMG 2011). La spécification QVT a une nature déclarative / impérative hybride et définit trois langages de transformation associés: les relations, les mappages opérationnels et le noyau. Le méta-modèle et le langage des relations prennent en charge la correspondance de modèles d'objets complexes et la création de modèles d'objet avec une approche conviviale de haut niveau. Les mappages opérationnels peuvent être utilisés pour implémenter une ou plusieurs relations à partir d'une spécification de relations, lorsqu'il est difficile de fournir une spécification purement déclarative de la façon dont une relation doit être remplie. Le méta-modèle et le langage de base sont définis à l'aide d'extensions minimales d'EMOF et de langage OCL (Object Constraint Language) au niveau inférieur.

ATL est l'un des modèles les plus populaires pour modéliser les langages de transformation. Une fois que le méta-modèle cible et le méta-modèle source sont disponibles, un fichier de transformation ATL peut produire un modèle cible à partir d'un modèle source. Pendant la transformation, un attribut d'une instance de modèle cible reçoit la valeur de retour d'une expression OCL basée sur une instance de modèle source. L'environnement de développement intégré ATL (IDE), qui est développé au-dessus de la plate-forme Eclipse, fournit un certain nombre d'outils tels qu'un éditeur de texte avec coloration syntaxique, un débogueur et un interpréteur qui vise à faciliter le développement des transformations ATL.

Il existe de nombreux autres langages de transformation de modèles [93]. De plus, certains projets comme Kermeta, Fujaba et GReAT (ces projets appliquent une approche de développement axée sur les modèles) fournissent des méthodes de transformation M2M. Des explications détaillées sur les langages et méthodes de transformation des modèles sont données dans [94] [92].

À ce stade, il est important de noter que la possibilité de définir différentes transformations de modèle pour un même modèle source. Ceci est illustré dans la figure 4 en étendant l'exemple d'ATL 'Families to Persons' donné dans [95]. Le modèle Family est conforme au méta-modèle Family tandis que le modèle Person est conforme au méta-modèle Person. Families2Persons.atl est une illustration simple de la transformation M2M. Si nous définissons un autre fichier de transformation de modèle comme Families2Mothers.atl, il peut produire un modèle cible différent à partir du même modèle source. De plus, il est également possible de changer le méta-modèle cible, et donc les transformations du modèle. Cependant, la définition de méta-modèles précis et le choix de la meilleure transformation de modèle sont les activités difficiles de MDD.

VI.2. Exécution de la transformation

Jusqu'à présent, nous avons défini les méta-modèles de méthodologie prédictive et de méthodologie Agile. Dans cette section, nous présentons les règles de transformation utilisées pour passer des méta-modèles génériques de méthodologie prédictive à la méthodologie Agile.

Pour appliquer toutes les transformations, nous avons choisi le langage de transformation ATL [96] [97]. Nous présentons maintenant des extraits du code ATL que nous avons utilisé pour transformer les méta-modèles proposés pour la méthodologie Prédictive en méta-modèle proposé pour la méthodologie Agile. Ces méta-modèles définis présentent le niveau PIM (Platform Independent Model) selon l'architecture dirigée par les modèles [98] [99].

```
rule Predictive_Methodology_Element2Agile_Methodology_Element{  
  from  
  s: Predictive_Methodology_Element  
  to  
  t: Agile_Methodology_Element(  
    name<- s.name
```

```

    )
}
rule Processus_Element2Cycle_Element {
    from
    s: Processus_Element
    to
    t: Cycle_Element(
        Cycle_id<- s.Processus_id,
        Cycle_name<- s.Processus_name
    )
}
rule Product_Element2Product_Element {
    from
    s: Product_Element
    to
    t: Product_Element(
        Product_id<- s.Product_id,
        Product_Desc <- s.Product_Desc
    )
}
rule Ressource2Ressource {
    from
    s: Ressource
    to
    t: Ressource (
        Ressource_id<- s.Ressource_id,
        Ressource_Name <- s.Ressource_Name
    )
}
rule Language_Ress2Language_Ress {
    from
    s: Language_Ress
    to
    t: Language_Ress (
        Language<- s.Language
    )
}
rule Guide2Guide {
    from
    s: Guide
    to
    t: Guide (
        Guide_id<- s.Guide_id,
        Guide_Name <- s.Guide_Name,
        Guide_Version <- s.Guide_Version,
        Date <- s.Date
    )
}
rule Producer2Producer {
    from
    s: Producer
    to

```



```

t: Producer (
    Producer_Id<- s.Producer_id,
    Role <- s.Role
)
}
rule Activity2Activity {
    from
    s: Activity
    to
    t: Activity (
        Activity_Id <- s.Activity_Id,
        Activity_Name <- s.Activity_Name
    )
}
rule Stage2Stage {
    from
    s: Stage
    to
    t: Stage (
        Stage <- s.Stage
    )
}
rule Task2Task {
    from
    s: Task
    to
    t: Task (
        Task_id<- s.Task_id,
        Task_Desc <- s.Task_Desc,
        Definition_Done <- s.Definition_Done
    )
}
rule Technique2Technique {
    from
    s: Technique
    to
    t: Technique (
        Technique_id<- s.Technique_id,
        Technique_Desc <- s.Technique_Desc,
    )
}
rule Documents2Artifact {
    from
    s: Documents
    to
    t: Artifact (
        Artifact_id<- s.Doc_id,
        Artifact_Desc <- s.Doc_Desc
    )
}
rule Outcome2Increment {
    from

```

```

s: Outcome
  to
t: Increment (
  Increment_id<- s.Outcome_id,
  Testing_Createria <- s.Testing_Createria
)
}
rule Product2Product {
  from
  s: Product
  to
  t: Product (
    Product_id<- s.Product_id,
    Product_Desc <- s.Product_Desc
  )
}
rule Process2Cycle {
  from
  s: Process
  to
  t: Cycle (
    Cycle_id<- s.Process_id,
  )
}
rule Mesurement2Agile_Mesurement {
  from
  s: Mesurement
  to
  t: Agile_Mesurement (
    Mesurement_id<- s.Mesurement_id,
    Mesurement_Desc <- s.Mesurement_Desc
  )
}

```

Tableau 24 : Aperçu du code ATL de la transformation.

