



HAL
open science

Evaluation de la maturité des technologies médicales pilotées par logiciel : vers un modèle non contraignant pour la phase de recherche

Jean-Loup Haberbusch

► **To cite this version:**

Jean-Loup Haberbusch. Evaluation de la maturité des technologies médicales pilotées par logiciel : vers un modèle non contraignant pour la phase de recherche. Génie logiciel [cs.SE]. Université Grenoble Alpes [2020-..], 2024. Français. NNT : 2024GRALM019 . tel-04757201

HAL Id: tel-04757201

<https://theses.hal.science/tel-04757201v1>

Submitted on 28 Oct 2024

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

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

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : MSTII - Mathématiques, Sciences et technologies de l'information, Informatique

Spécialité : Informatique

Unité de recherche : Translational Innovation in Medicine and Complexity

Evaluation de la maturité des technologies médicales pilotées par logiciel : vers un modèle non contraignant pour la phase de recherche

Assessing the Maturity of Software-Driven Medical Technologies: Towards a Non-Binding Model for the Research Phase

Présentée par :

Jean-Loup HABERBUSCH

Direction de thèse :

Emmanuel PROMAYON

PROFESSEUR DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES

Directeur de thèse

Guillaume DARDENNE

CHARGE DE RECHERCHE, INSERM DELEGATION GRAND OUEST

Co-encadrant de thèse

Rapporteurs :

OLIVIER CARTIAUX

SENIOR SCIENTIST, ECOLE CENTRALE DES ARTS ET METIERS

JEAN-MICHEL BRUEL

PROFESSEUR DES UNIVERSITES, UNIVERSITE DE TOULOUSE

Thèse soutenue publiquement le **4 juillet 2024**, devant le jury composé de :

ERIC STINDEL,

PROFESSEUR DES UNIVERSITES, UNIVERSITE DE BREST-BRETAGNE OCCIDENTALE

Président

EMMANUEL PROMAYON,

PROFESSEUR DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES

Directeur de thèse

OLIVIER CARTIAUX,

SENIOR SCIENTIST, ECOLE CENTRALE DES ARTS ET METIERS

Rapporteur

JEAN-MICHEL BRUEL,

PROFESSEUR DES UNIVERSITES, UNIVERSITE DE TOULOUSE

Rapporteur

YANNICK MORVAN,

DOCTEUR EN SCIENCES, STRYKER

Examineur

SANDRINE VOROS,

DIRECTRICE DE RECHERCHE, INSERM DELEGATION AUVERGNE-RHONE-ALPES

Examinatrice

ALEXANDRE MOREAU-GAUDRY,

PROFESSEUR DES UNIVERSITES - PRATICIEN HOSPITALIER, UNIVERSITE GRENOBLE ALPES

Examineur

Invités :

GUILLAUME DARDENNE

CHARGE DE RECHERCHE, INSERM DELEGATION GRAND OUEST



À Jessy, Charlize, Hector & Érine,

RÉSUMÉ

De nombreux obstacles entravent l'émergence de nouvelles technologies médicales issues de la recherche. L'un des principaux défis du transfert de technologie réside dans le temps et les efforts nécessaires pour se conformer aux réglementations. Dans ce contexte, l'évaluation de la maturité des technologies médicales devient un aspect incontournable, mais représente un véritable défi.

Les méthodes d'évaluation de la maturité technologique sont des approches utilisées pour évaluer le niveau de développement et de fiabilité d'une technologie spécifique. Dans le domaine des systèmes d'information, ces méthodes sont couramment utilisées pour évaluer la maturité des infrastructures technologiques, des logiciels et des processus informatiques. Elles permettent d'identifier les forces et les faiblesses d'une technologie, d'orienter les décisions d'investissement et d'optimiser les processus de développement en vue d'une utilisation efficace de ces technologies dans leur contexte d'utilisation finale.

Cependant, l'utilisation des méthodes d'évaluation de la maturité existantes dans la phase de recherche n'est pas adaptée ni appropriée, étant donné le caractère contraignant qu'elles impliquent. L'objectif de cette thèse est de démontrer qu'il est possible de mettre en place une approche non contraignante pour évaluer et mesurer la maturité technologique dans la phase de recherche des technologies médicales pilotées par logiciel (SdMT, pour Software-driven Medical Technology).

Pour atteindre cet objectif, nos travaux fournissent les bases nécessaires pour la construction d'un modèle de maturité. Ce modèle vise à permettre l'évaluation quantitative et objective du niveau de préparation d'une SdMT, à un moment donné, en vue de son potentiel transfert. Ce transfert peut concerner des projets de recherche collaboratifs, des essais cliniques ou des processus soumis à des contraintes réglementaires. Le modèle proposé peut fournir des indications sur les actions à entreprendre pour accroître le niveau de maturité d'une SdMT.

Les travaux réalisés dans le cadre de cette thèse proposent une base solide composée d'un modèle d'artefacts de la norme ISO/IEC 62304, d'un questionnaire d'évaluation appelé SMAQ (SdMT Maturity Assessment Questionnaire) et d'un score de maturité. La validité du SMAQ a été démontrée par le biais d'une enquête menée auprès d'experts du domaine, ce qui a permis d'obtenir une version consolidée du questionnaire.

Sur la base des résultats obtenus, il a été possible de définir les premiers éléments d'un modèle de maturité, tels que les niveaux de maturité, la trajectoire vers la maturité la plus élevée et les questions d'évaluation de la maturité. Les dimensions ainsi que les sous-catégories du modèle de maturité restent à définir plus précisément dans le cadre de travaux futurs.

ABSTRACT

Many obstacles hinder the emergence of new medical technologies derived from research. One of the main challenges of technology transfer lies in the time and effort required to comply with regulations. In this context, the assessment of medical technology maturity becomes essential but represents a real challenge.

Technology maturity evaluation methods are approaches used to assess the level of development and reliability of a specific technology. In the field of information systems, these methods are commonly used to assess the maturity of technological infrastructures, software, and computer processes. They help identify the strengths and weaknesses of a technology, guide investment decisions, and optimize development processes for effective use of these technologies in their intended context.

However, using existing maturity evaluation methods in the research phase is not suitable or appropriate, given the constraints they entail. The objective of this thesis is to demonstrate that it is possible to establish a non-binding approach to evaluate and measure technological maturity in the research phase of software-driven medical technologies (SdMT).

To achieve this objective, our work provides the necessary foundations for constructing a maturity model. This model aims to enable quantitative and objective evaluation of the readiness level of an SdMT at a given point in time, with a view to its potential transfer. This transfer can involve collaborative research projects, clinical trials, or processes subject to regulatory constraints. The proposed model can provide indications of actions to be taken to increase the maturity level of an SdMT.

The work carried out in this thesis provides a solid foundation consisting of an artifact model based on the ISO/IEC 62304 standard, an evaluation questionnaire called SMAQ (SdMT Maturity Assessment Questionnaire), and a maturity score. The validity of SMAQ was demonstrated through a survey conducted among domain experts, which led to a consolidated version of the questionnaire.

Based on the results obtained, it was possible to define the initial elements of a maturity model, such as maturity levels, the path to the highest maturity, and maturity evaluation questions. The dimensions and subcategories of the maturity model are yet to be defined more precisely in future work.

REMERCIEMENTS

En concluant ce chapitre significatif de ma vie, je tiens à exprimer ma gratitude profonde à tous ceux qui ont jalonné le parcours qui m'a mené jusqu'à ces lignes finales. Chaque mot de cette thèse porte l'empreinte des nombreuses années passées au-delà des murs de l'université, dans le monde de l'entreprise, où j'ai appris que derrière chaque défi technique se cache une richesse humaine, et que chaque accomplissement est une aventure collective.

Je suis particulièrement reconnaissant envers mes encadrants de thèse, Emmanuel (Mahnu) Promayon et Guillaume Dardenne, dont l'expertise et l'immense patience ont été des phares dans la complexité de mes recherches. La capacité d'Emmanuel à guider sans imposer, mais sans transigeance non plus, a été importante pour moi. Il a apporté une résonance particulière à mes années d'expérience tout en valorisant ma quête d'approfondissement scientifique. Guillaume a su m'orienter sur des visions différentes avec son immense expertise de l'environnement clinique et scientifique. À chaque rencontre, il a su « booster » les voies alternatives pour avancer dans la direction la plus juste et pointer les incohérences.

Mes collègues de l'équipe GMCAO du laboratoire TIMC ont été une source constante d'inspiration et de soutien. Leurs perspectives diverses ont enrichi mon expérience, transformant chaque jour passé parmi eux en une leçon d'humilité et d'apprentissage. Jocelyne Troccaz, particulièrement, sans qui cette thèse n'existerait, tout simplement, pas ; merci à toi. Merci, Sandrine Voros, Taha Chikhaoui et Céline Fouard, pour votre soutien, qui au bon moment a permis de maintenir la continuité de mes travaux, et cela, bien avant la thèse.

Je tiens également à exprimer ma gratitude envers tous les doctorants, postdocs et ingénieurs du laboratoire. Les discussions techniques, souvent prolongées au-delà des heures habituelles, ont été tout aussi précieuses que les moments de détente partagés lors de nos pauses café. Ces instants de camaraderie et d'échange ont grandement contribué à rendre cette aventure académique à la fois humaine et joyeuse. Votre enthousiasme a été une ressource inestimable qui a illuminé même les jours les plus difficiles. À ceux qui ont soutenu et à ceux qui vont soutenir ; merci.

Mais surtout, merci à ma petite famille, qui a enduré non seulement cette longue absence physique lors de mes périodes de recherche, mais aussi mes absences mentales quand les défis se faisaient plus pressants. Mes enfants et ma douce épouse, si aimant, ont tant accepté. Cette petite famille si réconfortante et apaisante, sans qui, c'est sûr, tout accomplissement ne serait tout simplement pas possible. Jessy, Charlize, Hector et Érine : Merci. Votre soutien inébranlable a été mon ancrage et ma motivation. Ils m'ont donné la force de poursuivre cette aventure exigeante.

Chaque mot de cette thèse est dédié à chacun de vous, qui avez modelé l'homme et le professionnel que je suis aujourd'hui. Votre influence est tissée à travers les pages de ce travail, tout comme elle l'est dans le tissu de ma vie. Merci d'avoir cru en ce projet et de m'avoir fait confiance.

Table des matières

1	Les Gestes Médicaux Chirurgicaux Assistés par Ordinateur	1
1	Contexte et applications des GMCAO	1
1.1	La naissance des GMCAO : contexte et évolution	2
1.2	Les GMCAO vus par les cliniciens	5
1.3	La place des logiciels dans la recherche en GMCAO : perception, raisonnement et action	11
2	LabEx CAMI : Entre développement Logiciel et Innovation en Recherche Translationnelle	13
2.1	Présentation du LabEx CAMI	14
3	Du laboratoire vers le bloc opératoire	18
4	Développement logiciel et maturité technologique dans la recherche fondamentale	19
5	Organisation du Manuscrit	19
2	Maturité et Systèmes Médicaux Pilotés par Logiciel	21
1	Définition et trajectoire des technologies médicales pilotées par logiciel	21
1.1	Nos objectifs	21
1.2	Caractérisation des technologies médicales pilotées par logiciel	22
1.3	La recherche translationnelle et les technologies médicales pilotées par logiciel	24
1.4	Processus et terminologie des dispositifs médicaux	27
1.5	Problématiques du transfert clinique	32
2	Les parcours de progression des technologies médicales et le concept de Maturité	36
2.1	Le parcours de la technologie médicale	37
2.2	Échelle de maturité technologique	39
2.3	La qualité du point de vue des normes internationales	40
2.4	Naviguer entre réglementation et innovation en ingénierie logicielle	41
2.5	Les modèles de maturité dans les systèmes d'information	42
3	Conclusion	44
3	Artefacts de génie logiciel et modèle de maturité pour les SdMT	47
1	Rôle et évolution des artefacts de génie logiciel dans les SdMT	47
1.1	Genèse et définition des artefacts en génie logiciel	48
1.2	Amélioration de l'intégration des SdMT par les ontologies	49
1.3	Rôles et taxonomies des SEA dans les SdMT	51
1.4	Dynamique et adaptabilité des SEA dans le développement des SdMT	53
1.5	Évaluation de la maturité technologique des SEA dans les SdMT	58

2	Notre proposition : quantification de la maturité des SEA dans les SdMT	60
3	Conclusion	62
4	Évaluation de la maturité des technologies médicales pilotées par logiciel	65
1	Postulats	65
2	Les outils normatifs pour construire une référence de maturité technologique	68
2.1	La norme ISO/IEC 62304 : un cadre pour le développement logiciel médical	69
2.2	SEVOCAB : un guide terminologique	71
3	Méthodologie	73
3.1	Annotation du texte de la norme ISO/IEC 62304	75
3.2	Mappage	79
3.3	Classification des SEA	84
3.4	Synthèse des étapes d'annotation, de mappage et de classification	85
3.5	Conception de la méthode d'évaluation du score de maturité	88
4	Synthèse de la démarche	92
4.1	L'approche	92
5	Conclusion	93
5	Application à un projet de recherche développant une SdMT	95
1	Présentation du projet « FollowKnee »	96
1.1	Axes d'innovation de « FollowKnee »	96
2	Objectif de l'étude sur FollowKnee	97
3	Méthodologie d'expérimentation et collecte des données pour FollowKnee	98
4	Présentation des Résultats	98
4.1	Analyse des Résultats de l'Expérimentation sur FollowKnee	101
4.2	L'interview	103
4.3	Discussion sur l'expérimentation FollowKnee	104
5	Implications et ajustements de la méthodologie	105
6	Conclusion sur l'étude FollowKnee	106
6	Validation du modèle de Maturité	109
1	Validation statistique	109
1.1	Méthodologie systématique pour la conception et validation du SMAQ	111
1.2	Les tests de validité	112
1.3	Validité apparente	114
1.4	Validité de contenu	115
2	Application des techniques de validation au SMAQ	120
2.1	Formulaires de validation	120
2.2	L'enquête d'évaluation de la validité	123
3	Validation de nos résultats	123
3.1	SMAQ	123
3.2	Profils des experts	123
3.3	Validé apparente	124
3.4	Validité de contenu	125

3.5	Mise à jour du SMAQ	126
4	Synthèse de la validation du SMAQ	129
4.1	Experts	129
4.2	Validité apparente et de contenu	130
5	Conclusion	131
7	Discussion et Conclusion	133
1	Synthèse des contributions	133
1.1	Une démarche en trois étapes	133
1.2	Développement du modèle d'évaluation de maturité	134
2	Limitations et piste d'améliorations	138
2.1	La classification des SEA	138
2.2	Questionnaire d'évaluation	139
2.3	Modèle de maturité	139
3	Perspectives	140
4	Conclusion	141
A	Annexes	143
A.1	Listings	143
A.2	Termes annotés	146
A.3	SEA	151
A.4	Calcul du score sur un projet réel	158
	Bibliographie	175

TABLE DES FIGURES

Figure 1.1	Le canard digérateur de Jacques de Vaucanson (Image : Wikipedia(WIKIPEDIA 2024a))	3
Figure 1.2	Le conducteur de lumière de Philipp Bozzini, le premier endoscope (Image : Wikipedia(WIKIPEDIA 2024b))	3
Figure 1.3	Les principes méthodologiques des GMCAO aujourd’hui (TIMC 2024)	4
Figure 1.4	Coupe de volume tumoral reconstruite à partir de limites définies par tomodensitométrie informatisée, reformatée de manière orthogonale par rapport à l’approche chirurgicale. La position du rétracteur chirurgical (cercle) et celle du faisceau laser CO ₂ (curseur) sont affichées en relation avec la taille de la tumeur. (KELLY et al. 1986)	6
Figure 1.5	Station de neuronavigation et marqueur optique (GONÇALVES et al. 2018)	7
Figure 1.6	Simulation effectuée sur le biomodèle 3D (BAGARIA et al. 2017)	9
Figure 1.7	Robot da Vinci de la société Intuitive Surgical (VIVALTO-SANTÉ 2024)	10
Figure 1.8	Pour une immersion maximale, le chirurgien se trouve devant un écran stéréo et commande le système à l’aide de dispositifs haptiques pilotés par logiciels (STAUB et al. 2012)	13
Figure 1.9	Dispositif expérimental. Robot Prosper (1) ; aiguille (2) ; fantôme (3) ; sonde US (4). (LAPOUGE et al. 2021)	14
Figure 2.1	Le processus d’innovation Biodesign de Stanford	28
Figure 2.2	Phases possibles pour une SdMT	30
Figure 2.3	La progression de l’idée vers le patient MOREAU-GAUDRY et al. (2010)	38
Figure 2.4	Modèle générique de maturité	44
Figure 3.1	Le modèle en cascade	48
Figure 3.2	Utilisation de CIFMeDD pour la conception de dispositifs médicaux	51
Figure 3.3	Trois niveaux de perception pour les SEA	55
Figure 3.4	Metamodel de SEA	56
Figure 3.5	Structure en trois niveaux des SEA	57
Figure 3.6	Technologie médicale pilotée par logiciel (SdMT)	61
Figure 3.7	Évolution et mesure de la maturité d’une SdMT	63
Figure 4.1	Extraction et mappage des termes de la normes ISO/IEC 62304 en SEA	69
Figure 4.2	Exigence 5.6.8 de la norme ISO/IEC 62304	71
Figure 4.3	Quatres étapes pour construire le score de maturité	74
Figure 4.4	Annotation de l’exigence 5.6.8 de la norme ISO/IEC 62304. Pour le terme « ANOMALIES » la règle RA1 est appliquée (voir ci-dessous).	76