VI.3. Test et validation de la transformation

Dans cette section, nous allons présenter une expérimentation sur notre transformation, cette transformation qui prend comme input le Predictive methodology metamodel vers un output qui est un modèle d'Agile methodology metamodel, la figure suivante représente le résultat de temps d'exécution de transformation d'un projet IT géré par la méthodologie prédictive, qui va être transformé en une gestion par la méthodologie agile. Cette transformation a été réalisée par le langage de transformation ATL [100] [101]. Pour réaliser cette transformation nous avons utilisé trois instances de projets, un petit projet appelé Project_1 (de l'ordre d'un mois), un moyen projet nommé Project_2 (de l'ordre deux cents

jours hommes), et un grand projet qui est Project_3 (de l'ordre trois milles jours hommes). Les résultats de cette transformation sont présentés dans le tableau suivant :

	Project 1	Project 2	Project 3
Transformation time (s)	323	358	391
Durée (w.d.)	30	200	3000

Tableau 25 : Temps de transformation en utilisant trois types de projets

La figure suivante illustre le résultat de temps de transformation en(s) en utilisant les trois projets : **Project_1, Project_2, Project_3** :

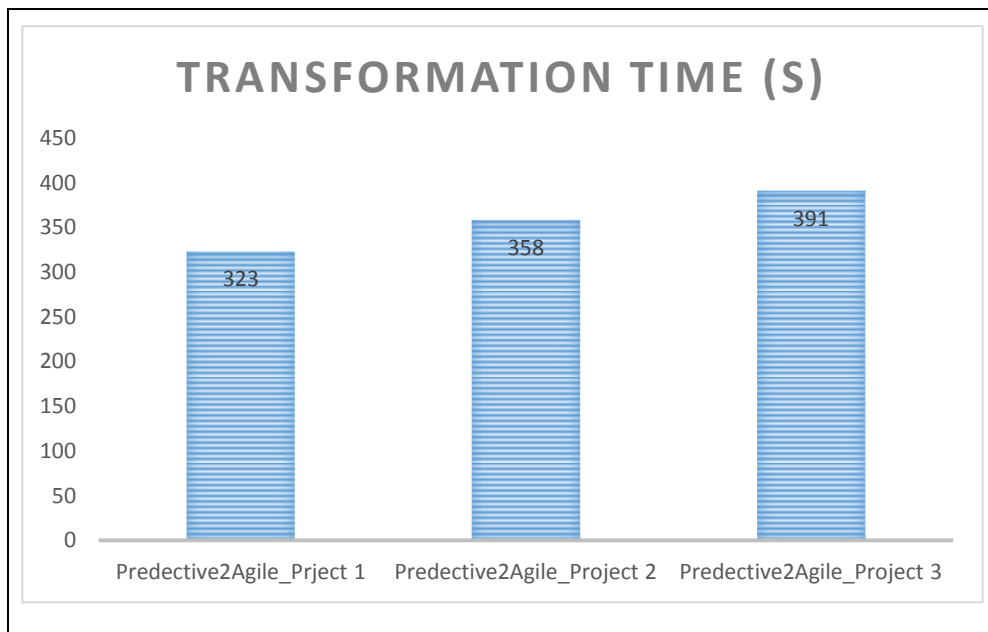


Figure 34 : Graphe de transformation en utilisant trois types de projets

VI.4. Discussion des résultats

Les méthodes agiles les plus populaires sont RAD, XP et Scrum se basent sur plusieurs pratiques d'assistance à la communication et à la coordination. La méthode RAD préconise la mise en place du rôle de facilitateur ayant pour mission de prendre des décisions en tant que « représentant du client » ainsi que de trouver des solutions aux problèmes de l'équipe de conception. Elle introduit également la pratique du codage en binôme. XP mise sur l'intelligence collective en réutilisant la pratique de facilitateur, tout en favorisant l'émergence d'une cohésion de groupe. Elle met également en place des sessions de planning poker pour estimer la durée des tâches à réaliser. Enfin, Scrum instaure un cycle de développement régulier dans lequel s'inscrivent des réunions quotidiennes. Ces réunions

permettent aux acteurs de la collaboration d'échanger sur leur avancement et sur leurs besoins. Scrum peut également s'appuyer sur des sessions de planning poker [102] [103].

Nous constatons que ces trois méthodes agiles possèdent des caractéristiques communes et misent sur des pratiques facilement applicables dans l'une ou l'autre méthode, les rendant malléables. Les pratiques utilisées dans la plupart des méthodes agiles sont compatibles entre elles. [104] Propose l'utilisation conjointe de Scrum et XP, en mélangeant les pratiques telles que le planning poker, le code en binôme, ou encore le rôle de facilitateur.

VI.5. Etude de cas

VI.5.1. Projet « Solution cible Tax Authority »

Les deux réglementations FATCA -Foreign Account Tax Compliance Act- et CRS – Commun Reporting Standard- sont destinées à lutter contre l'évasion fiscale internationale et à renforcer la transparence, l'entreprise s'est engagé à être conforme aux exigences requises par les réglementations FATCA et CRS. Tous les entités financières ont la responsabilité d'appliquer les obligations réglementaires introduites par FATCA et CRS (si pays de l'entité engagé à CRS) en matière de :

Identification de l'ensemble des clients PP et PM au titre de FATCA (qualification) et CRS (qualification et pays de résidence fiscale) lors de l'entrée en relation et durant la relation d'affaire.

Contrôle : Existence d'un programme de conformité (procédures, contrôles opérationnels LOD1, ...) pour s'assurer de la bonne application des obligations FATCA et CRS.

Reporting annuel des comptes des clients déclarables aux administrations fiscales concernées, au titre de FATCA et au titre de CRS.

VI.5.1.1.1. Vision du produit, objectifs et périmètre

L'objectif est d'avoir un outil (en full ou en modules) qui couvrirait les 4 étapes pour les deux réglementations en limitant les risques opérationnels, en optimisant les délais et charges du processus actuel tout en garantissant la traçabilité et piste d'audit et la production d'indicateurs de suivi. La valeur ajoutée de cette solution réside aussi dans sa capacité à contribuer à l'amélioration de l'expérience Client au sein de l'agence et fiabiliser ses données KYC. Ainsi, le produit a pour objectif de servir les besoins qui suivent :

- Identification des Clients à l'entrée en relation et lors des changements de circonstances en mode online et gestion des statuts clients
- Etablissement d'un contrôle opérationnel périodique (LOD 2)
- Génération des Reporting règlementaires.

Il existe aujourd'hui, en réponse à la réglementation FATCA, une solution de contournement basée sur 4 étapes :

- L'identification à l'EER et pour le changement de circonstance se fait selon une procédure manuelle (Echanges entre Front office et RCO)
- La gestion des documents se fait au niveau du RCO (scan au niveau de l'agence et envoi par mail ou par workflow KYC)
- Le contrôle opérationnel périodique se fait à l'aide d'extractions régulières (quotidienne, mensuelle, trimestrielle). Ces extractions sont traitées au niveau d'un outil dédié (Moulinette FATCA)
- La gestion des statuts se fait dans le CBS. Le Front Office change manuellement les statuts. L'historique des statuts n'est pas répertorié.

La génération et envoi des reportinf se fait en 4 temps :

- Préparation du fichier sous excel par les RCO
- Conversion du fichier en XML via l'applicatif FATCA Light
- Cryptage du fichier XML via l'outil IDES Tool
- Envoi du fichier crypté par la cellule FATCA/CRS

Pour la Réglementation CRS, les entités concernées au niveau AFMO répondent au besoin comme suit :

- L'identification à l'EER et pour le changement de circonstance se fait selon une procédure manuelle (Echanges entre Front office et RCO)
- La gestion des documents se fait au niveau du RCO (scan au niveau de l'agence et envoi par mail ou par workflow KYC)
- Il n'existe pas de contrôle périodique CRS

- La gestion des statuts se fait dans le CBS. Le Front Office change manuellement les statuts. L'historique des statuts n'est pas répertorié.

La génération et envoi des reporting se fait en 4 temps :

- Extraction des fichier XML du CBS
- Modification des données par les RCO au niveau de l'outil FITAX
- Formatage du fichier XML via l'outil FITAX
- Cryptage du fichier XML
- Envoi du fichier crypté

VI.5.1.1.2. Organisation et lots du projet

Ci-dessous l'organisation du projet :

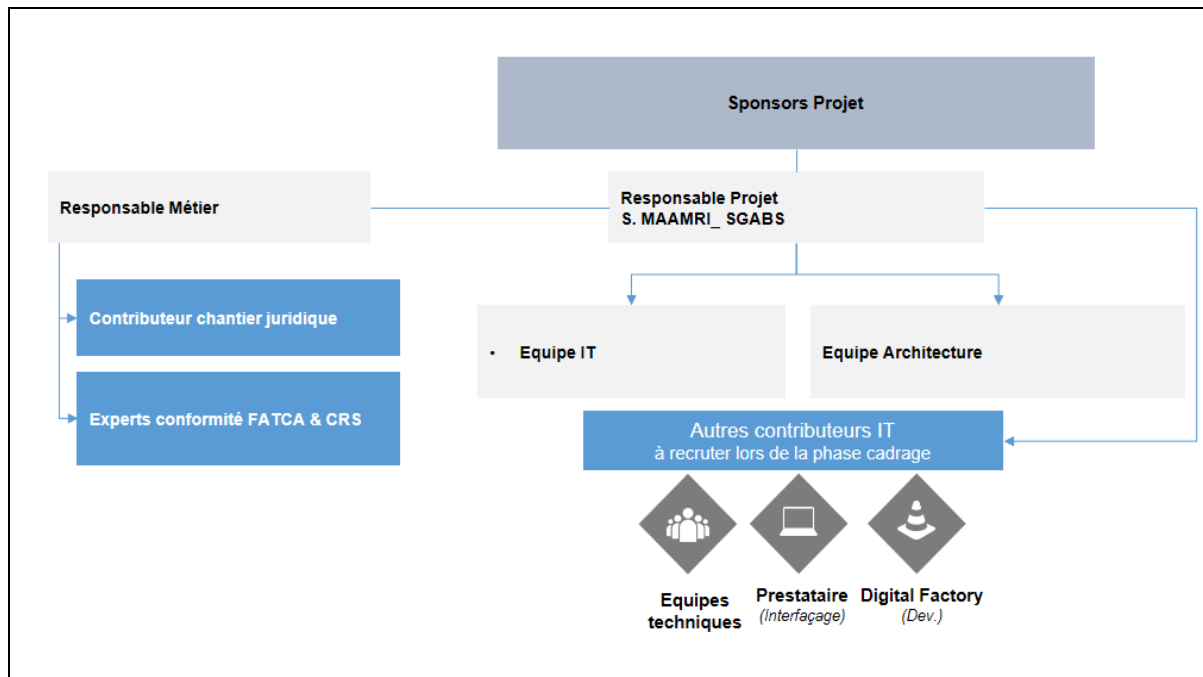


Figure 35 : Organisation projet de la Solution cible Tax Authority

Ci-dessous les lots du projet :

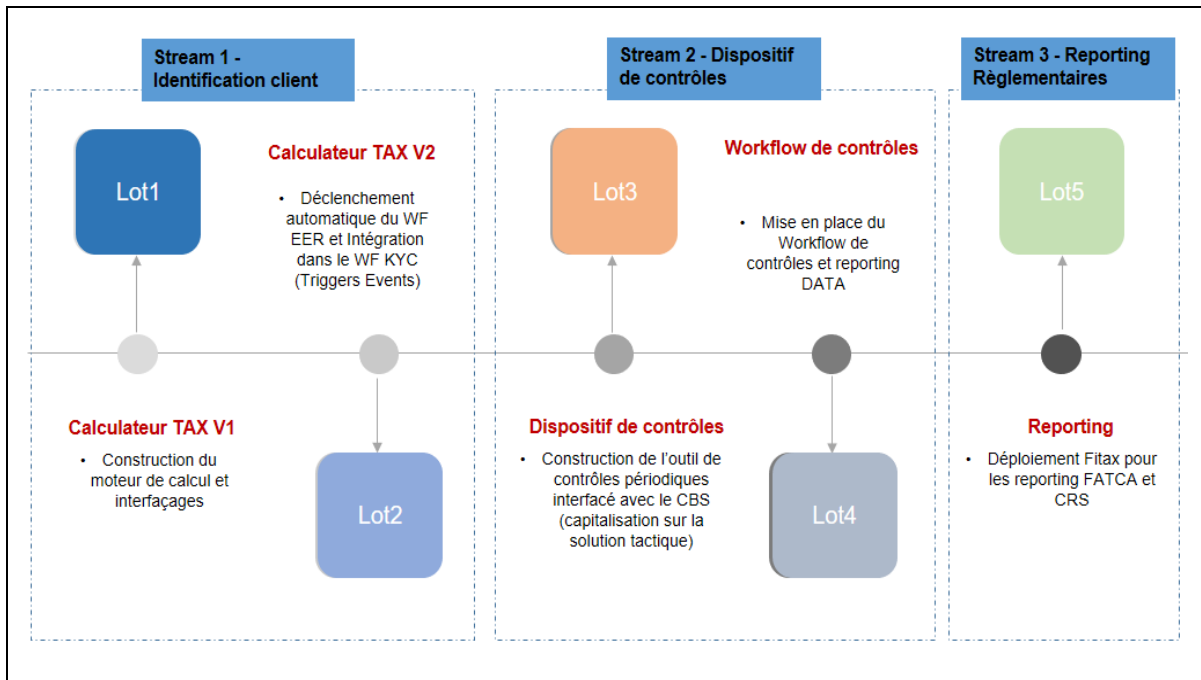


Figure 36 : Planning du projet Solution cible Tax Authority

Comitologie du projet :

Instances	Participants	Objectifs
Comité de pilotage projet	Sponsors Resp. projet métier Chef de projet	Contrôle du déroulement du projet et du suivi des objectifs, dans le respect des orientations stratégiques Fréquence : Bimestriel
Comité projet opérationnel	Resp. projet métier Chef de projet Autres membres de l'équipe projet	Pilotage hebdomadaire des actions entreprises / à entreprendre; décisions opérationnelles Fréquence : Hebdomadaire

Tableau 26 : Comitologie du projet Solution cible Tax Authority

VI.5.2. Transformation de la gestion du projet Tax Authority en agile

En appliquant la transformation que nous avons créé au début de ce chapitre, nous avons pu agréger le projet en trois chantiers seulement que nous avons parallélisé en plusieurs sprints, voir la figure plus bas :