Figure 4.5	Répartition des SEA parmi les ensembles de mappage \mathcal{D} , \mathcal{C} et \mathcal{M}	81
Figure 4.6	Répartition des SEA en fonction de leur nombre de citations.	82
Figure 4.7	Termes annotés et SEA avec le plus grand nombre de citations.	83
Figure 4.8	Répartition des SEA entre les classes ① et ②	85
Figure 4.9	Liste des SEA de classe ① et de classe ② correspondant aux 20 SEA les plus cités, toutes classes confondues.	86
Figure 4.10	Proportion de citations par clause (diagramme circulaire externe) et répartition entre les SEA ① et les SEA ② pour chaque clause de la norme (diagramme circulaire interne).	87
Figure 4.11	Nombre et pourcentage de citations pour chaque clause de la norme.	87
Figure 4.12	Progression du score de maturité selon la classe des SEA	90
Figure 4.13	Extrait du questionnaire d'évaluation de la maturité	91
Figure 4.14	Synthèse de la démarche	92
Figure 5.1	Les objectifs du projet « FollowKnee »(PTG : Prothèse Totale de Genou) (image tirée du projet).	97
Figure 5.2	Répartition des SEA dans les réponses des ingénieurs selon leur classe.	102
Figure 5.3	Répartition des clauses par ingénieur.	103
Figure 5.4	Répartition des types de SEA développés pendant les phases de Modèle et FollowKnee.	105
Figure 6.1	Types de validité et leurs relations selon TAHERDOOST (2016).	113
Figure 6.2	Répartition des experts en fonction de leur expérience dans la norme ISO/IEC 62304	124
Figure 6.3	Distribution des CVI calculés à partir des réponses des experts. La ligne rouge montre la valeur de coupure entre les questions acceptées et rejetées à la valeur seuil $\theta = 0.8$	126
Figure 6.4	Nombre de citations pour chaque activité principale de la norme	127
Figure 6.5	Artéfacts les plus cités après validation	128
Figure 6.6	Impact des mises à jour du SMAQ sur l'évaluation de la maturité FollowKnee sur la base des mêmes réponses.	128
Figure 7.1	Le modèle de maturité pour la phase recherche	137

LISTE DES TABLEAUX

Table 1.1	Projets emblématiques du Laboratoire d'Excellence (LabEx) Computer Assisted Medical Intervention (CAMI).	15
Table 2.1	Solutions proposées pour le transfert et l'implémentation des technologies médicales selon différentes approches.	36
Table 4.1	Termes et définitions dans SEVOCAB	72
Table 4.2	Exemple du calcul d'un score	89

Table 4.3	Question type présente dans le questionnaire d'évaluation de la maturité. La définition provient de Software and Systems Engineering VOCABulary (SEVOCAB)	91
Table 5.1	Réponses au SMAQ pour la partie « Planning » de FollowKnee	99
Table 5.2	Répartition des données de réponse par ingénieur.	101
Table 5.3	Calcul de scores communs optimiste et pessimiste et comparaison avec les scores individuels.	101
Table 5.4	Répartition des SEA en fonction des clauses de la norme selon les réponses par ingénieur.	102
Table 6.1	Glossaire pour l'évaluation de la validité	110
Table 6.2	Seuils de Kappa pour l'interprétation de la force de l'accord selon Landis et Koch (1977).	115
Table 6.3	Termes utilisés pour désigner les éléments de nos outils d'évaluation de la validité du SMAQ	116
Table 6.4	Seuils de CVR en fonction du nombre de panelistes pour un niveau de signification de 0.05.	117
Table 6.5	Interprétation des valeurs du Coefficient AC2 de Gwet	120
Table 6.6	Seconde page du formulaire d'évaluation de la validité du SMAQ	121
Table 6.7	Exemple de questions présentes dans le formulaire d'évaluation de la validité de contenu.	122
Table 6.8	Évaluation de la validité apparente du SMAQ	122
Table A.1	165 termes annotés et 537 citations au total	146
Table A.2	Les 156 SEA	151
Table A.3	Calcul du score de maturité pour le projet FollowKnee	158
Table A.4	Réponses à la présence ou l'absence des SEA de l'ingénieur 1	162
Table A.5	Calcul du score de maturité pour le projet FollowKnee selon les réponses de l'ingénieur 1.	164
Table A.6	Réponses à la présence ou l'absence des SEA de l'ingénieur 2	168
Table A.7	Calcul du score de maturité pour le projet FollowKnee selon les réponses de l'ingénieur 2.	170

LISTINGS

Listing 4.1	Balisateur des termes pour l'annotation dans le texte de la norme	76
Listing A.1	Définition d'un environnement article en \LaTeX	143
Listing A.2	Définition d'une commande term en \LaTeX	144

ACRONYMES

ANR	Agence Nationale de la Recherche
ANSM	Agence Nationale de Sécurité du Médicament et des produits de santé
CAD	Computer Assisted Diagnostic
CAMI	Computer Assisted Medical Intervention
CamITK	Computer Assisted Medical Intervention Tool Kit
CDO	Configured Definitional Ontology
CIC-IT	Centres d'investigation clinique – Innovation technologique
CIFMeDD	Concept Ideation Framework for Medical Device Design
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
CNIL	Commission Nationale de l'Informatique et des Libertés
CNN	Convolutional Neural Networks
CREPS	Concept, Recherche, Essais, Produit, Soins
CT	Computed Tomography
CTR	Clinical Translational Research
CTSA	Clinical and Translational Science Award
CVI	Content Validity Index
CVR	Content Validity Ratio
CVT	Content Validity Template
DEO	Definitional Elements Ontology
DM	Dispositifs Médicaux
EMBC	Engineering in Medicine & Biology Society
FDA	Food and Drug Administration
GMCAO	Gestes Médicaux Chirurgicaux Assistés par Ordinateur
HAS	Haute Autorité de Santé
ICVI	Item-level-Content Validity Index
ICube	Laboratoire des Sciences de l'Ingénieur, de l'Informatique et de l'Imagerie
IEC	International Electrotechnical Commission
IRM	Imagerie par Résonance Magnétique
ISIR	Institut des Systèmes Intelligents et de Robotique
ISO	International Organization for Standardization
ITK	Insight Toolkit
JTC	Joint Technical Committee
LabEx	Laboratoire d'Excellence
LaTIM	Laboratoire de Traitement de l'Information Médicale
LIRMM	Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier

LTSI	Laboratoire de Traitement du Signal et de l'Image
MDR	Medical Device Regulation
MDS	Medical Device Software
NASA	National Aeronautics and Space Administration
NIH	National Institutes of Health
PET	Positron Emission Tomography
RMCM	Risk Management Capability Model
ROI	Region Of Interest
ROS	Robot Operating System
SaMD	Software as Medical Device
SCVI	Scale-level-Content Validity Index
SC	Sub-Committee
SdMD	Software-driven Medical Device
SdMT	Software-driven Medical Technology
SEI	Software Engineering Institute
SI	Système d'Information
SiMD	Software in Medical Device
SEA	Software Engineering Artifact
SEVOCAB	Software and Systems Engineering VOCABulary
SiMD	Software in Medical Device
SMAQ	SdMT Maturity Assessment Questionnaire
TIMC	Recherche Translationnelle et Innovation en Médecine et Complexité
TRL	Technology Readiness Level
VTK	The Visualization ToolKit

1

LES GESTES MÉDICAUX CHIRURGICAUX ASSISTÉS PAR ORDINATEUR

1	Contexte et applications des GMCAO	1
2	LabEx CAMI : Entre développement Logiciel et Innovation en Recherche Translationnelle	13
3	Du laboratoire vers le bloc opératoire	18
4	Développement logiciel et maturité technologique dans la recherche fondamentale	19
5	Organisation du Manuscrit	19

L'innovation, c'est une situation que l'on choisit parce qu'on a une passion brûlante pour quelque chose.

Steve Jobs

Les travaux présentés dans ce manuscrit trouvent leur fondement dans le contexte des technologies médicales pilotées par logiciel. Aujourd'hui, elles sont omniprésentes dans le parcours médical et ce phénomène devrait encore s'amplifier dans les années à venir. Elles peuvent être présentes tout au long de la chaîne de soins au patient et plus encore, dans tous les types d'environnements de la mise en œuvre de ces soins. Aussi bien à l'hôpital qu'au domicile des patients.

Par conséquent, pour faire face aux nombreux défis qui s'annoncent encore, les recherches dans ce domaine sont primordiales. Les Gestes Médicaux Chirurgicaux Assistés par Ordinateur (GMCAO) ont une influence majeure dans le développement de ces technologies, comme nous allons le voir.

En France, le LabEx CAMI, a permis de créer un environnement particulièrement dynamique pour le développement de ces recherches. Le logiciel y tient une place importante.

1 CONTEXTE ET APPLICATIONS DES GMCAO

La chirurgie moderne a subi une transformation remarquable grâce aux avancées technologiques. Parmi ces progrès, les GMCAO ont émergé comme un domaine en pleine expansion.

Les GMCAO désignent l'utilisation de logiciels spécialisés, d'algorithmes, de robots ou d'autres équipements, pour assister et guider les cliniciens dans leurs interventions. Ils permettent d'atteindre un niveau de précision et de sécurité très élevé et contribuent ainsi à améliorer significativement les résultats cliniques.

Dans cette section, nous présenterons le domaine des **GMCAO**. Nous examinerons leur évolution historique. Nous tracerons les origines des **GMCAO**, depuis leurs premières applications jusqu'à leur rôle actuel dans la chirurgie moderne. Nous nous efforcerons de mettre l'accent sur l'intégration croissante des logiciels de plus en plus complexes et avancés dans leur fonctionnement.

Nous explorerons ensuite les différentes applications cliniques des **GMCAO** mettant en lumière les avantages en termes de précision et de réduction des risques pour les patients. Nous verrons également des exemples concrets illustrant l'impact qu'ont les **GMCAO** sur les pratiques chirurgicales.

Enfin, cette section aborde les progrès et les défis liés au domaine des **GMCAO**. Nous discuterons de l'importance de la recherche continue et du développement de logiciels en phase de recherche. La recherche continue en technologie médicale représente un effort incessant pour innover et adapter les solutions technologiques aux besoins médicaux évolutifs et aux nouvelles découvertes scientifiques.

Les perspectives seront également abordées, en particulier, la place croissante des logiciels dans les nouvelles technologies, leur rôle moteur dans le processus d'innovation et l'obligation croissante de prendre en compte la réglementation en vigueur dès les projets de recherches.

1.1 La naissance des GMCAO : contexte et évolution

Historiquement les **GMCAO** trouvent leurs racines dans la nécessité constante d'améliorer les pratiques médicales et chirurgicales. Ils tentent de répondre aux défis posés par la complexité inhérente aux gestes médicaux, à la précision requise, mais aussi à la minimisation des risques pour les patients. Ces systèmes ont évolué de manière significative au fil du temps en s'attachant à répondre à l'ensemble de ces défis.

Au XVIII^e siècle, l'automatisation a marqué les premiers pas vers ce qui deviendrait les **GMCAO**. L'invention du « Canard digérateur » présentée en 1738 (figure 1.1) par Jacques de Vaucanson, en collaboration avec le chirurgien La Cat, a jeté les bases conceptuelles des futures technologies médicales (MORAN 2007). Cette innovation a montré l'intérêt d'utiliser des machines pour reproduire et comprendre les fonctions biologiques, mais aussi l'importance de la synergie entre technologie et médecine.

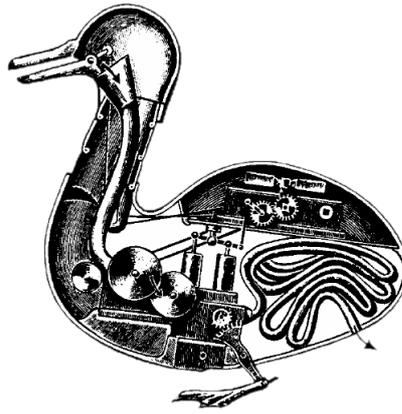


FIGURE 1.1 – Le canard digérateur de Jacques de Vaucanson (Image : Wikipedia(WIKIPEDIA 2024a))

Le XIXe siècle a vu d'importants progrès en optique et mécanique, culminant avec l'invention de l'endoscope (figure 1.2) par Philipp Bozzini en 1806 (ENGEL 2003). Cette avancée a permis une exploration visuelle interne du corps humain.

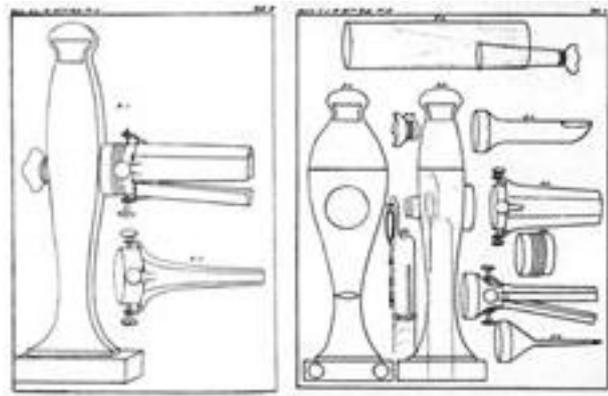


FIGURE 1.2 – Le conducteur de lumière de Philipp Bozzini, le premier endoscope (Image : Wikipedia(WIKIPEDIA 2024b))

Au XXe siècle, la convergence des progrès en physique, mathématiques et informatique a créé des opportunités pour les GMCAO. Des domaines tels que la modélisation anatomique, la simulation des gestes médicaux et la planification chirurgicale assistée par ordinateur ont commencé à se développer. La conjonction de techniques provenant de chacun de ces domaines a contribué à fournir des outils plus précis et des informations détaillées. Ces avancées ont permis l'avènement des interventions assistées par ordinateur (CINQUIN et al. 1995) préfigurant l'arrivée des GMCAO comme nouvelle discipline. Les principes méthodologiques ont été décrits dans l'article de TROCCAZ et al. (1996). Dans ces travaux les auteurs définissent un processus articulé autour de la « perception », de la « décision (ou raisonnement) » et de l'« action ». Cette méthodologie fonde l'approche adoptée dans ce domaine et qui a évolué vers « la roue des GMCAO », comme illustrée par la figure 1.3.

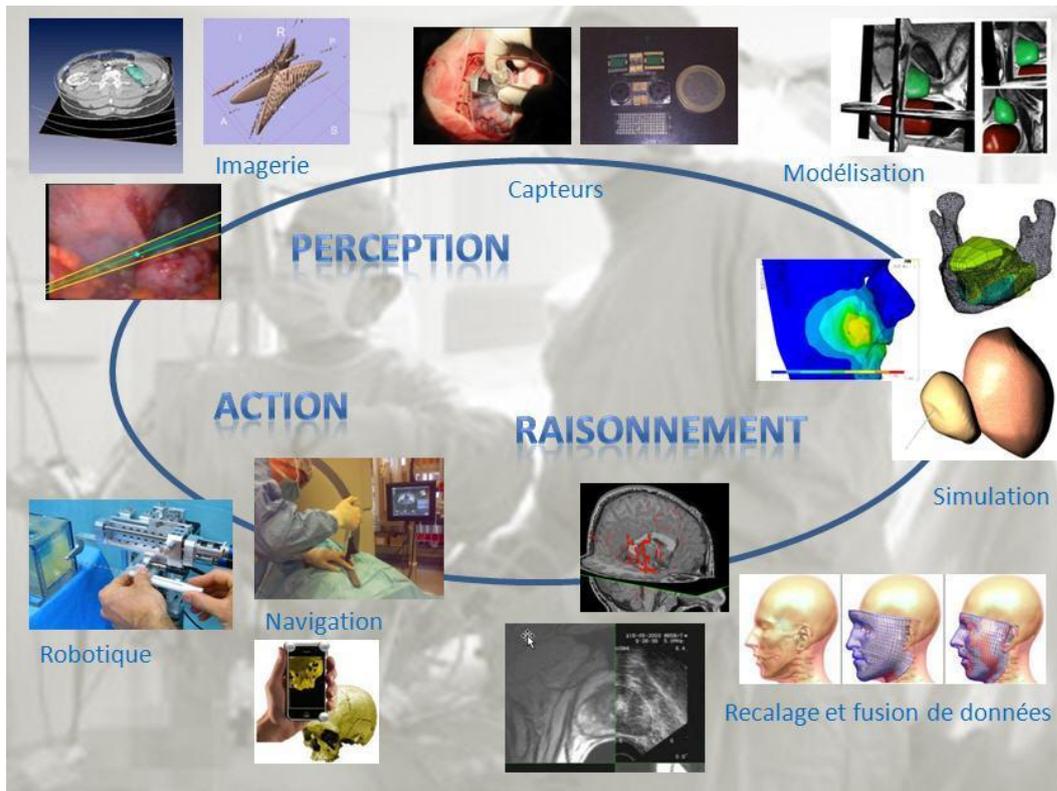


FIGURE 1.3 – Les principes méthodologiques des GMCAO aujourd’hui (TIMC 2024)

La perception repose sur l’utilisation des données acquises au moyen de technologies d’imagerie ou de capteurs, et cela, tout au long du traitement du patient. L’information fournie permet de compléter la connaissance directe du clinicien. Ces données peuvent être utilisées pour créer des modèles numériques liés au patient lui-même, comme des structures anatomiques par exemple. Elles permettent également le suivi des changements structurels ou le déplacement des dispositifs médicaux. Comme les mouvements des organes ou les déplacements des outils suivis en cours d’intervention ou pour une analyse ultérieure, par exemple.

Le raisonnement regroupe l’ensemble des outils d’aide à la décision que le clinicien pourra utiliser à toutes les phases du traitement. Il peut s’agir d’outils de planification avant l’opération chirurgicale, d’outils d’ajustement au cours de l’intervention, ou après.

La planification en chirurgie est un processus qui consiste à établir un plan d’action détaillé avant de procéder à une intervention chirurgicale. Il comprend l’évaluation de l’état de santé du patient, l’analyse des résultats des examens médicaux, la définition de la stratégie chirurgicale la plus appropriée et la préparation des équipements et du personnel nécessaires. La planification chirurgicale permet d’anticiper les différentes étapes de l’intervention, ainsi que de prévenir les complications potentielles.