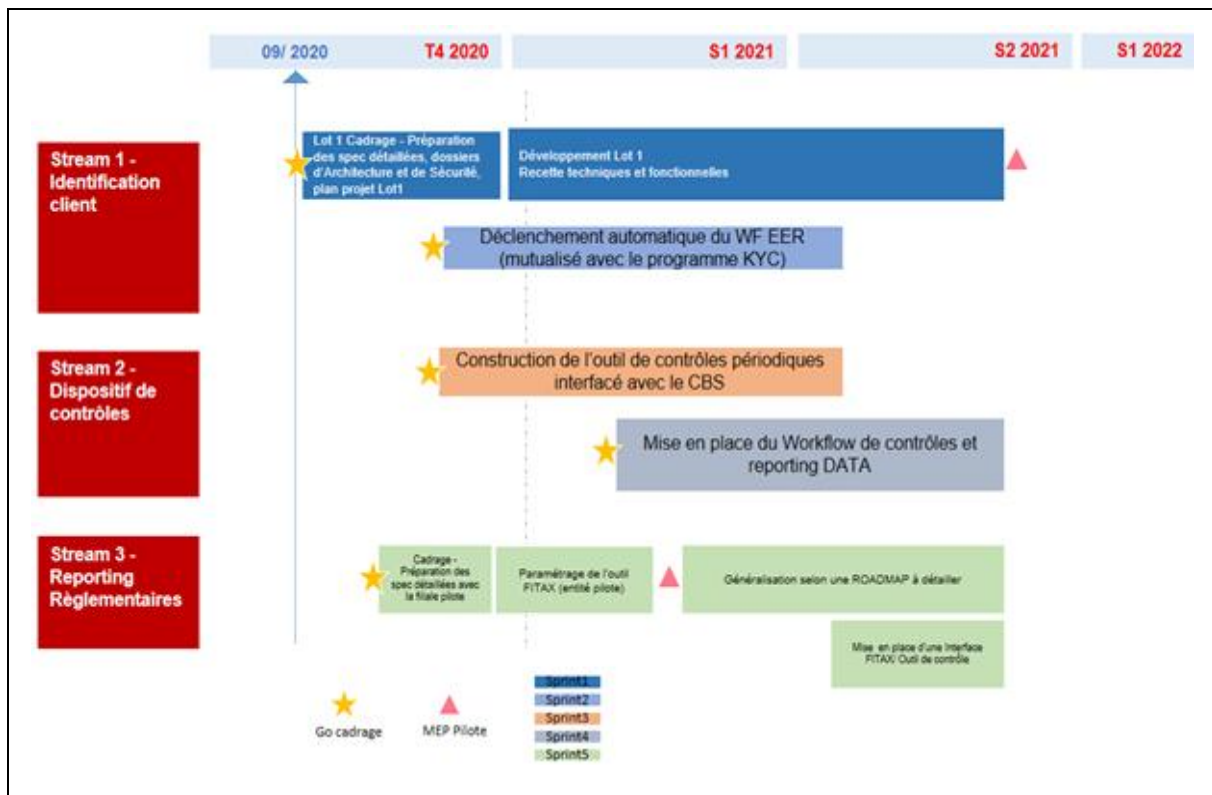


Figure 37 : Nouveau planning du projet Solution cible Tax Authority

En transformant le projet en agile nous avons gagné en efficacité en terme de planification, l'utilisateur va consommer les livrables de chaque sprint et apprécier les livrables du projet. Néanmoins, il nous reste un facteur essentiel dans le bon déroulement de notre projet, il s'agit de la conduite de changement des équipes développement et les fonctionnelles afin de garantir le succès de notre projet.