Les outils en lien avec le raisonnement sont basés sur des algorithmes permettant d’étudier ou de considérer différents scénarios. Il peut donc, également s’agir de système de simulation. Par exemple, les algorithmes peuvent utiliser des don-

nées basées sur l'analyse statistique de patients antérieurs ou de patients spécifiques, mais également des analyses faites sur l'interaction des outils avec les tissus.

L'action se situe autour de la préparation, de l'exécution et du suivi de l'intervention du clinicien. Pour cela il utilise des systèmes informatiques qui vont l'assister dans la réalisation de l'acte. Lors de la préparation, ces systèmes peuvent amener à la réalisation d'examen complémentaires ou d'actes de préparation supplémentaires. Pendant la réalisation de l'acte, lors de la phase peropératoire pour une intervention chirurgicale par exemple, le système utilisé peut guider le geste et permettre une meilleure précision en fournissant des informations de localisation. Ce principe d'action sous-entend des algorithmes potentiellement capables de retour en temps réel pour permettre une adaptation dynamique des systèmes.

Dans les dernières décennies, l'évolution des GMCAO a vu la mise en œuvre, complexe, de techniques et technologies avancées comme la robotique chirurgicale (TROCCAZ 2009) et la chirurgie assistée par ordinateur, par exemple. De plus, les avancées en intelligence artificielle et en imagerie médicale, par exemple les travaux de DUPUY et al. (2023), laissent entrevoir de très nombreuses perspectives d'évolution.

Aujourd'hui, les GMCAO représentent le fruit d'une évolution constante. Cet essor continu des GMCAO est propulsé par une collaboration étroite entre professionnels de santé, ingénieurs et chercheurs dans de nombreux domaines. Cette discipline offre des outils de pointe pour améliorer les résultats cliniques et la qualité des soins prodigués. Pour le montrer, nous verrons dans la suite quelques applications concrètes de ces technologies dans divers domaines médicaux en nous concentrant sur le bénéfice apporté au monde médical ainsi qu'au patient.

1.2 Les GMCAO vus par les cliniciens

Les GMCAO trouvent des applications variées dans divers domaines médicaux. Dans cette section, nous ne nous focalisons pas sur une discipline ou un domaine chirurgical en particulier. Nous fournissons des exemples les plus parlants possibles pour illustrer nos propos autour des technologies médicales pilotées par logiciel, notamment dans les domaines de la neurochirurgie, de la chirurgie orthopédique et d'une manière plus large, de la chirurgie minimalement invasive, indépendamment du domaine chirurgical dans lequel elle est mise en œuvre. Ce que nous voulons montrer est le gain qu'apportent les technologies de GMCAO et l'importance des logiciels pour atteindre ces résultats.

QUELQUES EXEMPLES EN NEUROCHIRURGIE

Les GMCAO ont été utilisés avec succès dans de nombreux cas de traitements des pathologies. Par exemple, la neuronavigation assistée par ordinateur a eu un impact énorme dans le domaine de la neurochirurgie dès le milieu des années 80. Dans leur étude, KELLY et al. (1986) indiquent comment la technologie de navigation a permis une localisation précise en trois dimensions du volume tumoral. Pour cela, ils ont utilisé une technique chirurgicale appelée stéréotaxie, qui permet de cibler des

zones très spécifiques à l'intérieur du cerveau ou d'autres tissus en utilisant une méthode basée sur un système de coordonnées tridimensionnelles. En couplant cette technique de repérage avec des équipements et des logiciels on obtient le guidage stéréotaxique. Comme le montre la figure 1.4, le logiciel a permis la résection d'un volume tumoral dans l'espace stéréotaxique précis, une planification chirurgicale détaillée et le guidage du laser CO₂ dans cet espace stéréotaxique. Nous sommes ici en présence de technologies médicales pilotées par logiciel. Elles apportent une assistance et un guidage au chirurgien sur plusieurs aspects de sa tâche.

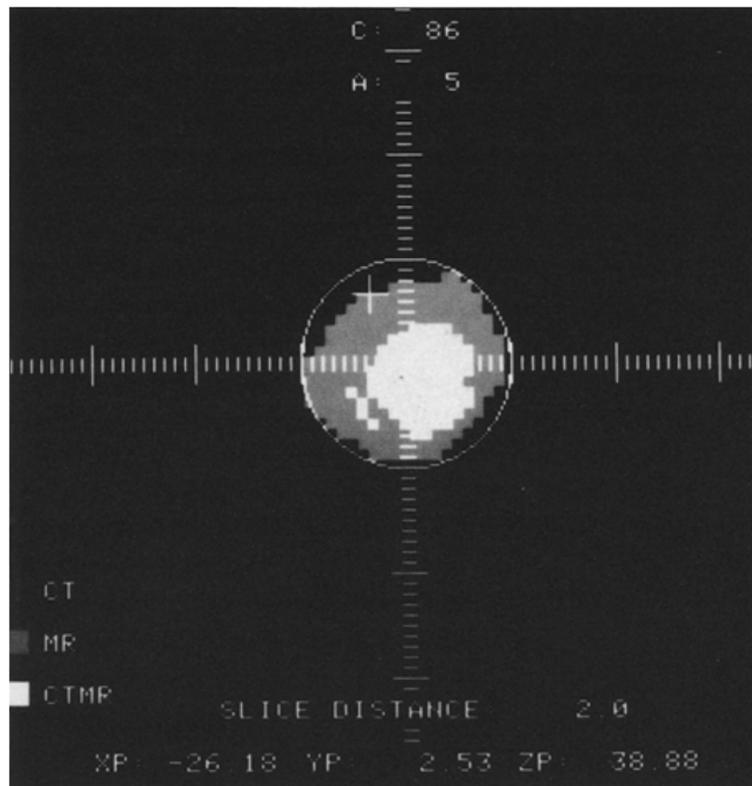


FIGURE 1.4 – Coupe de volume tumoral reconstruite à partir de limites définies par tomographie informatisée, reformatée de manière orthogonale par rapport à l'approche chirurgicale. La position du rétracteur chirurgical (cercle) et celle du faisceau laser CO₂ (curseur) sont affichées en relation avec la taille de la tumeur. (KELLY et al. 1986)

Plus récemment, dans leur étude MOLITERNO et al. (2012), montrent comment l'utilisation courante du guidage stéréotaxique, de l'imagerie peropératoire, de l'imagerie par résonance magnétique fonctionnelle et de la surveillance physiologique a amélioré la capacité du chirurgien à réaliser l'ablation d'une tumeur tout en protégeant le patient d'une atteinte neurologique.

BROMMELAND et al. (2002) ont montré que, non seulement la précision du geste chirurgical est améliorée, mais aussi que l'utilisation de systèmes pilotés par logiciel permet une mise en œuvre facilitée et un plus large éventail d'interventions.

Plus encore, RIZEA et al. (2019) indiquent que l'utilisation des systèmes de neuro-navigation pour la résection des tumeurs intradurales de la moelle épinière entraîne une diminution des risques de mauvaise localisation de la tumeur. Elle permet une technique chirurgicale moins invasive et une réduction de l'ablation osseuse.

RIZEA et al. (2019) montrent, en se basant sur une étude de STEFINI et al. (2018), que les systèmes de neuronavigation permettent une diminution de la mortalité et de la morbidité. Ce qui est confirmé dans de nombreuses études et notamment dans les travaux de GEBHARD et al. (2006) pour qui, l'utilisation de dispositifs chirurgicaux assistés par ordinateur, dans le cadre de la chirurgie du rachis, permet de diminuer le temps d'irradiation peropératoire.

Ces études que nous venons de décrire soutiennent le fait que l'intégration de logiciels a permis des avancées importantes dans le domaine de la neurochirurgie. Mais au-delà, on observe que l'outil informatique apporte une réelle valeur ajoutée à une technique par nature très complexe.

Le logiciel, en conjonction avec d'autres éléments comme des technologies de localisation (figure 1.5), permet de dépasser cette complexité et, notamment, de rendre possible la prise en compte de cas très difficiles. Dans les cas que nous venons de décrire, le bénéfice de l'utilisation des technologies médicales pilotées par logiciel est démontré. Elles apportent une assistance optimale au chirurgien, diminuent les risques pendant l'opération et diminuent les complications postopératoires.



FIGURE 1.5 – Station de neuronavigation et marqueur optique (GONÇALVES et al. 2018)

QUELQUES EXEMPLES EN CHIRURGIE ORTHOPÉDIQUE

De la même manière en chirurgie orthopédique les **GMCAO**, au travers de différents outils comme les systèmes de navigation ou d'assistance à la chirurgie, ont été mises en œuvre avec succès dans de nombreux cas. Là encore les technologies médicales pilotées par logiciel prennent une part importante. Par exemple, pour une implantation précise de prothèses de hanche ou pour le remplacement total du genou de nombreuses études montrent l'apport essentiel des systèmes de navigation assistés par ordinateur. Il s'agit d'ensembles technologiques utilisés pour guider avec précision les instruments chirurgicaux pendant les interventions médicales.

Ces systèmes fournissent une représentation visuelle en temps réel de la position des instruments chirurgicaux par rapport à l'anatomie du patient, en s'appuyant sur des données acquises lors des phases pré et peropératoires. Ils offrent au chirurgien une meilleure orientation spatiale et des informations détaillées pendant l'intervention comme nous l'avons déjà constaté pour la navigation stéréotaxique.

Ces avantages sont démontrés dans l'étude menée par DEEP et al. (2017). Elle repose sur l'analyse des données spécifiques du registre australien des articulations. Ils y comparent les taux de révision des arthroplasties¹ du genou réalisées avec l'aide des techniques de navigation à celles réalisées par les techniques traditionnelles. Ce taux de révision est réduit dans le cas de l'utilisation de techniques assistées par navigation. Ce résultat est rendu possible par un meilleur positionnement et une meilleure précision dans les interventions de remplacement de la hanche et du genou, du fait de l'utilisation de ces techniques assistées par ordinateur. Les auteurs mettent ainsi en évidence l'apport important des technologies médicales pilotées par logiciel à la chirurgie orthopédique. Cette étude montre aussi le bénéfice pour le patient dans la qualité des soins qui lui sont apportés.

Dans l'étude de BAGARIA et al. (2017), le développement de nouveaux logiciels a permis de mettre au point des outils de planification innovants. BAGARIA et al. (2017) ont utilisé des fantômes² issus de l'impression 3D dans la planification chirurgicale et la simulation.

Cinquante cas chirurgicaux ont été analysés, pour lesquels des biomodèles 3D ont été proposés aux chirurgiens orthopédiques. Ces modèles ont servi à comprendre l'anatomie pathologique et à effectuer des simulations chirurgicales dans le cadre de la planification préopératoire. La figure 1.6 présente l'utilisation qui est faite du biomodèle 3D pour le cas réel présenté dans cette étude, afin de préparer l'intervention. Tous les chirurgiens ont rapporté que les biomodèles leur ont fourni des informations supplémentaires par rapport à l'imagerie conventionnelle. Leur compréhension de la pathoanatomie complexe a été améliorée. Ils ont trouvé cela utile pour la planification préopératoire, la simulation chirurgicale, la navigation chirurgicale, la sélection préopératoire des implants et la gestion de l'inventaire.

Comme on peut le voir, les logiciels de planification et de navigation aident les chirurgiens à optimiser la position et l'orientation des implants, avec l'objectif d'offrir une meilleure mobilité articulaire par un positionnement optimal des implants.

1. L'arthroplastie qui peut être totale ou partielle est une intervention chirurgicale qui consiste à réparer ou remplacer une articulation endommagée

2. un « fantôme » est un terme utilisé pour désigner un modèle physique qui simule les propriétés d'un tissu biologique humain.

Cela conduit à de meilleurs résultats fonctionnels pour les patients, un rétablissement plus rapide et une véritable diminution du risque.

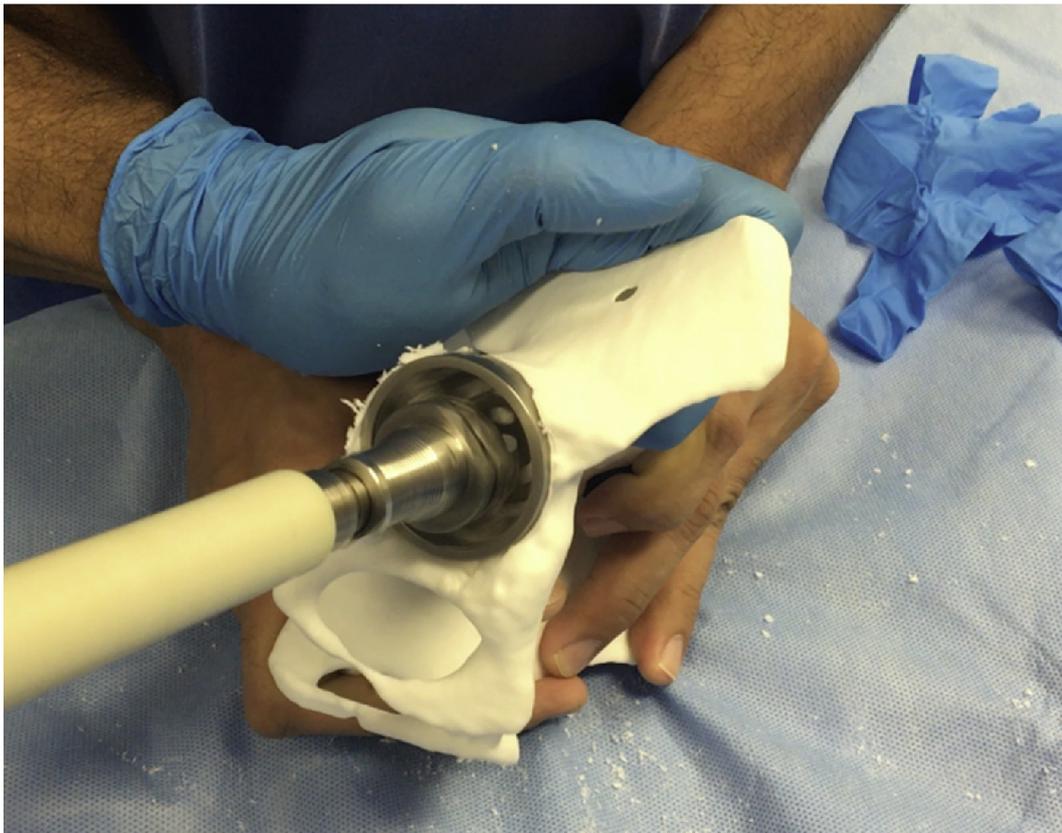


FIGURE 1.6 – Simulation effectuée sur le biomodèle 3D (BAGARIA et al. 2017)

QUELQUES EXEMPLES EN CHIRURGIE MINI-INVASIVE

Les technologies médicales pilotées par logiciel touchent encore bien d'autres domaines. L'univers médical et notamment celui de la chirurgie comprennent encore de nombreuses disciplines où les technologies des GMCAO ont été utilisées avec succès.

La chirurgie minimalement invasive³ où les technologies ont permis de réelles améliorations en apportant de l'assistance et plus de confort dans les gestes, pour le chirurgien.

Dans ce domaine, on peut évoquer le robot da Vinci de la société « Intuitive Surgical » (INTUITIV 2023), il est un concentré d'innovation et de technologie. La figure 1.7 présente ce robot en situation au Centre médical chirurgical obstétrical de la Côte d'Opale.

Bien sûr, la chirurgie minimalement invasive est un domaine très large, qui n'est pas exclusivement lié au robot « Da Vinci ». D'autres exemples permettent d'illustrer le développement des technologies autour du domaine de la chirurgie minimalement invasive et de la robotique. Voros et al. (2010) ont proposé le porte endo-

3. La chirurgie mini-invasive, également appelée chirurgie à faible effraction ou chirurgie laparoscopique, est une technique chirurgicale qui utilise des incisions beaucoup plus petites que celles employées en chirurgie traditionnelle à ciel ouvert. Par exemple, on parle de chirurgie minimalement invasive pour la mise en place de prothèse de genou si l'incision fait 10 cm.

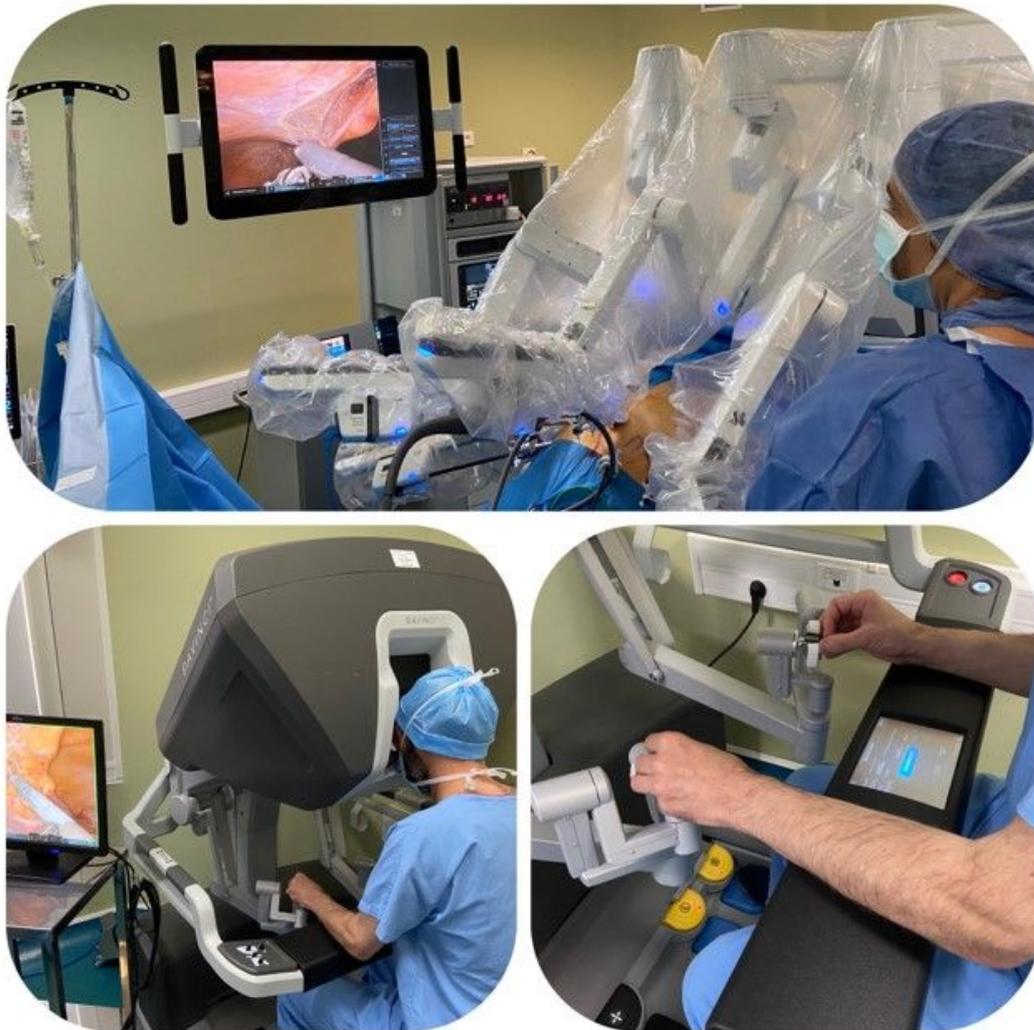


FIGURE 1.7 – Robot da Vinci de la société Intuitive Surgical (VIVALTO-SANTÉ 2024)

scope « ViKY ». Il est conçu pour améliorer l'autonomie du chirurgien et la précision des procédures laparoscopiques en fournissant une "troisième main" robotique.

Ce système léger et entièrement stérilisable peut être commandé par pédale ou par commande vocale. Les essais cliniques initiaux ont démontré l'utilité de ViKY dans de nombreuses applications chirurgicales.

Dans ce dispositif, la part du logiciel est importante, entre autres, pour améliorer les interactions entre le chirurgien et le robot. Comme la possibilité d'utiliser un logiciel de reconnaissance vocale pour permettre au chirurgien de commander les mouvements de l'endoscope de manière intuitive. Un mode de contrôle alternatif basé sur le suivi visuel des instruments chirurgicaux est également proposé.

La robotique chirurgicale est, bien entendu, utilisatrice de technologies logicielles. Les logiciels contrôlent précisément les bras robotisés, comme dans les exemples que nous venons d'évoquer. Ils traduisent les intentions du chirurgien en mouvements précis. Dans son article TROCCAZ (2009) présente un panorama de la robotique chirurgicale et de l'apport du logiciel. L'auteur souligne également les problématiques soulevées dans ce domaine particulier de la recherche.

En radiologie interventionnelle, on utilise le traitement d'image pour augmenter le niveau d'information auquel le radiologue interventionnel a accès. Dans le cadre de la chirurgie cardiaque et dans le domaine de la curiethérapie, également, on pourrait citer beaucoup de publications qui présentent leurs résultats indiquant l'apport qu'ont eu les technologies logicielles.

Ces quelques exemples montrent comment les GMCAO et plus particulièrement les technologies médicales pilotées par logiciel peuvent être intégrées dans des solutions en vue d'assister l'équipe médicale dans le traitement de pathologies.

Ayant mis en lumière ces quelques applications des GMCAO dans le domaine médico-chirurgical, nous allons maintenant nous concentrer sur les aspects du logiciel lui-même. Pour cela, nous nous positionnons plus en amont, au niveau de la recherche, qui représente les activités auxquelles nous nous intéressons dans ces travaux. Plus particulièrement, nous examinons comment les technologies logicielles sont développées dans le domaine des GMCAO.

1.3 La place des logiciels dans la recherche en GMCAO : perception, raisonnement et action

Comme nous venons de le voir, les GMCAO reposent largement sur le logiciel. Cette sous-section explore, du point de vue des GMCAO, plus particulièrement de la boucle « perception-raisonnement-action », cet aspect des technologies sans toutefois dresser une liste exhaustive.

PERCEPTION Dans les systèmes de GMCAO, la perception s'applique au traitement des données médicales pour fournir au clinicien une représentation la plus précise possible du patient.

Par exemple, les algorithmes de segmentation d'image, tels que ceux basés sur des réseaux de neurones convolutifs (Convolutional Neural Networks (CNN)), jouent un rôle de plus en plus important dans l'identification des structures anatomiques. Les travaux de KER et al. (2018) montrent comment l'intégration des CNN dans les systèmes GMCAO et plus particulièrement dans l'analyse d'image, peuvent améliorer la précision dans la préparation des gestes chirurgicaux grâce à l'extraction automatique d'informations à partir des images médicales (Imagerie par Résonance Magnétique (IRM), tomodensitométrie ou échographie).

Dans leur étude, DERATHÉ et al. (2020) utilisent des techniques d'intelligence artificielle pour prédire la qualité de l'exposition chirurgicale. Ce terme désigne la qualité de la visibilité et de l'accessibilité de la scène chirurgicale lors d'une opération. Cela donne un autre exemple d'amélioration de la perception.

D'autres techniques, plus conventionnelles, peuvent être appliquées à l'imagerie bien sûr. Par exemple, on peut évoquer les méthodes de filtrage et d'amélioration de l'image. L'objectif est d'améliorer la qualité visuelle des images et de les préparer à une future analyse, par exemple pour une segmentation. À partir des images acquises, on peut également utiliser des techniques de reconstruction 3D en utilisant un ensemble de contours 2D extraits d'une série d'images (IRM ou scanner, etc.). Cette technique permet de construire des modèles tridimensionnels des structures anatomiques voulues.

En plus, de ces techniques de reconstruction on peut aussi citer les techniques de fusion d'images provenant de différentes modalités (Positron Emission Tomography ([PET](#)), Computed Tomography ([CT](#)), [IRM](#)). Il s'agit là d'imagerie multimodale qui permet de créer une image composite offrant une vue plus complète de la structure anatomique et de la fonction physiologique. L'imagerie multimodale repose elle-même sur la mise en œuvre de techniques logicielles complexes pour être à même de procéder à la fusion.

Des avancées significatives sont réalisées dans le traitement et l'analyse des images médicales. Elles facilitent la tâche de localisation des régions d'intérêt (Region Of Interest ([ROI](#))) pour les cliniciens.

De plus, l'intégration d'autres types de traitement, comme la réalité augmentée, qui offre une vue « temps réel » et superposée de la zone chirurgicale, améliore encore la perception spatiale et la compréhension de l'anatomie complexe par le chirurgien. Ces techniques sont continuellement raffinées pour fournir une représentation toujours plus fidèle et utile du champ opératoire.

RAISONNEMENT Cette étape a pour objectif d'évaluer les données perçues pour élaborer des plans chirurgicaux personnalisés et proposer des stratégies d'intervention. Des logiciels de simulation utilisent des données biomécaniques pour prédire la réaction des tissus aux différentes techniques opératoires. Cela aide à éviter des complications potentielles et à optimiser les résultats chirurgicaux. Par exemple, [SELMI et al. \(2014\)](#) décrivent l'amélioration d'un simulateur de biopsie de la prostate guidée par échographie transrectale grâce à la simulation des mouvements et déformations réalistes de la prostate en temps réel. Les auteurs utilisent un modèle biomécanique discret pour modifier une texture 3D d'un volume d'image par ultrasons (échographies). Cela permet de simuler rapidement les déplacements et déformations réels. Pour y parvenir, ils utilisent le « framework » Computer Assisted Medical Intervention Tool Kit ([CamiTK](#)) dans lequel ils mettent en œuvre différentes techniques au moyen de différentes bibliothèques (The Visualization ToolKit ([VTK](#)), Insight Toolkit ([ITK](#)), etc.) déjà connectées à [CamiTK](#) et en utilisant le langage C++.

ACTION L'action en GMCAO peut se matérialiser par l'exécution concrète de la procédure médicale, guidée et assistée par des systèmes logiciels avancés. Les outils de navigation chirurgicale et les bras robotiques, pilotés par des logiciels, permettent une plus grande précision dans les mouvements chirurgicaux. Ces technologies permettent la conversion des gestes de la main en commandes chirurgicales.

C'est le cas dans les travaux de [STAUB et al. \(2012\)](#), où les auteurs mettent en place une gestion des entrées haptiques qui facilite l'interaction physique avec les outils chirurgicaux.

Un autre exemple de technologies médicales pilotées par logiciel dans le domaine de l'action est le guidage d'aiguilles robotisées. Dans l'article de [LAPOUGE et al. \(2021\)](#), une nouvelle méthode de guidage de l'aiguille en 3D est présentée. Le démonstrateur est présenté à la figure 1.9, il est piloté par un prototype logiciel construit avec l'aide du « framework » [CamiTK](#) et opéré par un système robotique développé au laboratoire, le robot « PROPSER ».



FIGURE 1.8 – Pour une immersion maximale, le chirurgien se trouve devant un écran stéréo et commande le système à l’aide de dispositifs haptiques pilotés par logiciels (STAUB et al. 2012)

Dans ce dernier exemple, l’utilisation du « framework » [CamiTK](#) a permis la production rapide d’un prototype d’expérimentation. Il existe bien d’autres types d’éléments logiciel en relation avec le domaine de la recherche en [GMCAO](#).

La mise en œuvre des concepts présentés dans la boucle, perception-décision-action sont à la base d’une recherche collaborative. Elle a mené à l’émergence d’un projet national structurant dans le domaine des [GMCAO](#) que nous allons décrire dans la section suivante.

2 LABEX CAMI : ENTRE DÉVELOPPEMENT LOGICIEL ET INNOVATION EN RECHERCHE TRANSLATIONNELLE

Le [LabEx CAMI](#) ([LABEX CAMI 2015](#)) se distingue par son approche intégrée. Elle allie recherche fondamentale et applications cliniques concrètes en faisant collaborer toutes les disciplines intervenant dans l’élaboration des solutions technologiques thérapeutiques.

Ce [LabEx](#) représente un carrefour innovant où les avancées technologiques rencontrent les besoins cliniques. Les travaux de cette thèse s’inscrivent ainsi parfaitement dans le contexte du [LabEx](#). Dans cette section, nous examinerons comment le [LabEx CAMI](#) fusionne la recherche avec le développement logiciel. Elle illustre

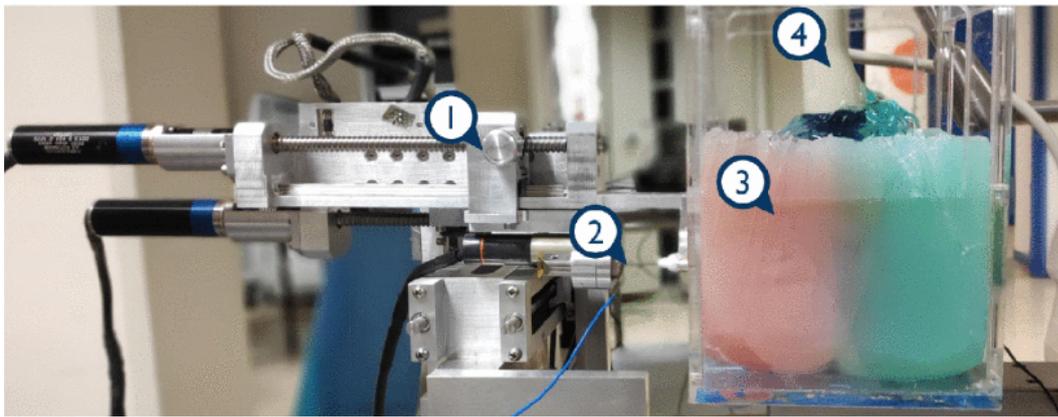


FIGURE 1.9 – Dispositif expérimental. Robot Prosper (1) ; aiguille (2) ; fantôme (3) ; sonde US (4). (LAPOUGE et al. 2021)

son rôle dans l'élaboration de technologies médicales innovantes potentiellement adaptables aux contextes cliniques.

2.1 Présentation du LabEx CAMI

Le **LabEx CAMI** est établi sous l'égide de l'Agence Nationale de la Recherche (ANR) dans le cadre du programme Investissement d'Avenir (projet ANR-11-LABX-0004-01). Le **LabEx** a débuté en 2012 puis a été renouvelé en 2019 pour 5 ans pour se terminer en 2024. Il s'agit d'une collaboration entre six laboratoires de recherche français reconnus pour leurs recherches dans le domaine des GMCAO.

Le laboratoire de « Recherche Translationnelle et Innovation en Médecine et Complexité (**TIMC**) », rattaché à Grenoble INP-UGA, au CNRS et à VetAgro Sup, est un laboratoire de recherche interdisciplinaire en sciences numériques, mathématiques appliquées et sciences du vivant. Spécialisé dans l'étude des processus normaux et pathologiques en santé, il développe des systèmes pour l'aide au diagnostic et à la thérapie.

Le « Laboratoire de Traitement de l'Information Médicale (**LaTIM**) », est une unité mixte de recherche associant l'INSERM, l'université de Bretagne occidentale, l'IMT Atlantique et le CHRU de Brest et mène une recherche multidisciplinaire intégrant les sciences de l'information et les sciences de la santé.

Le Laboratoire de Traitement du Signal et de l'Image (**LTSI**) est un laboratoire de recherche affilié à l'Université de Rennes 1, spécialisé dans le traitement du signal et de l'image pour les applications médicales. Il œuvre dans la recherche interdisciplinaire, combinant les sciences de l'information et la santé, avec un accent sur la technologie médicale innovante.

Le laboratoire « Institut des Systèmes Intelligents et de Robotique (**ISIR**) » est un institut de recherche français situé à Paris, spécialisé en systèmes intelligents et robotique. Relevant de Sorbonne Université et du CNRS, avec une collaboration avec l'INSERM, l'ISIR développe des technologies innovantes dans les domaines de la santé, de l'industrie et des services à la personne.

Le « Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (**LIRMM**) », affilié à l'Université de Montpellier et au CNRS, est un labora-

toire de pointe en informatique, robotique et microélectronique. Il se distingue par ses recherches avancées dans les technologies de l'information et de la communication.

Le « Laboratoire des Sciences de l'Ingénieur, de l'Informatique et de l'Imagerie (ICube) », rattaché à l'Université de Strasbourg, excelle en sciences de l'ingénieur, informatique et imagerie. Il se distingue par son approche interdisciplinaire, liant technologie et applications dans la santé et l'environnement.

Chacun de ces laboratoires apporte une expertise dans le domaine des interventions médicales assistées par ordinateur.

VISION ET OBJECTIFS Les axes principaux d'objectifs du **LabEx CAMI** peuvent se résumer ainsi :

- **Développement technologique** : l'Accent est mis sur l'innovation dans les outils de navigation chirurgicale et les logiciels d'analyse d'images médicales, pour améliorer la précision et la sécurité des procédures chirurgicales.
- **Applications cliniques** : il s'agit ici de transposer des avancées technologiques en solutions pratiques visant à optimiser les soins et améliorer les résultats post-opératoires.
- **Projets intégrés** : cela consiste en la réalisation de projets spécifiques pour proposer des améliorations ciblées dans les pratiques médicales et chirurgicales. Certains de ces projets sont présentés dans la table 1.1.
- **Collaborations transdisciplinaires** : les parties prenantes s'engagent dans des partenariats multidisciplinaires pour combiner les expertises en ingénierie, informatique et médecine.
- **Formation et éducation** : au travers de cet axe, il s'agit de contribuer à l'évolution de la formation médicale par le biais de technologies innovantes basées sur la simulation et la réalité virtuelle.

Cette vision se traduit dans de nombreux projets menés depuis le démarrage du **LabEx CAMI**, chacun animé ou coanimé par un ou plusieurs des six laboratoires impliqués. Ces projets illustrent l'approche interdisciplinaire du **LabEx CAMI**, visant à apporter des solutions novatrices dans différents domaines médicaux.

TABLE 1.1 – Projets emblématiques du **LabEx CAMI**.