VI.5.2.1.1. Backlog projet

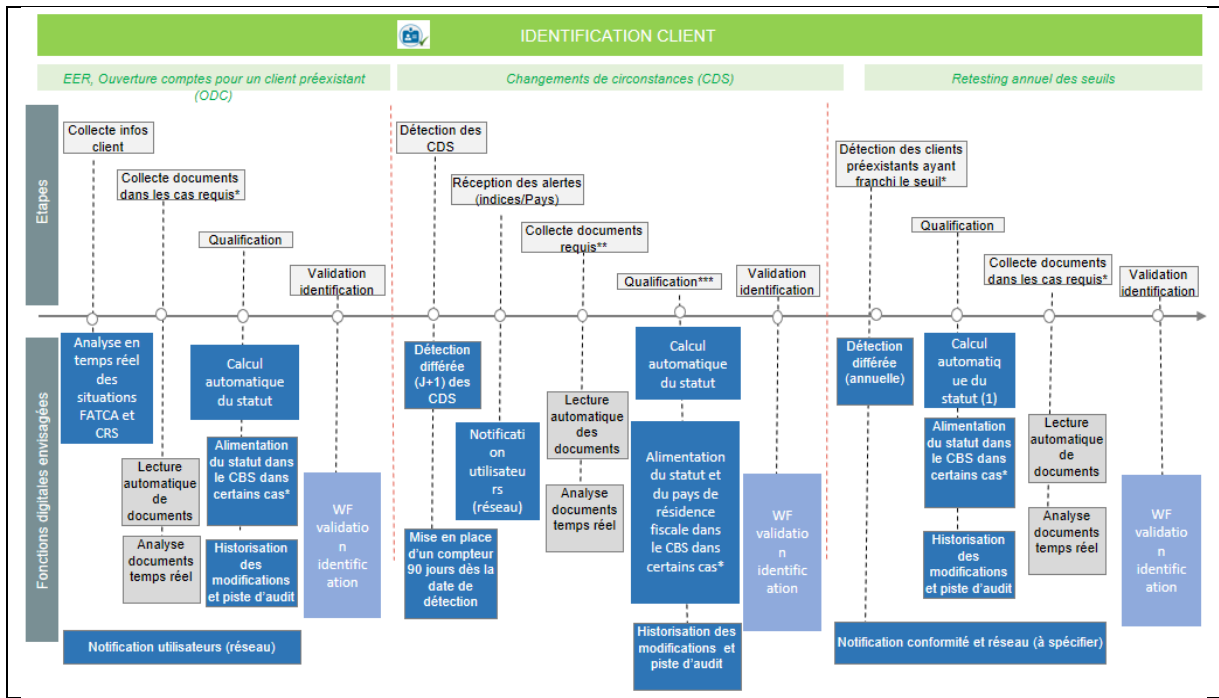


Figure 38 : Backlog restitué du projet Tax Authority

VI.5.2.1.2. Nouvelle organisation du projet

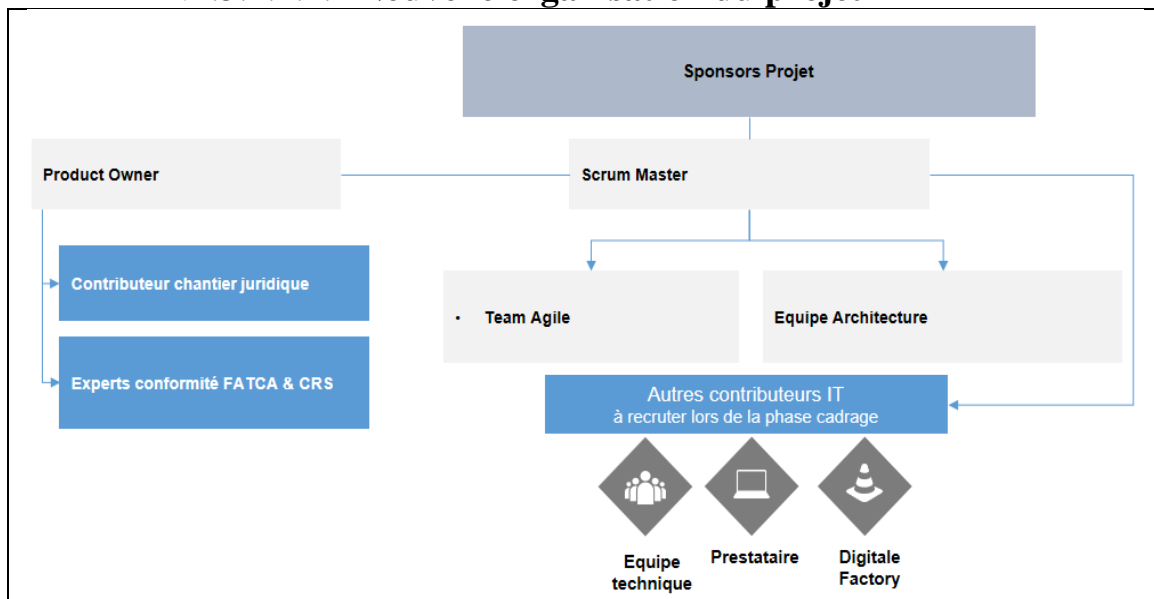


Figure 39 : Nouvelle organisation projet de la Solution cible Tax Authority

VI.6. Conclusion

Nous avons proposé deux méta-modèles de gestion de projets (Agile et Prédictive) [88] qui englobent les concepts de gestion de projets, puis nous avons créé une transformation qui

permet de déduire le modèle Agile à partir d'un modèle Prédicatif et nous avons appliqué le modèle de transformation sur un exemple concret de gestion de projets.

L'objectif global de cette contribution est d'aider les organisations dans leurs transformations digitales en utilisant les nouvelles technologies avec la manière la plus optimale, sans oublier que l'élément clé d'une transformation restera le volet humain qui nécessite un programme de gestion de changement au niveau de l'organisation

Conclusion générale

Ce travail de recherche a permis de contribuer dans les domaines de l'ingénierie dirigée par les modèles, la gouvernance IT et la gestion de projets en répondant aux objectifs suivants :

- i) Distinction entre les référentiels standards et les bonnes pratiques de la gouvernance IT et la classification de ses référentiels à travers une analyse multicritère **WSM**
- ii) Proposition d'une **application** aux professionnels qui permet de choisir entre les différents référentiels selon les exigences.
- iii) Proposition d'une **approche** de construction d'un méta-modèle fiable en utilisant la théorie des ensembles.
- iv) Proposition d'un **méta-modèle** global de **gestion de projet**.
- v) Proposition de deux **méta-modèles** générique des méthodes **agile SCRUM** et **PRINCE 2** en se basant sur les concepts de base des méthodologies.
- vi) Evaluation du méta-modèle de gestion de projets en se basant sur les techniques de transformation qu'offre l'approche IDM en spécifiant des cas d'études de gestion de projets depuis le prédictive à l'agile en évitant l'effet tunnel.
- vii) La transformation digitale d'un projet de la méthode PRINCE 2 vers la méthode SCRUM

Cependant, les axes de recherche futurs ouverts par notre humble contribution sont très nombreux ; Tout d'abord, nous sommes entrain d'appliquer notre approche sur la gestion des services informatiques, puis nous allons étaler nos travaux sur l'ensemble des composantes de la gouvernance IT en prenant en charge la transformation digitale appliquée à chaque composant. Ensuite l'agrégation des modèles remontés dans un unique modèle.

Nous ajoutons un autre axe important de recherche qui est la modélisation de la gouvernance d'entreprise et sa liaison avec la gouvernance IT, nous pouvons créer des leviers IT qui vont pouvoir aider les organisations à mieux réaliser leurs objectifs stratégiques.

Références

- [1] CIGREF, Les référentiels de la DSI, Cigref Edition, 2009.
- [2] T. N. C. Center, developing a successful governance startegy, 2005.
- [3] B. Ménard, CobiT : Pour une meilleure gouvernance des systèmes d'information, Eyrolles, 2009.
- [4] W. Van Grembergen, «Introduction to the minitrack IT governance and its mechanisms,» chez Proceedings of the 35th Hawaii International Conference on System Sciences, HICSS, 2002.
- [5] W. D. H. S. Van Grembergen, Measuring and improving information technology governance through the balanced scorecard, Inf. Syst. Control, 2005.
- [6] J. Duffy, IT governance and business value part 1, 27291: IDC document, 2002.
- [7] Helmut Schindlwick, IT Gouvernance : IT Governance: How to Reduce Costs and Improve Data Quality through the Implementation of IT, Dunod, 2017.
- [8] J. K. D. Callahan, «The evolution of IT Governance @ NB Power,» chez Van Grembergen W. (ed.) Strategies for Information Technology Governance, Idea Group Publishing, 2003.
- [9] R. H. Alexandre Steigmeier, Gouvernance de la sécurité : comment articuler les différentes normes et méthodes, 2009.
- [10] P. R. J. Weill, Don't Just Lead, Govern: Empowering Effective Enterprise Use of, Boston: Harvard Business School Press, 2004.
- [11] R. H. Alexandre Steigmeier, Gouvernance de la sécurité : comment articuler les différentes normes et méthodes, Novembre 2009.
- [12] J. S. Martine Otter, Guide des certifications SI, Dunod, 2009.
- [13] H. k. L. M. H. Askari, Reformation and Development in the Muslim World: Islamicity Indices as Benchmark, Springer, 2017.
- [14] R. Tricker, ISO 9001:2015 for Small Businesses, Edition, 2016.
- [15] R. Basque, CMMI 1.3 - Guide complet de CMMI-DEV et traduction de toutes les pratiques CMMI-ACQ et CMMI-SVC, Dunod, 2011.
- [16] M. M. S. Y. D. Jarke, «Conceptual Model-ing and Knowledge Based Suppor of Information Systems Development Process,» chez Technique et Science Informatiques, 1990, pp. 122-133.
- [17] Z. I. Batouta, «Multi-criteria analysis and advanced comparative study between automatic generation approaches in software engineering,» 2015, p. Vol.81. No.3.
- [18] O. o. Commerce, ITIL 2011 Amélioration continue des services, stationery office, 2011.
- [19] O. o. Commerce, ITIL 2011 Conception des services, stationery office, 2011.
- [20] O. o. Commerce, ITIL 2011 Exploitation des services, stationery office, 2011.
- [21] O. o. Commerce, ITIL 2011 Stratégie des services, stationery office, 2011.
- [22] O. o. Commerce, ITIL 2011 Transition des services, stationery office, 2011.
- [23] S. S.-M.-L. Abbadi, «Proposition de méthode d'implémentation d'ITIL,» 2eme édition