Projets & Partenaires	Détails	Applications cliniques	Transfert	Apport innovant
CAMMA LATIM, LTSI, TIMC, LIRM, ISIR 2013	* Analyse et modélisation computationnelles des activités médicales dans les salles d'opération * Développer un système d'acquisition de données multicapteurs	Créer de nouveaux outils d'assistance et de soutien à la décision pour le personnel chirurgical		Système d'acquisition de données multisenseurs pour le bloc opératoire

Continue sur la page suivante

Table 1.1 – suite de la page précédente

Projets & Partenaires	Détails	Applications cliniques	Transfert	Apport innovant
Flexter LIRMM, TIMC, ICUBE 2016	<ul style="list-style-type: none"> * Développement de microrobots pour des tâches non invasives dans le tractus gastro-intestinal * Utilisation de technologies de fabrication additive biocompatibles et des interactions bio-robotiques * Approche visant à créer des microrobots simples, à faible coût, avec un temps court vers les applications médicales 	<ul style="list-style-type: none"> * Collecte d'échantillons du microbiote intestinal * Production contrôlée de protéines anti-inflammatoires * Utilisation du microbiote comme biomarqueur dans le pronostic de pathologies complexes telles que la maladie de Crohn et la rectocolite hémorragique. 	Envisagé (Linksium à Grenoble)	Microrobots intestinaux
Endoscopie Augmentée TIMC, LTSI 2016	<ul style="list-style-type: none"> * Développement de méthodes de suivi des instruments basées sur le traitement d'images et l'apprentissage profond pour améliorer la perception et le raisonnement pendant les procédures endoscopiques * Développement d'un système de vision globale (Global Vision System) avec minicaméras intégrées à l'endoscope classique pour augmenter le champ de vision laparoscopique et améliorer la visualisation de la scène chirurgicale 	Utilisation dans des interventions chirurgicales complexes comme la prostatectomie radicale	Potentiel de transfert industriel identifié	Dispositifs et méthodes permettant la fusion d'images laparoscopiques et ultrasonores

Continue sur la page suivante

Table 1.1 – suite de la page précédente

Projets & Partenaires	Détails	Applications cliniques	Transfert	Apport innovant
SIMPLE LTSI, ISIR, TIMC 2017	<ul style="list-style-type: none"> * Développement de méthodes numériques prédictives et spécifiques aux patients pour simuler des traitements endovasculaires, en utilisant des modèles biomécaniques et des données d'imagerie des patients * Amélioration de la planification des traitements * Optimisation des stratégies interventionnelles en endovasculaire, en anticipant les complications grâce à la simulation 	Réduction des coûts computationnels pour une intégration plus aisée dans le flux de travail clinique	Collaboration avec des entreprises comme ANSYS et Therenva	Simulation de traitements endovasculaires
Cockpit Chirurgical ISIR, ICUBE, LIRMM 2016	<ul style="list-style-type: none"> * Optimisation des interactions entre les dispositifs et les praticiens en chirurgie laparoscopique, par la conception de fonctions adaptatives intégrant les schémas de coordination visuomotrice et les modèles d'apprentissage du chirurgien * Amélioration du confort et de la précision du chirurgien, grâce à la réduction de la fatigue posturale, l'amélioration de la coordination main-œil, et une meilleure perception de la profondeur 	Optimiser les interactions entre dispositifs et chirurgiens	<ul style="list-style-type: none"> * Instruments actifs transférés à un partenaire industriel (Endocontrol) * Dispositifs développés, comme les porte-instruments laparoscopiques, validés expérimentalement et en voie de transfert industriel vers Moon Surgical (précédemment MastOR) 	Innovation dans les interfaces entre chirurgiens et dispositifs médicaux

Ces projets reflètent l'engagement du [LabEx CAMI](#) dans l'avancement de la recherche translationnelle et le développement de solutions innovantes en [GMCAO](#).

L'idée est bien de contribuer à l'évolution des soins au bénéfice du patient. Le domaine dans lequel se prolongent, potentiellement, ces recherches est celui des dispositifs médicaux.

Dans ce contexte, nos travaux sur la maturité des technologies médicales pilotées par logiciel prennent tout leur sens. La nature translationnelle des recherches qui sont menées au sein du [LabEx CAMI](#) montre le besoin de mieux maîtriser et de mieux connaître les technologies développées, en disposant d'outils d'évaluation. En effet, la section suivante montre que l'évolution des technologies médicales pilotées par logiciel vers le statut de dispositifs médicaux intégrés dans un environnement clinique suit un parcours complexe.

3 DU LABORATOIRE VERS LE BLOC OPÉRATOIRE

Comme nous l'avons vu, l'innovation dans le domaine des [GMCAO](#) est fortement influencée par le logiciel, qui est au cœur de nombreux projets de recherche académique et appliquée. La transition de ces projets de la phase de recherche à des applications cliniques ou commerciales exige une conformité réglementaire rigoureuse.

Des normes et des réglementations, comme le règlement européen sur les dispositifs médicaux 2017/745 (EUR-LEX 2017) et les directives du groupe de travail « Logiciel en tant que Dispositif Médical » (Software as Medical Device ([SaMD](#))) du Forum International des Régulateurs de Dispositifs Médicaux ([IMDRF 2016](#)), ont récemment mené au renforcement des exigences pour les logiciels médicaux, même pendant les phases initiales de recherche et de développement.

La recherche sur la maturité technologique du logiciel dans les [GMCAO](#) peut permettre de mieux définir la capacité des prototypes logiciels à répondre aux exigences réglementaires et cliniques futures. Dans le cadre de la réglementation précédemment citée, ces éléments logiciels, lorsqu'ils sont utilisés dans les dispositifs médicaux, qu'ils soient combinés avec du matériel ou considérés comme des dispositifs à part entière, sont soumis à des normes strictes telles que l'ISO/IEC 62304 - Logiciel de dispositif médical - Processus de cycle de vie du logiciel ([ISO/IEC 2006](#)).

Cette norme, lors de la phase d'industrialisation, implique l'adoption de processus spécifiques pour garantir la qualité et la sécurité. Ces processus sont décrits dans le document « ISO 13485 :2016 - Dispositifs médicaux - Systèmes de management de la qualité - Exigences à des fins réglementaires ».

Comme nous le verrons dans le chapitre suivant, des études ont été menées pour apporter une méthodologie à l'implémentation de ces processus afin de faciliter la recherche translationnelle.

4 DÉVELOPPEMENT LOGICIEL ET MATURITÉ TECHNOLOGIQUE DANS LA RECHERCHE FONDAMENTALE

Ce que nous pouvons retenir de cette introduction du contexte est que le développement logiciel peut représenter une part importante des efforts fournis pour la recherche en GMCAO.

À cette étape, il s'agit d'activités de génie logiciel permettant l'exploration des pistes de développements possibles. Progressivement ces développements logiciels nécessiteront d'être évalués de manière à garantir la précision et la robustesse nécessaires pour les applications chirurgicales visées.

Les défis liés à la transformation d'un concept de recherche en Dispositifs Médicaux (DM) sont nombreux et ont un impact significatif sur les délais de mise sur le marché. Avant qu'un DM puisse être introduit sur le marché européen, par exemple, il doit au préalable obtenir le marquage CE, qui indique qu'il est conforme aux exigences fixées par la réglementation européenne.

Des inspections et des tests doivent être effectués pour garantir que le produit est conforme aux exigences. En fonction de la classe de risque du DM, un organisme habilité évalue sa conformité et, en cas d'évaluation satisfaisante, délivre un certificat de conformité, à durée limitée, permettant au fabricant d'apposer le marquage CE sur son produit et de le commercialiser sur le marché européen.

Plusieurs initiatives ont été entreprises pour optimiser cette transformation en prenant en compte ces exigences le plus en amont possible lors de la phase de recherche. Mais ces initiatives s'appuient sur l'intégration de techniques industrielles ou de processus de certification qui peuvent pénaliser la liberté et la flexibilité nécessaires pour développer un concept lors de cette phase de recherche.

Il y a donc un juste équilibre à trouver entre intégrer ces exigences très en amont, et d'autre part ne pas restreindre cette phase pour favoriser les échanges entre scientifiques et médecins, et l'émergence de nouveaux concepts répondant aux besoins cliniques.

Si ces exigences garantissent la sécurité et les bénéfices cliniques visés par le DM, cet équilibre est également à trouver chez les entreprises qui limitent parfois leur effort de R&D en raison de certaines exigences trop contraignantes.

L'objectif principal de nos travaux étant d'optimiser la transformation de logiciels médicaux, nous nous sommes donc concentrés sur les exigences de ces produits logiciels, notamment celles de la norme ISO/IEC 62304, afin d'optimiser cette phase de transformation.

5 ORGANISATION DU MANUSCRIT

Ce manuscrit est structuré en six chapitres principaux, chacun abordant un aspect différent de la recherche sur les technologies médicales pilotées par logiciel et leur évaluation de maturité. Les annexes et la bibliographie complètent ce contenu.

Le **chapitre 2** intitulé « Maturité et systèmes médicaux pilotés par logiciel » aborde le contexte spécifique des dispositifs médicaux pilotés par logiciel au travers du processus de recherche translationnelle. Ce chapitre apporte les fondements bi-

bliographiques qui ont supporté les raisonnements mis en œuvre dans ces travaux de thèse.

Le **chapitre 3** intitulé « Artefacts de génie logiciel et modèle de maturité pour les SdMT » présente le rôle et l'évolution des artefacts de génie logiciel, en discutant de leur genèse, de leur définition et de leur intégration améliorée grâce aux ontologies. En s'appuyant sur une définition dynamique des artefacts, il propose également une méthode de quantification de la maturité des artefacts logiciels dans les SdMT.

Le **chapitre 4** « Évaluation de la maturité des technologies médicales pilotées par logiciel » présente les hypothèses et les méthodes utilisées ainsi que les outils normatifs comme référence pour évaluer la maturité technologique des technologies médicales. Nous expliquons en détail notre méthodologie d'annotation des artefacts de génie logiciel, suivie d'une synthèse de la démarche entreprise.

Dans le **chapitre 5** « Application à un projet de recherche développant une SdMT » une première application de nos résultats est proposée sur un cas concret, le projet FollowKnee. Il décrit la méthodologie d'expérimentation, la collecte des données, l'analyse des résultats et les implications.

Le **chapitre 6** « validation du modèle de Maturité » se concentre sur la validation du modèle de maturité développé au moyen d'outils statistiques. Il discute de la validité des modèles de maturité et présente une validation statistique supportée par une enquête menée auprès d'experts dans les technologies médicales pilotées par logiciel.

Le **chapitre 7** intitulé « Discussion et conclusion » est le dernier chapitre. Il résume les principaux points abordés dans la thèse, discute des travaux futurs et conclut sur les contributions de ces travaux de recherche.

Les annexes comprennent des listings, questionnaires et autres documents de support. La bibliographie regroupe toutes les références citées dans ce manuscrit.

Ce manuscrit permet une compréhension approfondie des enjeux et des défis liés à l'évaluation de la maturité des technologies médicales pilotées par logiciel en phase recherche, en proposant des méthodologies innovantes et applicables dans des contextes de laboratoires de recherche ou départements de R&D.

2

MATURITÉ ET SYSTÈMES MÉDICAUX PILOTÉS PAR LOGICIEL

1	Définition et trajectoire des technologies médicales pilotées par logiciel	21
2	Les parcours de progression des technologies médicales et le concept de Maturité	36
3	Conclusion	44

L'innovation est une alliance entre recherche, marketing, instinct, imagination, produit et courage industriel.

Antoine Riboud

Dans le chapitre 1, nous avons présenté en détail le contexte dans lequel s'inscrivent nos travaux. Comme cela a été dit, le domaine des **GMCAO** fait intervenir un grand nombre de disciplines dans les projets de recherche. Nous nous sommes concentrés sur le domaine du génie logiciel, en particulier, les technologies médicales pilotées par logiciel. Et en avons montrés quelques exemples et applications.

Pour des recherches en **GMCAO**, l'élément logiciel est souvent indispensable et représente le tissu de l'ensemble de ces technologies médicales émergentes, mais il peut aussi être la technologie médicale à lui seul. Il est donc important de maîtriser du mieux possible le développement de ces éléments logiciels. C'est sur cette idée que repose notre raisonnement.

Pour fonder notre approche, nous nous appuyons sur trois piliers que sont les activités de génie logiciel, la maturité technologique et le contexte normatif et réglementaire des technologies médicales. Ces trois éléments agissent de manière forte sur les technologies médicales dès les premiers stades de la recherche.

Ce chapitre présente ces fondements indispensables à nos travaux et les situe dans l'état de l'art.

1 DÉFINITION ET TRAJECTOIRE DES TECHNOLOGIES MÉDICALES PILOTÉES PAR LOGICIEL

1.1 Nos objectifs

Notre objectif est de développer une méthode pour évaluer le niveau de maturité des systèmes de technologies médicales pilotées par logiciel, dès la phase

de recherche en laboratoire. Cela, en tenant compte des impératifs de flexibilité et d'exploration nécessaires à cette phase de découverte.

Nous souhaitons proposer une évaluation préliminaire de la maturité. Nous pensons que cela aidera à guider le développement des dispositifs médicaux du point de vue de l'ingénierie logicielle. Cette évaluation préliminaire est conçue pour préparer la technologie médicale pilotée par logiciel à suivre des processus du type de ceux que l'on trouve en phase industrielle, processus qui permettront de maximiser la sécurité, la conformité et l'efficacité au bénéfice du patient. Cette évaluation préliminaire doit permettre d'évaluer le travail nécessaire pour qu'une technologie médicale pilotée par logiciel intègre un processus sous contrainte réglementaire.

Notre idée repose donc sur le fait qu'en anticipant les exigences des phases réglementées et critiques telles que les essais cliniques ou les étapes de transfert de technologie, nous faciliterons le passage vers ces étapes régulées, tout en évitant de contraindre les acteurs de la recherche avec des considérations industrielles qui ne les concernent pas. La validation de notre hypothèse repose sur plusieurs aspects qui sont présentés dans la suite.

Pour aller plus avant dans les développements, il est important de bien définir ce que sont les technologies médicales dont nous voulons mesurer la maturité. Nous l'aborderons dans la suite de cette section.

1.2 Caractérisation des technologies médicales pilotées par logiciel

Comme nous l'avons vu au chapitre 1, les technologies médicales pilotées par logiciel englobent une vaste gamme de systèmes et il est difficile d'être exhaustif dans leur liste. Ces technologies peuvent intégrer des logiciels avancés dans des dispositifs médicaux très complexes. Leur influence sur les pratiques médicales permet d'accroître la précision et l'efficacité des gestes des praticiens. Ces technologies s'insèrent dans des systèmes de diagnostic automatisés, des dispositifs d'assistance chirurgicale et des systèmes de gestion des données de santé, par exemple. Pour préciser ce que nous entendons par technologie médicale pilotée par logiciel, nous allons rapidement en lister quelques caractéristiques.

ARCHITECTURES LOGICIELLES ET SYSTÈMES DE CONTRÔLE

Les technologies médicales pilotées par logiciel s'appuient sur des architectures logicielles qui doivent être à la fois robustes et flexibles et elles doivent être capables de répondre aux exigences critiques des environnements médicaux. Ces architectures comprennent souvent des couches multiples pour la gestion du matériel, le traitement des données et l'interaction utilisateur. Le développement de telles architectures peut reposer sur des bibliothèques logicielles existantes telles que Robot Operating System (ROS) ¹, *Plus Toolkit* ². Ces bibliothèques fournissent des codes qui peuvent eux-mêmes se retrouver intégrés dans le code de la technologie médicale pilotée par logiciel. Les architectures logicielles des technologies médicales pilotées par logiciel, en multicouches ou modulaires, illustrent bien la nature complexe des technologies et des dispositifs.

1. <https://www.ros.org/>

2. <https://github.com/PlusToolkit>

TRAITEMENT DES DONNÉES ET ALGORITHMES

Le cas des systèmes de diagnostic assisté par ordinateur montre le type de traitement qui peut être fait sur les données numériques liées à l'acquisition, par exemple. De nombreux travaux explorent les aspects algorithmiques qui permettent de réaliser ces traitements. Cela peut être utilisé pour procéder à l'analyse sur des images cérébrales issues d'IRM ou pour construire des modèles précis, mais aussi pour bien d'autres cas et ils sont très nombreux. Ces algorithmes sont de ce fait précieux pour les diagnostics médicaux. Ces technologies logicielles implémentent des algorithmes directement issus des recherches.

Pour être utilisées, ces fonctions algorithmiques et logicielles sont intégrées dans un cadre qui facilite le traitement et l'analyse des données pour les praticiens de santé, comme pour les systèmes « CAD » (Computer Assisted Diagnostic (CAD)). La nature de la technologie médicale pilotée par logiciel peut être de différents types, comme un système avancé de segmentation et d'analyse d'imagerie cérébrale. Il peut être utilisé en environnement clinique pour le diagnostic. Ces algorithmes sont intégrés en tant qu'éléments de dispositifs médicaux.

Les exemples fournis au chapitre 1 et la liste que nous venons de voir suffisent à définir ce que peut être une technologie médicale pilotée par logiciel, ainsi qu'à expliquer comment elle peut être utilisée. Nous pouvons donc avancer une définition qui permettra de fixer le sens de ce terme pour le présent document.

Ainsi, dans le cadre de nos travaux, nous considérons qu'une technologie médicale pilotée par logiciel est :

Tout système ou unité logicielle, intégrés dans un matériel ou non, au stade de la recherche en laboratoire, présent ou qui a vocation à être présent dans une application de santé ou un dispositif médical.

Le domaine de ces technologies est considéré dans le champ des [GMCAO](#). Leur nature peut varier d'un logiciel autonome jusqu'à un système logiciel intégré dans un système matériel.

Avec cette définition, nous sommes à même de caractériser les technologies, mais aussi de mieux les positionner à différents stades de leur évolution. L'environnement où ces technologies peuvent les contraindre à respecter certaines règles liées au cadre réglementaire dans lesquels elles vont être utilisées. La norme ISO/IEC 62304 fixe des règles pour son cycle de vie comme nous le verrons. Tout cela crée un environnement où les dénominations sont également régulées.

Mais avant d'aborder pleinement, les considérations terminologiques, il est important de situer les phases et les étapes communément admises lors du déroulement des processus d'innovation. Pour mieux comprendre comment l'innovation dans les technologies médicales pilotées par logiciel s'articule étape par étape, il est nécessaire de considérer le déroulement de l'innovation médicale en général. C'est ce que nous abordons dans la section suivante sur la recherche translationnelle.

1.3 La recherche translationnelle et les technologies médicales pilotées par logiciel

Comme les [GMCAO](#), la recherche translationnelle est apparue avec l'essor des technologies en suivant l'évolution des savoir-faire. En fait, on pourrait la situer au début du 20^e siècle ([VALENT et al. 2016](#)). C'est à cette époque que Paul Ehrlich a posé les bases de la médecine moderne avec ses travaux sur l'immunologie qui ont conduit à l'ère de la chimiothérapie. Ehrlich a développé le concept de la « magische Kugel » ou « balle magique », une idée selon laquelle on pourrait cibler spécifiquement les agents pathogènes sans nuire à l'organisme hôte. Cette approche a conduit à la création du premier médicament chimiothérapeutique, le salvarsan, utilisé pour traiter la syphilis.