- du congrès International de génie Industriel et Management des systèmes, pp. 5-6, 21
Mai 2015.
- [24] F. Georget, *IT Gouvernance : Maîtrise d'un système d'information*, Dunod, Mai 2015.
 - [25] Axelos, *Managing Successful Projects with PRINCE2*, AXELOS, 2015.
 - [26] J. S. Ken Schwaber, *The Definitive Guide to Scrum: The Roles of the Game*, www.scrum.org, 2017.
 - [27] A. & H. M. Davies, *The business of projects: managing innovation in complex products and systems*, Cambridge: Cambridge University Press, 2015.
 - [28] A. J. & D. D. Shenhar, *Reinventing project management: The diamond approach to successful growth and innovation*, Cambridge: Harvard Business School Press, 2007.
 - [29] S. L. A. & L. D. Gunnarson, «Is standardisation applicable to project managers of multi-project companies? In paradoxes of project collaboration in the global economy: Interdependences, complexity and ambiguity,» chez *Proceedings of the IRNOP IV Conference*, 2000, p. 136–146.
 - [30] J. Charvat, *Project management methodologies: Selecting, implementing and supporting methodologies and processes for projects*, Hoboken: NJ: Wiley, 2003.
 - [31] P. W. & P. J. K. Morris, *The Wiley guide to managing projects*, Hoboken: NJ: Wiley, 2004.
 - [32] C. & R. B. H. Sauer, «Rethinking IT project management: Evidence of a new mindset and its implications,» *International Journal of Project Management*, vol. 27, n° 12, p. 182–193, 2009.
 - [33] H. J. Thamhain, «A manager's guide to effective concurrent project management,» chez *PM Network*, 1994, pp. 8(11), 6–10.
 - [34] D. & F. J. White, «Current practice in project management: An empirical study,» *International Journal of Project Management*, vol. 20, n° 11, pp. 1-11, 2002.
 - [35] Z. Al-Zoabi, «Introducing discipline to XP: Applying PRINCE2 on XP projects,» chez *Third International Conference on Information and Communication Technologies, Syria, From Theory to Applications*, The Arab International University, 2008.
 - [36] A. J. & D. D. Shenhar, *Reinventing project management: The diamond approach to successful growth and innovation*, Cambridge: Harvard Business School Press, 2007.
 - [37] M. Cross, *Public sector IT failures*, 2005.
 - [38] L. Scott, *NHS employers: "We must attract, retain and develop talent"*, 2009.
 - [39] J. Oates, *Failed probation system "masterclass in sloppy management."*, *The Register*, 2009.
 - [40] R. G. Cooper, «Managing technology development projects,» chez *Engineering Management Review*, IEEE, 2007, p. 67–76.
 - [41] N. Jayaratna, *Understanding and evaluating methodologies*, London: McGraw-Hill, 1994.
 - [42] Office of Government Commerce (OGC), *Managing successful projects with PRINCE2*, London: OGC, 2009.
 - [43] J. & W.-H. T. McManus, *The IT trade off., Project*, 2008.
 - [44] J. H. R. & K. H. K. Iivari, «A dynamic framework for classifying information systems development methodologies and approaches,» *Journal of Management Information Systems*, vol. 17, n° 13, p. 179–218, 2000.

- [45] L. D. & W. E. A. Introna, «Against method–ism: Exploring the limits of method. Information Technology & People,» vol. 10, n° %11, pp. 31-45.
- [46] E. S. Andersen, «Perspectives on projects,» chez Proceedings of the PMI Research Conference 2006, Canada, 2006.
- [47] B. Boehm, «Get ready for agile methods, with care.,» Computer, vol. 35, n° %11, p. 64–69, 2002.
- [48] B. & T. R. Boehm, Balancing agility and discipline: A guide for the perplexed., Boston: Addison Wesley, 2003.
- [49] R. K. Wysocki, Effective project management., Indianapolis: John Wiley & Sons, Inc. , 2007.
- [50] G. Chin, Agile project management: how to succeed in the face of changing project requirements, New York: AMACOM, 2004.
- [51] S. W. T. T. J. & H. D. Cicmil, «Rethinking Project Management: Researching the actuality of projects,» International Journal of Project Management, vol. 24, n° %18, p. 675–686, 2006.
- [52] S. W. C. H. B. & S. C. Collyer, «Aim, fire, aim – Project planning styles in dynamic environments.,» Project Management Journal, vol. 41, n° %14, p. 108–121, 2010.
- [53] S. C. T. C. L. & R. K. Cicmil, Exploring the complexity of projects: Implications of Complexity Theory for project management practice, Newtown Square, PE: Project Management Institute, 2009.
- [54] K. Aguanno, Managing agile projects, Canada: Multi–Media Publications Inc, 2004.
- [55] N. O. E. Olsson, «Management of flexibility in projects,» International Journal of Project Management, vol. 24, p. 66–74, 2006.
- [56] T. Williams, «Assessing and moving on from the dominant project management discourse in the light of project overruns,» chez Transactions on Engineering Management, IEEE, 2005, pp. 52(4), 497–508.
- [57] A. Cockburn, Agile Software Development: The Cooperative Game, Boston: Addison Wesley Professional, Pearson Education, Inc., 2006.
- [58] J. Highsmith, Agile project management, Boston: Addison–Wesley, 2004.
- [59] S. Rakitin, Letters, Manifesto elicits cynicism, Computer, 2001.
- [60] N. D. G. T. S. M. & O. P. Fogelstrom, «The impact of agile principles on market–driven software product development,» J. Softw. Maint. Evol.: Res. Pract, vol. 22, n° %11, p. 53–80, 2010.
- [61] T. & C. D. Chow, «A survey study of critical success factors in agile software projects,» The Journal of Systems and Software, vol. 81, n° %16, p. 961–971, 2008.
- [62] M. & B. S. Coram, « The impact of agile methods on software project management,» chez the 12th IEEE International Conference and Workshops on the Engineering of Computer, USA, Based Systems, 2005.
- [63] J. Charvat, Project Management Methodologies, Wiley, NJ, 2003.
- [64] K. B. Haas, «The blending of traditional and agile project management,» PM World Today, vol. 9, n° %15, 2007.
- [65] L. Virine, «Adaptive Project Management,» PM World Today, vol. 10, n° %15, 2008.
- [66] F. Atigui, Approche Dirigée Par Les Modèles Pour l’implantation et La Réduction d’entrepôts de Données, Toulouse, 2013.