C'est avec le programme Clinical and Translational Science Award ([CTSA](#)) en 2006 que les National Institutes of Health ([NIH](#)) aux États-Unis ont formalisé la recherche translationnelle et rationalisé son processus. Ce programme a permis de mieux intégrer la recherche clinique et fondamentale, ainsi que d'augmenter la probabilité qu'une innovation médicale apparaisse dans l'environnement clinique du patient ([WOOLF 2008](#)).

Ce programme a eu un impact mondial sur la recherche translationnelle et l'organisation des étapes de l'évolution des innovations médicales au cours de ce processus a une large audience.

En effet, la recherche translationnelle constitue une base très importante dans l'évolution de la science médico-chirurgicale. Elle établit un lien direct entre les découvertes scientifiques de base et leur concrétisation dans des applications cliniques. Même si traditionnellement elle est souvent considérée dans le domaine de la recherche sur les molécules et des traitements, elle a maintenant bien intégré les recherches sur les [DM](#) ([FERNANDEZ-MOURE 2016](#)). La recherche translationnelle embrasse tous les domaines de la médecine, que ce soit pour l'élaboration de nouveaux traitements, l'établissement de protocoles de prise en charge novateurs, ou d'autres initiatives visant à améliorer la prévention, le diagnostic et le traitement des maladies.

Le processus qui fonde la recherche translationnelle est rendu possible grâce à une dynamique d'échanges continue entre le monde de la recherche et le milieu médical. Ces notions d'échanges, de continuité et d'itérations sont primordiales, comme nous le verrons plus loin dans ce chapitre.

Dans cette section, nous explorerons de manière approfondie le rôle fondamental de la recherche translationnelle dans l'évolution de la science médicale. Nous mettrons l'accent sur son impact transversal dans tous les domaines de la médecine. Nous détaillerons le processus de recherche translationnelle, en soulignant ses différentes étapes, ainsi que son rôle essentiel dans la garantie d'une réponse efficace aux besoins des patients.

LE PROCESSUS DE RECHERCHE TRANSLATIONNELLE

La recherche translationnelle peut être définie comme le processus dynamique qui intègre les découvertes de la recherche fondamentale dans la pratique clinique et, inversement, qui utilise les observations cliniques pour orienter la recherche fondamentale.

La recherche translationnelle occupe un rôle pivot. FONTANAROSA et al. (2002) soulignent l'importance de traduire les découvertes en biologie moléculaire et génomique en nouvelles approches thérapeutiques. Cette vision est élargie dans les travaux de MAYRINK et al. (2022), qui systématise les études sur les dispositifs médicaux. Mais la mise en œuvre des activités de recherche translationnelle n'est pas évidente et sa complexité ne cesse de croître. Cela est dû aux défis multidisciplinaires de la recherche translationnelle (MAYRINK et al. 2022). HARTL et al. (2021) indiquent que la complexité inhérente à ce domaine est davantage accentuée par l'intégration des technologies de pointe, comme l'intelligence artificielle et la robotique médicale.

Que l'on aborde la recherche sur la génomique, la pharmacologie ou les technologies médicales basées sur l'intelligence artificielle, la recherche translationnelle suit un parcours similaire où la complexité est la brique fondamentale. Ce parcours est fréquemment divisé en plusieurs phases, chacune ayant ses propres caractéristiques. Ces étapes comprennent la recherche préclinique, le passage de la recherche préclinique à la recherche clinique, la validation clinique, et enfin, l'intégration des résultats dans la pratique médicale quotidienne. Voyons maintenant, comment s'articule la recherche translationnelle autour de ces étapes.

UN CADRE POUR LE DÉVELOPPEMENT CLINIQUE ET PRÉCLINIQUE

Le respect des étapes dans la recherche translationnelle permet d'obtenir la rigueur et la reproductibilité nécessaire dans la recherche médicale. Dans WICHMAN et al. (2021), les auteurs soulignent l'importance d'une approche rigoureuse et reproductible pour faire avancer la recherche translationnelle qu'ils appellent Clinical Translational Research (CTR). Pour cela, ils proposent d'utiliser des méthodes scientifiques optimales et d'impliquer les collaborateurs issus de différentes disciplines. Les auteurs définissent la recherche translationnelle comme le processus de transfert des recherches du laboratoire jusqu'au chevet du patient, puis aux communautés, et inversement. Les phases de la recherche translationnelle, de T₀ à T₄ (T pour « translationnel »), varient. T₀ correspond à la recherche fondamentale, tandis que T₄ concerne la traduction de la recherche aux populations ou communautés.

LES PHASES T₀ À T₄ DE LA RECHERCHE TRANSLATIONNELLE

Les phases T₀ à T₄ dans la recherche translationnelle représentent les différentes étapes du transfert des recherches du laboratoire (*bench*) au chevet du patient (*bedside*) et aux communautés. Voici un résumé, de chaque phase :

T₀ : recherche fondamentale. Cette phase se concentre sur la recherche de base, souvent réalisée en laboratoire, et vise à accroître la compréhension générale des processus technologiques, biologiques et des maladies. Dans leurs travaux MAYRINK et al. (2022) indiquent que les étapes finales de la recherche T₀ intègrent déjà des aspects réglementaires, la recherche clinique, l'analyse de marché, le transfert de technologie, la production et l'incorporation de technologies dans le système de santé.

T₁ : Traduction de la recherche fondamentale aux humains. Elle implique l'application des découvertes de la recherche fondamentale à des modèles humains. Il s'agit souvent d'essais cliniques précoces ou d'études sur la sécurité et l'efficacité des traitements. Ces essais peuvent se présenter sous la

forme d'études précliniques sur animaux, études de physiologie humaine, premières études chez l'homme avec des volontaires sains, concept de preuve, et essais cliniques de phase I (évaluation de l'efficacité et de la sécurité d'une intervention biomédicale) (CTSI OF THE UNIVERSITY OF MINNESOTA 2023).

T2 : Traduction des découvertes aux patients. Cette phase concerne l'application des résultats de la recherche fondamentale et des essais cliniques précoces pour développer des traitements ou des interventions qui bénéficient directement aux patients. Selon le « CTSI OF THE UNIVERSITY OF MINNESOTA (2023) » on peut trouver dans cette phase les essais cliniques de phase II (étude de l'intervention biomédicale ou comportementale dans un groupe plus large de personnes) et de phase III (investigation clinique large pour évaluer une intervention expérimentale par rapport à une intervention standard ou contrôlée).

T3 : Traduction de la recherche à la pratique générale. Elle implique l'intégration des découvertes de recherche et des nouvelles thérapies dans la pratique clinique courante, ainsi que l'évaluation de leur efficacité dans des contextes cliniques divers. Plus précisément, le « CTSI OF THE UNIVERSITY OF MINNESOTA (2023) » indique qu'il s'agit généralement d'essais cliniques de phase IV (études après la commercialisation de l'intervention pour surveiller l'efficacité dans la population générale), recherche sur les services de santé, diffusion, communication, mise en œuvre, recherche sur les résultats cliniques.

T4 : Traduction de la recherche aux populations ou communautés. Cette dernière phase se concentre sur l'impact de la recherche et des interventions sur la santé publique et les populations à grande échelle, y compris les études sur les déterminants de la santé et les interventions de santé publique.

Les étapes de la recherche translationnelle montrent que les technologies médicales font l'objet d'investigation tout au long de leur cycle de vie et pas seulement au moment de leur découverte.

Dans le contexte des technologies médicales pilotées par logiciel, la recherche translationnelle revêt une importance particulière. Les logiciels médicaux nécessitent une validation rigoureuse pour évaluer leur précision, leur fiabilité et leur sécurité. La recherche translationnelle implique cette validation en fournissant un cadre pour permettre l'évaluation de la performance clinique des logiciels médicaux.

Ce cadre implique un grand nombre d'acteurs qui interviennent collaborativement tout au long des quatre grandes étapes décrites plus haut. Les collaborations, les retours sur les avancées et les itérations réalisées dans les activités sont essentiels dans le développement des innovations médicales et tout particulièrement pour les technologies médicales pilotées par logiciel.

Ainsi, la recherche translationnelle joue un rôle fondamental dans le développement et l'évaluation des technologies médicales pilotées par logiciel. Elle structure la progression de la recherche médicale de manière significative en concrétisant les étapes de transition des découvertes fondamentales vers des applications pratiques bénéfiques pour les patients. Elle place la vision clinique au centre du processus de recherche translationnelle.

Nous venons de voir l'articulation des grandes étapes qui guident ces innovations médicales. Ces étapes sont suivies dans une grande partie du monde qui innove pour produire de nouvelles technologies au bénéfice du patient. Elles permettent de gérer la complexité de ce type d'innovations, mais nécessitent une mise en application dans divers domaines. Ces domaines vont des laboratoires de recherche aux environnements cliniques, en passant par les systèmes de conception et de production industriels.

Dans le cadre de ces étapes, un ensemble spécifique de termes émerge. Ces termes fondent une réglementation stricte qui, tout en encadrant le développement des technologies, est essentielle pour assurer la fiabilité et la sécurité des soins aux patients.

Ce contexte montre que le processus d'innovation dans les technologies médicales implique, à toutes les étapes, de nombreux acteurs différents des informaticiens qui sont des spécialistes du génie logiciel.

Nous allons voir dans la suite de cette section, comment peuvent se préciser ces processus. Nous aborderons cette complexité qui se met en place avec l'apparition progressive de la réglementation dans les activités techniques. Nous verrons que dès les laboratoires de recherche des solutions sont proposées pour faciliter la gestion de cette complexité et permettre à la technologie de progresser vers son utilisation en environnement clinique.

1.4 Processus et terminologie des dispositifs médicaux

La terminologie joue un rôle important dans le cadre de notre recherche. Elle permet de caractériser et de situer dans le temps la technologie que nous considérons. En analysant les technologies médicales pilotées par logiciel sur la durée des activités, il nous est possible de mieux comprendre les contraintes qui peuvent s'imposer à ces technologies et d'adopter une réflexion proactive.

Cette réflexion a permis à de nombreux auteurs d'envisager l'enchaînement des activités d'innovation dans le domaine des technologies médicales du point de vue des processus.

LE PROCESSUS D'INNOVATION

Dans leur ouvrage, Yock et al. (2015) proposent un processus standard de l'innovation pour les technologies médicales, il est illustré par la figure 2.1 qui est tirée du livre. Leurs travaux sont une référence en la matière, ils proposent l'identification de toutes les étapes qui mènent à lancer la commercialisation d'un dispositif médical. Ils présentent les trois grandes phases et les six grandes étapes qui constituent le processus et qui peuvent se résumer comme suit :

Phase 1 : identifier (Identify)

Étape 1 : Recherche de besoins il s'agit de l'identification et de l'exploration des besoins cliniques non satisfaits.

Étape 2 : Tri des besoins Il s'agit de la sélection et de la priorisation des besoins pour déterminer lesquels offrent les meilleures opportunités de développement.

Phase 2 : inventer (Invent)

Étape 3 : Génération de concepts il s'agit ici de la génération de diverses solutions potentielles pour répondre aux besoins cliniques identifiés.

Étape 4 : Évaluation des concepts il s'agit de l'évaluation des concepts générés pour sélectionner ceux qui sont les plus prometteurs en termes de faisabilité technique, réglementaire et commerciale.

Phase 3 : implémenter (Implement)

Étape 5 : Développement de stratégies il s'agit du développement de stratégies pour la propriété intellectuelle, la recherche et développement, les essais cliniques, la réglementation, la gestion de la qualité, le remboursement, le marketing, les ventes et la distribution.

Étape 6 : Planification commerciale il s'agit de créer un plan d'affaires intégrant toutes les stratégies, préparation pour la commercialisation et la mise sur le marché.

Cette présentation qu'ils font du processus d'innovation dans les technologies médicales peut nous servir de repère pour préciser notre positionnement.

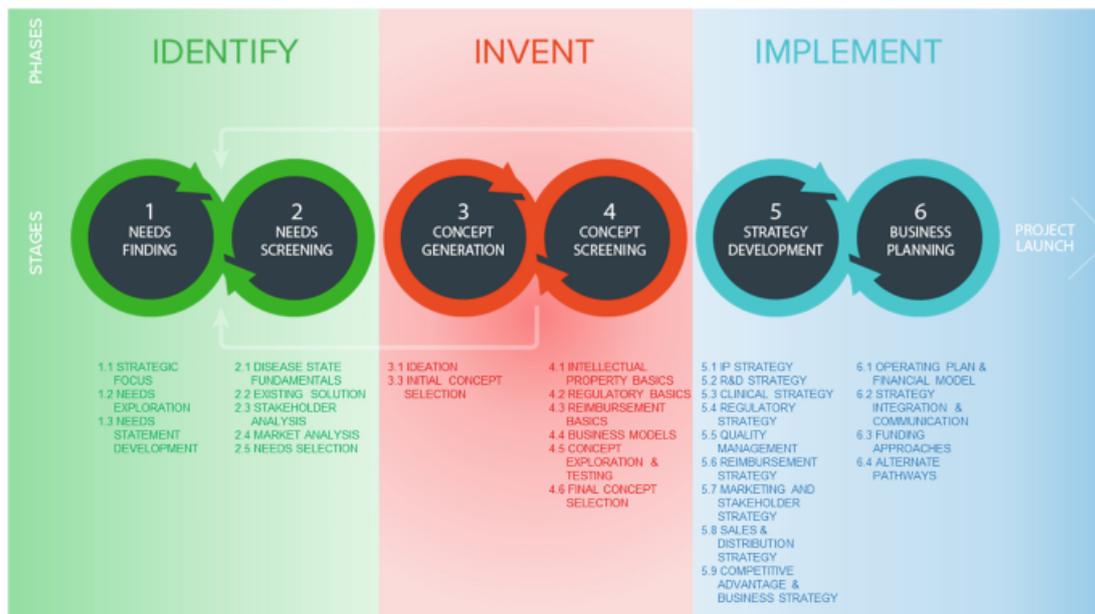


FIGURE 2.1 – Dans leur proposition, Yock et al. (2015) indiquent un processus en 3 phases et 6 étapes, de l'invention jusqu'à la commercialisation, qui est couvert par 25 activités.

Ce processus, proposé par Yock et al. (2015), couvre un grand nombre d'activités dont la plupart dépassent l'étendue de nos travaux. Les activités de recherche technique sont placées dans les étapes de l'*invention* et de R&D dans l'étape d'*implémentation* de nouvelles technologies médicales.

- **Étape d'Invention**

- **Idéation** - Conception préliminaire et simulation.
- **Sélection initiale des concepts** - Évaluation de la faisabilité technique des concepts.

- **Exploration et tests des concepts** - Tests approfondis pour évaluer la performance, la durabilité et l'adéquation aux besoins.
- **Étape d'Implémentation**
 - **Stratégie de R&D** - Développement technique et optimisation des produits.
 - **Gestion de la qualité** - Développement de protocoles techniques pour assurer la qualité et la conformité du produit.

Dans la réalité, les limites ne sont pas si claires, mais ce déroulement illustre bien l'enchaînement des diverses activités.

Plusieurs étapes techniques clés sont réalisées en laboratoire. Lors de la sélection initiale des concepts, des expérimentations préliminaires testent la faisabilité technique des idées à travers des prototypages et des évaluations fonctionnelles. Cette phase est suivie par une exploration et des tests plus approfondis où les prototypes subissent des essais rigoureux pour valider leurs performances et leurs conformités aux exigences cliniques et réglementaires. En parallèle, la stratégie de recherche et développement intègre le développement de procédés de fabrication et l'optimisation des produits. Elle prend en compte l'ajustement basé sur les retours des essais cliniques. En progressant dans ces activités techniques, les considérations réglementaires entrent en jeu pour garantir l'intégration des technologies de manière sûre et fiable pour le patient.

Cette vision est une application du processus de recherche translationnelle selon un point de vue « économique et juridique » et fait donc une description des activités plutôt orientées dans ce sens. Nous ne prenons pas en considération les aspects économiques et juridiques de la trajectoire d'une technologie médicale pilotée par logiciel. Les activités que nous considérons pour élaborer notre approche sont intégrées dans les étapes d'invention et d'implémentation.

La plupart du temps, pendant les projets de recherche en [GMCAO](#), les parties prenantes séparent les considérations techniques de la technologie médicale des considérations économiques que peut susciter un projet de transfert. Ces considérations économiques et juridiques font l'objet d'autres types de projets et de financements.

Nous verrons plus loin dans cette section, qu'il existe plusieurs approches, moins influencées par une vision économique et juridique, pour caractériser la progression d'une technologie médicale vers le patient, elles-mêmes pouvant intégrer cette démarche d'innovation.

Ainsi, pour cette étude, nous considérons globalement trois phases principales du processus d'innovation selon lesquelles les technologies peuvent évoluer en fonction des choix qui sont faits pendant le projet de recherche. La succession de ces trois grandes phases est illustrée par la figure [2.2](#).

La première phase débute avec une nouvelle idée de technologie. C'est ce que nous appelons la phase de recherche ou la phase de recherche en laboratoire. Elle correspond à la phase d'invention, telle que présentée par Yock et al. (2015). Cette phase dont on ne traite que des activités techniques se termine lorsque les activités techniques d'industrialisation commencent.

Nous appelons cette deuxième phase, la phase d'industrialisation. Elle couvre toutes les activités de conception et de fabrication du futur produit et les travaux

réalisés pour sa mise sur le marché. Elle a pour objet principal de garantir que le produit résultant est conforme à toutes les règles d'assurance qualité en minimisant le risque lié à son utilisation médicale. Là encore, nous n'entrons pas dans des activités autres que techniques.