- [67] T. D. N. J. P. P. A. & W. H. Meijler, «Supporting fine-grained generative model-driven evolution,» chez *Software & Systems Modeling*, 9 (3), 2010, pp. 403-424.
- [68] N. Tankovic, *Model Driven Development Approaches: Comparison and Opportunities*.
- [69] N. V. D. & Z. M. Tankovic, «Rethinking model driven development: analysis and opportunities,» chez *Information Technology Interfaces (ITI)*, Proceedings of the ITI 2012 IEEE 34th International Conference, 2012, pp. 505-510.
- [70] B. Combemale, *Ingénierie Dirigée Par Les Modèles (IDM) -- État de l'art., Management*, 2008.
- [71] OMG, *MDA Guide version 1.0.1*, OMG Document, 2003.
- [72] S. B. F. M. & M. S. Chaouni, *MDA based-approach for UML Models Complete Comparison*, arXiv preprint arXiv:1105.6128, 2011.
- [73] J. Bezivin, *Model Engineering for Software Modernization*, WCRE (p. 4), 2004.
- [74] C. & K. T. Atkinson, «foundation, Model-driven development: a metamodeling,» chez *Software*, IEEE, 2003, pp. 36-41.
- [75] A. G. W. J. B. & B. W. Kleppe, *MDA explained: the model driven architecture: practice and promise*, Addison-Wesley Professional, 2003.
- [76] O. M. Group, Inc. *Meta Object Facility (MOF) 2.0 Core Specification, Final Adopted Specification*, 2006.
- [77] J. E. a. M. B. Jean-Marie Favre, «Ingénierie Dirigée par les Modèles : au-delà du MDA,» chez *Informatique et Systèmes d'Information*, Hermes Science, lavoisier edition, 2006.
- [78] J. a. O. G. Bezivin, *Towards a Precise Definition of the OMG/MDA Framework*, *Bibliographie* 141, 2005.
- [79] X. a. O. S. Blanc, *MDA En Action: Ingénierie Logicielle Guidée Par Les Modèles*, Eyrolles, 2011.
- [80] F. F. R. O. T. a. Z. G. Atigui, «Modèle Unifiée Pour La Transformation Des Schémas En Constellation,» chez *7ème Journées Francophones Sur Les Entrepôts de Données et l'Analyse En Ligne*, France, EDA, 2011.
- [81] M. e. a. Lütjen, «Model-driven Logistics Engineering—Challenges of Model and Object Transformation,» *Procedia Technology*, 2014, pp. 303-312.
- [82] I. & S. Gmati, *Un cadre de référence pour analyser les exigences d'alignement métier / système d'information*, 2012.
- [83] C. Rolland, *A Comprehensive View of Process Engineering*, 2012.
- [84] T. N. C. Center, *Developing a successful governance strategy*, Novembre 2005.
- [85] M. M. S. Y. D. Jarke, «Conceptual Model-ing and Knowledge Based Support of Information Systems Development Process,» chez *Technique et Science Informatiques*, 1990, pp. 122-133.
- [86] B. Ménard, *COBIT : Pour une meilleure gouvernance des systèmes d'information*, Eyrolles, 2009.
- [87] H. I. a. B. Abdessamad, «A Built-in Criteria Analysis for Best IT Governance Framework,» *International Journal of Advanced Computer Science and Applications(IJACSA)*, vol. 10, n° %110, 2019.
- [88] H. I. a. A. Belangour, «Project Management Metamodel Construction Regarding IT Departments,» *International Journal of Advanced Computer Science and*

- Applications(IJACSA), vol. 10(10),
n° %1<http://dx.doi.org/10.14569/IJACSA.2019.0101029>, 2019.
- [89] The Standish Group, *The Crisis in Software : The Wrong Process Produces the Wrong Results*, The Standish Group, 2011.
- [90] F. F. A. J. B. a. I. K. Jouault, *ATL: A Model Transformation Tool*, Science of Computer Programming, 2008.
- [91] T. a. P. V. G. Mens, «A Taxonomy of Model Transformation,» chez *The International Workshop on Graph and Model Transformation (GraMoT 2005)*, vol. 152, *Electronic Notes in Theoretical Computer Science*, 2005, p. 125–142.
- [92] K. a. S. H. Czarnecki, «Classification of Model Transformation Approaches,» chez *the 2nd OOPSLA Workshop on Generative Techniques in the context of Model Driven Architecture*, 2003.
- [93] M. Scheidgen, «Model Patterns for Model Transformations in Model Driven Development,» chez *the 4th Workshop on Model-Based Development of Computer-Based Systems*, 2006.
- [94] M. K. B. A. A. a. M. S. Dehayni, «Some Model Transformation Approaches: a Qualitative Critical Review,» *Journal of Applied Sciences Research*, vol. 5, n° %111, p. 1957–1965, 2009.
- [95] ATL, *ATL Basic Examples and Patterns: Families to Persons - A simple illustration of model to model transformation*, Nantes: ATLAS Group, 2011.
- [96] M. e. B. A. BANANE, «Towards a New Scalable Big Data System Semantic Web Applied on Mobile Learning,» *International Journal of Interactive Mobile Technologies (IJIM)*, vol. 14, n° %101, pp. 126-140, 2020.
- [97] M. e. B. BANANE, «A Big Data Solution To Process Semantic Web Data Using The Model Driven Engineering Approach,» *International Journal of Scientific; Technology Research*, vol. 9, n° %102, 2020.
- [98] M. a. A. B. Banane, «A new system for massive RDF data management using Big Data query languages Pig, Hive, and Spark,» *International Journal of Computing and Digital Systems*, vol. 9, n° %102, pp. 259-270, 2020.
- [99] A. a. A. B. Erraissi, «Meta-Modeling of Big Data Management Layer,» *International Journal of Emerging Trends in Engineering Research*, vol. 7, n° %107, pp. 36-43, 2019.
- [100] M. A. E. a. A. B. Banane, «SPARQL2Hive: An approach to processing SPARQL queries on Hive based on meta-models,» chez *8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO)*, IEEE, 2019.
- [101] A. Erraissi, «Using model Driven Engineering to transform Big Data query languages to MapReduce jobs,» *International Journal of Computing and Digital Systems*, vol. 9, pp. 1-9, 2020.
- [102] I. a. A. B. HAMZANE, «Implementation of a decision system for a suitable IT governance framework,» *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 17, n° %15, 2019.
- [103] I. a. A. B. HAMZANE, «A Multi-criteria Analysis and Advanced Comparative Study Between IT Governance References,» *International Conference on Advanced Intelligent Systems for Sustainable Development*.
- [104] H. KNIBERG, *Scrum and Xp from the Trenches 2nd Edition*. 2. S.l., Vols. %1 sur %2ISBN 978-1-329-22427-8, Lulu.com, 2015.

- [105] P. S. a. R. S. G. Regev, Taxonomy of flexibility in business processes, Business Process Modelling, Development and Support (BPMDS), 2006.
- [106] M. S. B. a. S. M. P. W. Zhang, Stability of networked control systems, In IEEE Control Systems, 2001.
- [107] J. T. a. M. P. Heimdalh., Extending the product family approach to support, dimensional and hierarchical product lines.
- [108] P. Drucker., The age of discontinuity : guidelines to our changing society, In Harper and Row, New York.
- [109] M. B. V. B. B. C. P. D. K. S. F. T. R. W. L. e. Z. M. LINDVALL, «Empirical Findings in Agile Methodologies.» chez Extreme Programming and Agile Methodologies — XP/Agile Universe 2002. S.l., Springer Berlin Heidelberg, 2002, pp. 197-207.
- [110] M. ZELKOWITZ, Advances in Computers: Advances in Software Engineering. S.l., Vols. %1 sur %2ISBN 978-0-08-047190-7, Elsevier, 2004.
- [111] J. MARTIN, Rapid Application Development, Vols. %1 sur %2ISBN 978-0-02-376775-3, New York: Macmillan Coll Div., 1991.
- [112] K. BECK, Extreme Programming Explained: Embrace Change, Vols. %1 sur %2ISBN 978-0-321-27865-4, Boston: Addison Wesley, 2004.
- [113] J. e. A. P. KOSKELA, «On-Site Customer in an XP Project: Empirical Results from a Case Study. In : DINGSYR,» chez Software Process Improvement. S.l., Torgeir, Springer Berlin Heidelberg, 2004, pp. 1-11.
- [114] J. GRENNING, «Planning Poker or How to avoid analysis paralysis while release planning,» S.l., 2002. [En ligne]. Available: <https://wingman-sw.com/articles/planningpoker>.
- [115] N. C. HAUGEN, «An empirical study of using planning poker for user story estimation,» chez AGILE 2006, 2006, pp. 9-34.
- [116] H. e. N. I. TAKEUCHI, The New New Product Development Game, H. B. Review, Éd., 1986.
- [117] K. e. B. M. SCHWABER, Agile Software Development with Scrum, Vols. %1 sur %2ISBN 978-0-13-067634-4, Pearson: Upper Saddle River, 2001.
- [118] M. D. S. e. L. C. PAASIVAARA, «Using Scrum in Distributed Agile Development: A Multiple Case Study,» 2009.
- [119] H. e. N. I. TAKEUCHI, «The New New Product Development Game,» chez Harvard Business Review, Harvard Business Review, 1986.
- [120] M. D. S. e. L. C. PAASIVAARA, «Using Scrum in Distributed Agile Development: A Multiple Case Study,» chez Fourth IEEE International Conference on Global Software Engineering, Fourth IEEE International Conference on Global Software Engineering. S.l., 2009, pp. 195-204.
- [121] J. e. S. K. SUTHERLAND, The scrum papers: nut, bolts, and origins of an Agile framework, Scrum inc, 2011.
- [122] R. N. N. R. R. e. M. E. WEINREICH, Scrum as Method for Agile Project Management Outside of the Product Development Area, S.l., 2015.
- [123] P. e. a. Ribbers, «Designing information technology governance processes: diagnosing contemporary practises and competing theories,» chez Proceedings of the 35th Hawaii International Conference on System Sciences, Hawaii, HICCS), 2002.