La troisième phase pourrait être intégrée dans la deuxième de notre point de vue. Il s'agit de la phase de commercialisation. Elle regroupe des activités qui sont majoritairement économiques et juridiques, mais aussi des activités techniques plus orientées sur la maintenance. Nous ne considérons pas non plus de ressources économiques ou financières qui viennent en soutien de ces projets sur les trois phases que nous proposons.



FIGURE 2.2 – Les trois grandes étapes dans un trajet possibles pour les technologies médicales pilotées par logiciel. Notre étude couvre les activités menées au laboratoire et le démarrage de celles menées par les équipes de R&D dans la phase industrielle.

Nos travaux eux-mêmes se concentrent uniquement sur la première phase et quelques activités liées au processus de transfert de technologie qui peuvent intervenir au démarrage de la deuxième phase avec les équipes de R&D, comme le montre la figure 2.2. Durant ces étapes, divers événements viennent impacter les activités de développement.

Notamment les premiers essais cliniques, dans ce cas un projet de recherche peut se trouver dans la nécessité d'établir un dossier de demande d'autorisation auprès de l'Agence Nationale de Sécurité du Médicament et des produits de santé (ANSM)³, en vue d'obtenir une autorisation pour réaliser ces essais cliniques. Cette étape comporte bien des difficultés, même si des aides existent pour permettre l'établissement de tels dossiers, auprès de l'ANSM ou comme le guide du parcours des dispositifs médicaux en France (HAUTE AUTORITÉ DE SANTÉ 2021) qui est établi par la Haute Autorité de Santé (HAS).

C'est à ce moment-là que le projet de recherche doit prendre en compte l'environnement réglementaire et sa terminologie. Il est alors confronté, non seulement au contexte réglementaire qui s'impose, mais aussi à la terminologie qu'il emploie.

LA TERMINOLOGIE

En effet, malgré la production croissante de guides restrictifs et de normes, il existe encore de nombreuses définitions non consensuelles pour décrire ce qu'est un logiciel dans le domaine médical.

Pour distinguer les logiciels médicaux des autres dispositifs médicaux, la Food and Drug Administration (FDA) et les NIH ont introduit le concept de « Santé numérique » ou « Digital Health ». Un moment important dans cette évolution a été l'introduction du « Digital Health Innovation Action Plan » par la FDA en 2017. Ce

3. <https://ansm.sante.fr/>

plan avait pour objectif de faciliter et d'accélérer le développement de produits de santé numériques. Il était alors nécessaire de mieux définir l'objet des politiques de réglementation.

Cela a été fait en incluant de nouvelles définitions dans le lexique réglementaire (IMDRF 2013) et en introduisant le terme **SaMD**. La définition (FDA (2018) et IMDRF (2013)) indique que les **SaMD** sont indépendants de tout autre dispositif (logiciel autonome) et doivent être distingués de deux autres types 1) les logiciels intégrés ou composants d'un dispositif médical (appelés Software in Medical Device (**SiMD**)) et 2) les logiciels utilisés dans la fabrication ou la maintenance d'un dispositif médical.

Cependant, ces définitions ne font pas l'unanimité. En particulier pour les **SaMD**, comme le mentionnent HERMON et al. (2021). Ce terme ne possède pas de frontières bien définies et sa définition demeure ambiguë.

Le terme Medical Device Software (**MDS**) que l'on trouve dans la norme ISO/IEC 62304 définit une gamme plus large de dispositifs, qu'il s'agisse de logiciels autonomes ou de parties d'un dispositif matériel. Aujourd'hui, les termes **MDS** et **SaMD** sont utilisés indifféremment, ce qui montre que la définition n'est toujours pas fixée dans les esprits.

Au niveau des normes internationales, les logiciels définis par le terme **MDS** sont couverts par la norme ISO/IEC 62304, qui englobe tous types de logiciels associés à des **DM**, qu'ils soient intégrés ou non. En revanche, la norme ISO/IEC 82304, publiée en 2016, est dédiée aux logiciels de santé autonomes, connus sous le nom de **SaMD**. Cette norme ne s'applique pas aux logiciels intégrés aux dispositifs médicaux.

Pour nos travaux, nous fondons notre terminologie sur le travail de Gordon et Stern (GORDON et al. 2019), qui semble mieux adapté à l'objet de notre étude. Et nous considérerons les technologies médicales pilotées par logiciel associées aux **DM**, en référence à la norme IOS/IEC 62304.

En effet, le point de vue de GORDON et al. (2019) intègre une notion juridique en accord avec la réglementation (EUR-LEX 2017) et une meilleure délimitation des dispositifs. Ils ont introduit le terme Software-driven Medical Device (**SdMD**), qui englobe les **SaMD** et tous les autres aspects de la « santé numérique » tels que le matériel avec d'importants composants logiciels (**SiMD**), y compris les applications mobiles ou les appareils sans fil. Comme les termes, **MDS** ou **SdMD** incluent tous deux le terme « Dispositif médical », ils impliquent que l'objectif médical prévu, c'est-à-dire la pathologie ou le trauma pour lequel le logiciel sera utilisé (REGULATION 2022), a été défini et fixé.

L'USAGE MÉDICAL PRÉVU

Il nous faut nous arrêter sur cette définition de l'« Usage Médical Prévu » ou « Medical Intended Purpose ». Son importance est grande et il doit être mieux défini.

Dans le contexte réglementaire des dispositifs médicaux, les termes *usage médical prévu* et *intended purpose* sont souvent utilisés de manière interchangeable, bien qu'ils puissent présenter de subtiles distinctions selon la réglementation et les régions géographique.

En Europe, le Règlement sur les dispositifs médicaux (Medical Device Regulation (**MDR**)) utilise ces termes pour décrire les objectifs que le fabricant attribue

au dispositif, notamment en ce qui concerne le diagnostic, le traitement ou la prévention des pathologies. Cette définition englobe également l'utilisation spécifique selon l'anatomie ou un processus physiologique que le dispositif médical vise à modifier ou à maintenir.

Aux États-Unis, la distinction entre *intended use* et *intended purpose* est parfois plus marquée. *Intended use* fait référence à l'utilisation générale du dispositif, comme défini par les déclarations du fabricant et par l'étiquetage, tandis que le terme *intended purpose* désigne plus spécifiquement les effets ou les actions que le dispositif est censé avoir selon l'étiquetage. Le fabricant doit définir clairement et justifier l'utilisation prévue du dispositif à travers des données cliniques. Il doit respecter les processus de conformité réglementaire pour cette utilisation.

L'usage médical prévu d'un dispositif médical est étroitement lié à la notion de risque dans la réglementation des dispositifs médicaux. Ce lien influence la classification de risque du dispositif, qui détermine les exigences de contrôle réglementaire avant sa mise sur le marché.

En effet, les dispositifs sont classés selon leur niveau de risque, avec des procédures de conformité variant de l'autocertification pour les dispositifs à faible risque à des évaluations plus rigoureuses pour ceux à haut risque. Cette évaluation prend en compte les données cliniques justifiant l'usage prévu et surveille le dispositif post-commercialisation pour assurer la sécurité et l'efficacité continue. Ainsi, l'usage médical prévu oriente toutes les étapes du cycle de vie réglementaire du dispositif, affectant le développement, la classification et la surveillance après commercialisation. C'est donc après avoir fixé l'usage médical prévu que l'on peut concrètement définir le risque lié à cet usage dans le cadre médical.

Cette dernière remarque est très importante pour nos travaux, car il implique que nous ne pouvons utiliser ni le terme **SdMD** ni le terme **MDS**. Au lieu de cela, et comme l'objectif médical prévu peut ne pas être entièrement défini ou fixé pendant la phase de recherche, nous avons donc introduit l'expression « Technologie médicale pilotée par logiciel » (Software-driven Medical Technology (**SdMT**)) pour englober tous les prototypes technologiques en cours de développement pendant la phase de recherche. Ce terme et ce qu'il représente comme nous l'avons vu plus haut sont valables jusqu'à ce que l'usage médical prévu soit clairement défini et que la notion de risque soit intégrée.

Cette notion d'usage médical prévu est très importante dans le domaine des **DM**. Il l'est tout autant pour les **SdMD** ou les **SaMD**. La réglementation est très exigeante quant à ce qu'implique l'usage que l'on prévoit pour un **DM**. Si ces préoccupations ne manquent pas de créer des difficultés potentielles aux futurs dispositifs médicaux, il en existe déjà qui entravent le passage des **SdMT** vers la phase de transfert. Nous allons nous pencher maintenant sur ces difficultés.

1.5 Problématiques du transfert clinique

De nombreux auteurs ont menés des réflexions pour imaginer un processus qui facilite le transfert d'une **SdMT** ou plus largement d'une technologie médicale.

UN GRAND NOMBRE DE PROJETS QUI NE BÉNÉFICIE PAS AU PATIENT

FICHTINGER et al. (2022) montrent, dans leur article, que sur 58 projets seulement 11

d'entre eux ont obtenu le marquage dans l'Union européenne ou aux États-Unis⁴. Quelques auteurs ont tenté de répondre à la question, « pourquoi aussi peu de technologies médicales arrivent auprès du patient? ».

Les travaux de BLEVINS et al. (2018) vont dans ce sens. Leur article est basé sur l'expérience accumulée au « Stanford Byers Center for Biodesign », un centre reconnu pour son expertise dans le domaine de l'innovation biomédicale. Leur proposition est d'explorer les raisons pour lesquelles de bonnes idées dans le domaine de l'innovation technologique médicale échouent souvent à se concrétiser en produits sûrs, fiables, adoptables et économiquement viables.

BLEVINS et al. (2018) souhaitent montrer comment des idées prometteuses peuvent se perdre en cours de route en raison de divers obstacles et erreurs courantes dans le processus d'innovation. Les auteurs ont cherché à faire ressortir les facteurs clés de succès et d'échecs et ainsi, aider à accroître la probabilité de succès des futures innovations dans les technologies médicales.

Pour cela, BLEVINS et al. (2018) ont proposé un examen détaillé des pratiques courantes qui entravent le succès et mettent en lumière les principes fondamentaux pour renforcer les chances de réussite dans l'innovation médicale.

BLEVINS et al. (2018) insistent sur le fait que la transition d'une idée novatrice à un produit médical sécuritaire et fiable est extrêmement difficile et incertaine. Ils montrent que cette transition est caractérisée par plusieurs hypothèses et suppositions clés 1) un « dérisquage » progressif de la technologie, 2) la différence entre invention et innovation, 3) le rôle essentiel de l'écosystème d'innovation.

Tout d'abord, le « dérisquage », selon les auteurs, est permis par le fait de mettre en place dès les premières étapes un « processus délibéré ». Il s'agit d'un processus méthodique et réfléchi dans le développement de l'innovation. Ce « processus délibéré » doit permettre une évolution progressive vers cette « sécurisation » de l'innovation. Ils indiquent qu'il s'agit là d'un processus itératif et non linéaire.

Toutefois, il faut ici détacher la notion de risque apportée par les auteurs de celle dont nous avons parlé dans le cadre des DM. Dans le contexte de leurs travaux, le « dérisquage » est abordé comme un effort pour minimiser les chances d'échec. C'est en suivant une approche réfléchie et progressive dans le développement de l'innovation que les incertitudes seront systématiquement réduites, à chaque étape du chemin.

L'importance d'un processus délibéré est essentielle pour réduire les risques et soutenir la capacité d'invention, définie par BLEVINS et al. (2018) comme la création d'idées ou de prototypes nouveaux. L'innovation, quant à elle, transforme ces inventions en produits ou services répondant à un besoin réel. Cette transformation implique le raffinement, la validation de marché, la production de masse, la distri-

4. Le marquage est obligatoire pour accéder au marché.

- Union européenne :
 - Règlement (UE) 2017/745 relatif aux dispositifs médicaux (MDR - Medical Device Regulation).
 - Directive 93/42/CEE (remplacée par le MDR).
- États-Unis :
 - Code of Federal Regulations (CFR) Title 21.
 - FDA's Center for Devices and Radiological Health (CDRH) directives.

bution et l'adoption par l'utilisateur final, un processus bien décrit par Yock et al. (2015) pour les technologies médicales.

Ainsi, dans leur contribution BLEVINS et al. (2018) donnent les « bonnes pratiques » et mettent en lumière qu'il faut mettre en œuvre un « processus délibéré ». Ils indiquent qu'il doit être progressif, itératif et non linéaire. Leur étude va au-delà des seules considérations techniques, mais on peut clairement considérer qu'elle les inclut. Cette voie qu'ils dessinent peut sans problème s'appliquer aux développements techniques, mais il faut caractériser les difficultés dont on parle.

DES PROPOSITIONS POUR FACILITER LE PASSAGE

L'article de FICHTINGER et al. (2022) met en lumière les défis inhérents au transfert des SdMT, spécifiquement dans le contexte des systèmes robotiques guidés par image. Les auteurs identifient plusieurs obstacles critiques. Ils vont des barrières réglementaires et financières à la complexité des intégrations technologiques, qui freinent la transition de la recherche à la pratique clinique. Cette analyse permet de mieux comprendre les difficultés rencontrées dans le domaine plus large des SdMT. Des propositions ont été faites pour permettre de surmonter ces difficultés.

Dans une de ses contributions, Mc CAFFERY et al. (2009) mettent en évidence que les défis principaux du transfert clinique des technologies médicales incluent non seulement la complexité intrinsèque des logiciels, mais aussi la rigueur nécessaire dans la gestion des risques. Il indique que cette rigueur doit être considérée tout au long du cycle de vie du développement du logiciel et vérifiée par les équipes de R&D. Les réglementations actuelles imposent aux fabricants de dispositifs médicaux de prouver cette gestion minutieuse des risques. Cela doit être fait de l'analyse des dangers jusqu'à la gestion opérationnelle des risques. C'est de cette gestion obligatoire que ressort une grande partie de la difficulté et c'est ce qui renforce la complexité du processus de transfert.

C'est pourquoi les équipes issues des laboratoires de recherche se retrouvent en difficulté quand il s'agit d'adapter la technologie pour lui donner une chance de passer vers la phase industrielle. Même en bénéficiant des aides et des soutiens nécessaires, cette difficulté persiste. Elle est souvent désignée comme la « vallée de la mort » et ressentie comme telle par les ingénieurs, chercheurs et autres personnels du domaine de la recherche. SEYHAN (2019) décrit la « vallée de la mort » comme la phase critique où de nombreuses découvertes précliniques échouent à se traduire en applications cliniques efficaces, en raison de hauts taux d'échec et de faible rentabilité des investissements en recherche.

La contribution de Mc CAFFERY et al. (2009) apporte une aide au démarrage de la phase industrielle. Ils soulignent que l'adaptation des modèles de gestion des risques aux exigences spécifiques des logiciels médicaux est une nécessité importante et certains modèles peuvent permettre de réussir cette gestion. Par exemple, comme cela peut être proposé par le modèle d'intégration de la maturité des capacités (Capability Maturity Model Integration (CMMI)).

En effet, CMMI est un modèle de développement qui guide les entreprises dans l'amélioration de leurs processus. Il a été développé à l'origine par le Software Engineering Institute (SEI) de l'université Carnegie Mellon. Ce modèle aide les organisations à évaluer leurs capacités à identifier les améliorations potentielles et à intégrer méthodiquement ces améliorations dans les processus existants. Le modèle est ap-

plicable à divers domaines, tels que le développement de logiciels, la production et les services. En fournissant un cadre structuré pour la gestion de processus, CMMI aide les entreprises à atteindre des niveaux de maturité supérieurs.

Toutefois, Mc CAFFERY et al. (2009) montrent que le CMMI n'est pas parfaitement adapté pour servir de modèle de gestion des risques dans le cadre des logiciels médicaux. Ils indiquent qu'il est nécessaire de combler ses carences en introduisant la notion de risque et proposent le Risk Management Capability Model (RMCM). Il est conçu pour répondre spécifiquement à ces exigences, en réduisant l'écart entre les normes de maturité des processus et les besoins uniques de l'industrie des dispositifs médicaux.

Mais cela ne règle pas le problème de la vallée de la mort, car ce type d'initiative sous-entend une certaine habitude de la gestion des processus et est proposée pour les équipes de R&D dans la phase post-transfert. Pour une équipe de laboratoire qui souhaite transférer sa technologie, ce type de méthodologie semble encore très difficile à mettre en œuvre.

La maîtrise des processus de développement est impérative pour faciliter l'adoption des cadres réglementaires stricts. Les travaux de Rust et al. (2016a) démontrent que l'élaboration d'un plan de développement logiciel conforme à la norme ISO/IEC 62304 est complexe, surtout pour les organisations novices dans ce domaine réglementé.

Les feuilles de route de mise en œuvre et les modèles de développement logiciel standardisés, tels que ceux discutés dans Rust et al. (2016b), offrent des ressources pour les entreprises. Mais elles sont difficilement applicables à la transition en fin de phase de recherche. Avant de pouvoir entrer dans ces systèmes d'aides que proposent Rust et al. (2016b), il est nécessaire d'avoir mené à bien un grand nombre d'activités d'adaptation pour mettre en place des processus qui n'existent pas dans la phase de recherche.

Bien sûr, ces outils qui guident les entreprises à travers les multiples exigences réglementaires sont importants. Ils renforcent également leur capacité à développer des logiciels sûrs et fiables. Ce que font ces entreprises dans leur parcours, en utilisant ces outils, est de progressivement augmenter la maturité de leur technologie en s'attaquant aux complexités des normes réglementaires qui s'imposent à elles.

Ainsi, en droite ligne avec les préconisations de BLEVINS et al. (2018) en ce qui concerne le concept de « processus délibéré », l'utilisation de plans de développement logiciel et de feuilles de route de mise en œuvre standardisée atténue les obstacles. Pour les entreprises dans le domaine des dispositifs médicaux, cela permet d'avancer progressivement vers le niveau de maturité souhaité. Nous avons vu que cette maturité peut être soutenue par des outils et des processus structurants dans les premières étapes du processus industriel, comme le résume la table 2.1.