- [124] R. Peterson, «Information strategies and tactics for information technology governance,» chez Van Grembergen, W. (ed.) Strategies for Information Technology Governance, Idea Group Publishing, 2004.
- [125] IT Governance Institute, IT Governance Global Status Report, ITGI.

Annexe A

Liste des publications

A.1 - Journaux internationaux (SCOPUS)

1. Safia Baali, Ibrahim Hamzane, Hicham Moutachaouik, Abdelaziz Marzak “A Multi-Criteria Analysis and Advanced Comparative Study of Recommendation Systems” International Journal of Engineering Trends and Technology 69.3(2021):69-75
<http://dx.doi.org/22315381/IJETT-V69I3P213>
2. HAMZANE Ibrahim, Belangour Abdessamad, Alae ERRAISSI and Mouad BANANE “MDA Transformation Process from predictive project management methodologies to agile project management methodologies” International Journal of Emerging Trends in Engineering Research, 2020, 8(9), pp. 5034–5042, 23
<http://dx.doi.org/10.30534/ijeter/2020/23892020>
3. HAMZANE Ibrahim and Belangour Abdessamad, “Project Management Metamodel Construction Regarding IT Departments” International Journal of Advanced Computer Science and Applications(IJACSA), 10(10), 2019.
<http://dx.doi.org/10.14569/IJACSA.2019.0101029>
4. HAMZANE Ibrahim and Belangour Abdessamad, “A Built-in Criteria Analysis for Best IT Governance Framework” International Journal of Advanced Computer Science and Applications(IJACSA), 10(10), 2019.
<https://dx.doi.org/10.14569/IJACSA.2019.0101026>
5. HAMZANE, Ibrahim et BELANGOUR, Abdessamad. Implementation of a decision system for a suitable IT governance framework. International Journal of Computer Science and Information Security (IJCSIS), 2019, vol. 17, no 5.
<https://sites.google.com/site/ijcsis/vol-17-no-5-may-2019>

A.2 - Conférences internationales (SPRINGER & SCOPUS)

1. HAMZANE Ibrahim, Belangour Abdessamad, Mouad Banane, Allae Erraissi “Proposed metamodels transformation from Predictive methodologies to Agile methodologies” International Conference on Decision Aid Sciences and Applications DASA’20, 2020
2. HAMZANE Ibrahim and Belangour Abdessamad, “Proposition d’un méta-modèle de la méthode PRINCE 2” International Conference on Research in Applied Mathematics and Computer Science, ICRAMICS, 2020.
3. HAMZANE Ibrahim and Belangour Abdessamad, “A Multi-Criteria Analysis and Advanced comparative study Between IT Governance references” The International Conference on Advanced Intelligent Systems for Sustainable Development - AI2SD’2019, 2019.
4. HAMZANE Ibrahim and Belangour Abdessamad, “Comparaison of ITIL implementation methods and different ontologies to model this reference” La 5ème conférence internationale sur les technologies de l’information et de modélisation, 2018.

A.3 - Conférences nationales

1. HAMZANE Ibrahim and Belangour Abdessamad, “Génération d’un méta-modèle de gestion de projets à partir d’une approche MDA” (JMA) 17ème journée de Mathématiques et Application, 2018
2. HAMZANE Ibrahim and Belangour Abdessamad, “Etude comparative des méthodes d’implémentation d’ITIL” Journée des Sciences de l’ingénieur (JSI 17), 2017.
3. HAMZANE Ibrahim and Belangour Abdessamad, “Vers la construction d’un méta-modèle ITIL” Journée des Sciences de l’ingénieur (JSI 16), 2016.

4. HAMZANE Ibrahim and Belangour Abdessamad, “Proposition d’un cadre de référence pour analyser les systèmes de gestion des services informatiques ITSM” Journée Mathématique et Applications (JMA 16), 2016.
5. HAMZANE Ibrahim and Belangour Abdessamad, “Etude comparative des ontologies d’ITIL” The 4th day on Information Technologies and Modeling (TIM), 2016.
6. HAMZANE Ibrahim and Belangour Abdessamad, “Etude comparative des méthodes d’implémentation d’ITIL” Journée des Sciences de l’ingénieur (JSI 16), 2016.

Annexe B

Travaux et certificats connexes

B.1 - Certificats de management

1. Project Management Essentials Certified, Management and strategie Institute (MSI), Juin 2020
2. Lean Six Sigma Yellow Belt, Management and strategie Institute (MSI), Juin 2020
3. Lean Six Sigma White Belt, Management and strategie Institute (MSI), Juin 2020
4. Certificat de gestion de projets PRISM, Société Générale, Août 2019
5. Agile Scrum Master, XIN Certification Institute, Mai 2019
6. Coach Agile certification, Cabinet IT-GOV Consulting International, Avril 2019
7. Technical Project Management, Cybrary, Juin 2018
8. Formation 5 jours PMP, Project Management Professional, Novembre 2016
9. Certification ITIL Fondation, Management des systèmes d'information, Mars 2016
10. Certification PRINCE 2 Fondation, Gestion de projets, Octobre 2015.

B.2 - Certificats de transformation digitale

1. KANBAN, Certiprof, 2020
2. Certification de formateur relais en Transformation Digital, Cabinet DigiTurns, Aout 2018
3. The Online Marketing Fundamentals, Google Digital Garage Certification, Novembre 2017

B.3 – Autres certificats

1. CompTIA certification Cloud+, Cybrary, 2019
2. Intro to AWS, Cybrary, 2019
3. CompTIA Linux+, Cybrary, 2019