Auteurs	Recommandations
RUST et al. (2016a)	Développement d'un modèle de planification du développement logiciel pour aider les organisations à répondre aux exigences de la norme IEC 62304.
RUST et al. (2016b)	Utilisation de modèles de design pour créer des artefacts "How-to" afin d'aider les PME à démontrer la conformité avec la norme IEC 62304.
BLEVINS et al. (2018)	Création d'une culture d'innovation et d'un engagement systématique envers le transfert technologique pour augmenter la probabilité de succès dans l'innovation médicale.
SEYHAN (2019)	Adoption de modèles d'innovation ouverte et collaboration renforcée entre les chercheurs, l'industrie et les régulateurs avec un accroissement de la transparence des données et l'optimisation des processus décisionnels
MC CAFFERY et al. (2009)	Adoption des modèles de gestion des risques qui intègrent des pratiques spécifiques au domaine médical, comme le RMCM .
FICHTINGER et al. (2022)	Établir des partenariats étroits entre développeurs, régulateurs et utilisateurs cliniques et promotion des programmes de formation complets pour les utilisateurs finaux.

TABLE 2.1 – Solutions proposées pour le transfert et l'implémentation des technologies médicales selon différentes approches.

Nous voyons ici que, la complexité apportée par les aller-retour dans l'innovation rend difficile la planification et rend difficile la mise en place de processus dans la recherche pour qui le transfert apparaît comme un mur réglementaire à franchir.

La section suivante présente ce concept de maturité en la positionnant dans le cadre de la réglementation qui s'impose aux logiciels médicaux.

2 LES PARCOURS DE PROGRESSION DES TECHNOLOGIES MÉDICALES ET LE CONCEPT DE MATURITÉ

La maturité au sens large recouvre un grand nombre de concepts et il n'est pas pertinent de tous les aborder dans le cadre de ces travaux. Comme nous l'avons vu plus haut, notre objet se situe essentiellement sur les concepts techniques. Bien sûr, d'autres aspects influent sur la maturité d'une technologie et nous aborderons

ceux pour lesquels cette influence sur l'évolution vers le transfert est la plus directe.

2.1 Le parcours de la technologie médicale

Ce concept de maturité technologique est un indicateur très utilisé pour évaluer le degré de développement et de fiabilité d'une technologie. Selon MANKINS (1995), les niveaux de maturité technologique (Technology Readiness Level (TRL)) sont utilisés pour systématiser et analyser le risque lié à l'implémentation de technologies nouvelles ou améliorées.

Il définit les niveaux TRL de la manière suivante :

- TRL 1 : Observation et rapport des principes de base.
- TRL 2 : Formulation du concept technologique et/ou de l'application.
- TRL 3 : Preuve de concept critique analytique et expérimentale.
- TRL 4 : Validation de composants et/ou d'un prototype de base dans un environnement de laboratoire.
- TRL 5 : Validation de composants et/ou d'un prototype de base dans un environnement pertinent.
- TRL 6 : Démonstration d'un modèle de système ou de sous-système dans un environnement pertinent.
- TRL 7 : Démonstration d'un prototype de système dans un environnement spatial.
- TRL 8 : Système réel complété et qualifié pour le vol.
- TRL 9 : Système réel prouvé en vol par des opérations de mission réussies.

Cette approche est particulièrement pertinente dans le domaine des SdMT.

Aujourd'hui, nous utilisons les TRL dans le langage courant et de manière générale, ils ne correspondent plus exactement à la définition de MANKINS (1995). Les TRL sont définis en plusieurs étapes que nous pouvons grossièrement établir de la manière suivante. Elles commencent par la preuve de concept, couverte par les niveaux TRL 1 à 3. Ils sont suivis de la validation en laboratoire couverte par les niveaux TRL 4 à 5. Et, finalement, la démonstration en environnement réel qui fait l'objet des niveaux TRL 6 à 9, avant la commercialisation. Ces étapes permettent aux parties prenantes de suivre le progrès technologique de manière structurée. Grâce à ces niveaux, il est ainsi possible de minimiser les risques associés à l'adoption de nouvelles technologies. On retrouve ici la notion de « dérisquage », liée au processus délibéré, avancée dans les travaux de BLEVINS et al. (2018).

Dans le cadre particulier des dispositifs médicaux, la contribution de MOREAU-GAUDRY et al. (2010) montre la progression de l'idée jusqu'à l'industrialisation. MOREAU-GAUDRY et al. (2010) mettent en évidence le cycle Concept, Recherche, Essais, Produit, Soins (CREPS) pour le développement des innovations technologiques en santé. Ils montrent l'importance de la collaboration entre différents acteurs : Industriels (I), Chercheurs (R), Patients (P), et les professionnels de soins (S). La figure 2.3 illustre l'évolution de l'innovation de la technologie médicale. Cette progression met en évidence la possibilité d'itérer dans le cycle proposé et donne une idée d'un processus non linéaire.

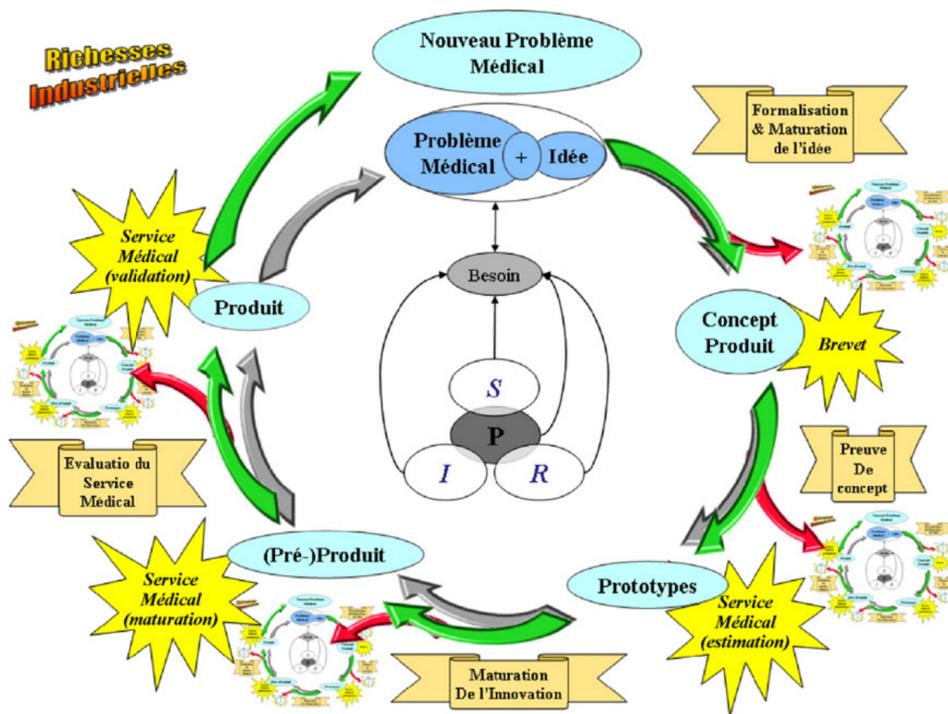


FIGURE 2.3 – La progression de l'idée vers le patient MOREAU-GAUDRY et al. (2010)

Cette description se rapproche des préconisations qui ont été proposées par BLEVINS et al. (2018) et qui indiquent l'importance d'un processus non linéaire et itératif. MOREAU-GAUDRY et al. (2010) s'en rapprochent davantage encore dans la place qu'ils donnent à la collaboration entre les acteurs et qu'ils mettent au centre de leur approche. Ils font de ces collaborations intersectorielles un élément incontournable de l'innovation dans le domaine des DM. Ces collaborations rendent plus efficaces l'intégration de tous les aspects qu'apporte un DM. Ici, les auteurs rendent plus concrètes ces collaborations au travers des dynamiques engendrées par les Centres d'investigation clinique – Innovation technologique (CIC-IT). Et elles font intervenir les quatre grands secteurs responsables du développement des DM :

- I - Industriels : les industriels transforment une innovation technologique en un produit commercialisable. Ils sont impliqués dans la production, la qualité et la distribution. Ils prennent également en charge la gestion des aspects réglementaires pour la mise sur le marché.
- R - recherche (chercheurs) : les chercheurs apportent l'expertise scientifique et technique nécessaire pour développer des innovations. Ils interviennent le plus souvent sur la conception, les développements préliminaires, les essais, et l'évaluation des performances et de la sécurité des innovations.
- P - Patients : les patients sont au centre du développement des innovations en santé. Ils sont les bénéficiaires finaux de ces technologies. Leur retour, notamment au travers des associations de patients, est essentiel pour ajuster les innovations à leurs besoins réels. Cela permet d'évaluer leur efficacité et leur acceptabilité et de valider l'utilité et l'efficacité des produits.
- S - Soins : les professionnels de soins qui utilisent pratiquement les innovations dans le contexte clinique. Ils permettent l'évaluation de l'impact réel des innovations sur les soins aux patients.

Dans la partie droite de la figure 2.3, on peut identifier le développement en laboratoire. On voit, avec l'évolution vers la partie gauche comment les secteurs de la recherche, industriel et réglementaire s'interpénètrent pour arriver au produit DM en créant une vraie valeur pour le patient qui est placé au centre du processus.

Ainsi, la gestion de la maturité technologique est donc essentielle et elle doit faire l'objet d'un « continuum » depuis le projet de recherche vers le lit du patient. Cette évolution de maturité doit permettre d'assurer que les innovations médicales répondent aux exigences réglementaires tout comme aux attentes des patients et des professionnels de la santé en termes de performance et de sécurité. Ces différents aspects, apportés et mis en évidence par les contributions que nous venons de décrire, influent directement sur l'évolution de la technologie elle-même et ont un fort impact sur les aspects techniques.

Nous voyons, dans la description que nous venons d'en faire et en appui avec les travaux de YOCK et al. (2015), de BLEVINS et al. (2018) et de MOREAU-GAUDRY et al. (2010) qu'une continuité du processus d'innovation est nécessaire. Elle va de l'invention à l'innovation et est ponctuée par les activités des projets de recherches, des essais cliniques et de l'étape de prétransfert, suivie des autres phases jusqu'au produit marqué.

Il est donc nécessaire de considérer comment les activités techniques, dans leur déroulement à travers les phases que nous avons vues, peuvent progresser dans leur maturité vers les étapes qui petit à petit intègre les notions de réglementation et de risque. Nous allons maintenant, explorer plus en détail la maturité technologique dans ce contexte complexe des SdMT.

2.2 Échelle de maturité technologique

Comme nous l'avons vu, une manière classique de considérer la maturité technologique est de l'évaluer selon l'échelle des niveaux de maturité technologique, aussi connue sous le nom d'échelle TRL (MANKINS 1995). La première version produite par la National Aeronautics and Space Administration (NASA) a subi diverses mutations et influences pour devenir un outil d'évaluation utilisé, par exemple, dans le cadre de la politique européenne d'innovation (HÉDER 2017).

Malheureusement, les échelles de maturité peuvent être sujettes à interprétation et les termes utilisés n'ont pas de définition unique. Par exemple, le terme prototype peut être l'équivalent d'un "démonstrateur" ou désigner un modèle non fonctionnel ou une maquette y compris au sein d'un même domaine. HÉDER (2017) montre qu'il existe un risque réel de confusion et d'abus dans l'utilisation d'échelles sur un plan général et qu'il est nécessaire de disposer de guides spécifiques aux disciplines et d'échelles plus adaptées.

Certaines initiatives ont été introduites pour apporter plus de précision selon un domaine scientifique spécifique, comme cela a été fait en chimie (BUCHNER et al. 2019). Selon l'activité, d'autres échelles ont été définies. Par exemple, l'échelle IDEAL-D (SEDRAKYAN et al. 2016) est une échelle adaptée aux dispositifs médicaux en environnement clinique. Elle est dérivée des recommandations IDEAL (McCULLOCH et al. 2009), qui sont spécifiques au processus de développement de l'innovation pour la thérapie interventionnelle dans un cadre clinique. Elle défi-

nit cinq étapes différentes (Idea, Development, Exploration, Evaluation, Long-term Study) qui marquent ce processus.

Il existe également l'échelle HIR (CIMIT 2018) (HealthTech Innovation Readiness), qui décrit le processus d'innovation technologique médicale en se basant sur quatre repères liés aux risques : 1) le risque clinique, 2) le risque économique et marché, 3) le risque réglementaire, 4) le risque technique.

Ces échelles, bien qu'elles aient le mérite de définir différents états de maturité, sont soit trop spécifiques et ne se rapportent pas directement aux logiciels, soit trop générales et ne fournissent pas de méthode quantitative spécifique pour évaluer le niveau de maturité d'un projet logiciel dans le domaine médical.

Nous allons aborder, dans la suite de cette section, le contexte normatif qui s'impose dans les activités d'innovation.

2.3 La qualité du point de vue des normes internationales

La phase industrielle est régie par des textes réglementaires basés sur des normes telles qu'ISO/IEC 62304, ISO/IEC 62366, ISO/IEC 82304, ISO 14971, ISO 13485 et bien d'autres. Toutes ces normes, et plus particulièrement les normes IEC, contiennent à la fois les spécifications techniques, mais également les activités qu'elles réglementent au travers des descriptions des exigences et des processus liés au contrôle de la qualité et de la sécurité.

La norme ISO/IEC 62304 est la norme principale en ce qui concerne les logiciels de dispositifs médicaux et, à ce titre, elle sera la seule norme étudiée dans le présent document. Cette norme contient les aspects techniques et les activités de l'ingénierie logicielle ainsi que les exigences et les processus liés au contrôle de la qualité et de la sécurité de tous les logiciels de dispositifs médicaux. Elle sera détaillée dans le chapitre 4 à la section 2.1 (voir page 69).

La norme ISO/IEC 62304 fournit une liste d'exigences réglementaires qui doivent être satisfaites pour commercialiser un SdMD conformément à la plupart des réglementations nationales, Européenne ou internationales (par exemple, l'approbation de la FDA pour les États-Unis d'Amérique et le marquage CE pour l'Union européenne). Toutes ces exigences indiquent ce qu'il faut vérifier dans le processus de conception et de maintenance et énumèrent les contrôles de qualité attendus.

En outre, la norme ISO/IEC 62304 définit trois classes de sécurité logicielle : A, B et C, la classe C étant la plus stricte. Néanmoins, le système de classification de la sécurité n'est pas pris en compte dans cette étude. En effet, la notion de risque est principalement régie par la norme ISO 14971 (ISO/IEC 2007). De plus, l'évaluation de la classe de sécurité dépend de l'usage médical prévu (REGULATION 2022), qui peut ne pas être entièrement connu ou décrit au cours de la phase de recherche, comme nous l'avons mentionné précédemment (voir section 1.4 à la page 31).

Ce contexte normatif s'impose de manière incontournable pour les activités d'ingénierie. Depuis 2017, la réglementation a changé (REGULATION 2022) et ce contexte normatif a pris une place très importante et s'invite de plus en plus dans les activités de recherche.

La qualité imposée par les normes est différente de la qualité en génie logiciel et est difficilement applicable en phase recherche.

Nous allons voir quelques initiatives qui sont proposées pour mieux intégrer cette charge réglementaire qui s'impose aux activités de développement des SdMT.

2.4 Naviguer entre réglementation et innovation en ingénierie logicielle

Des initiatives sont apparues pour faciliter la prise en compte de la réglementation dans les développements. C'est le cas par exemple des travaux de Rust et al. (2016b) qui proposent aux PME et petites structures innovantes ayant à mettre en œuvre les processus sous-jacents à la norme ISO/IEC 62304, d'utiliser des artefacts « How-To » élaborés en utilisant des « patterns » de conception. Chaque étape de la feuille de route qu'ils proposent fait référence à un « pattern » de conception appropriée, qui contient les informations de base pour guider l'équipe de développement logiciel dans la mise en œuvre du processus. Cette assistance, ne s'applique pas au développement ou à la recherche sur la technologie elle-même et correspond surtout à la phase du développement qui commence dans le prolongement des recherches pour faire émerger les nouvelles technologies sur le marché.

D'autres travaux se sont concentrés sur la conformité des logiciels médicaux lors du démarrage du processus d'industrialisation (Rust et al. 2015; Kim et al. 2019). Cependant, une lacune subsiste concernant le rapport à l'évaluation de la conformité réglementaire des prototypes en phase de recherche. La réglementation stricte en phase de recherche peut entraver l'innovation, où la flexibilité est essentielle pour développer des démonstrateurs en phase avec tous les objectifs des projets de recherche.

Kim et al. (2019) proposent de palier l'absence de prise en compte de la norme ISO/IEC 62304 par l'adoption d'une démarche permettant d'intégrer les processus qualité dans les processus de développement. Cette approche repose sur une ontologie qu'ils ont définie à partir de plusieurs normes et se poursuivrait jusqu'à la phase finale d'élaboration du produit. Mais les processus n'existent pas forcément en laboratoire de recherche où les objectifs peuvent ne pas être uniquement liés au transfert. De plus, il est difficile d'intégrer des méthodes intrusives qui puissent être facilement utilisées par les chercheurs.

De même, l'application de techniques industrielles semble plutôt mal adaptée au contexte de la recherche. Il paraît évident que, durant cette étape de recherche académique, il serait très difficile d'appliquer des techniques de l'industrie comme le propose N. Varachiu (Varachiu 2020) pour la conception des prototypes en recherche appliquée.

Dans ses travaux N. Varachiu propose d'élaborer les prototypes de recherche selon les règles de la conception robuste mettant en œuvre des techniques complexes comme le plan d'expérience, par exemple. Il s'agit dans cette technique de conception de créer des systèmes qui sont constamment performants, et cela, malgré la présence de variations ou d'incertitudes lors de la fabrication, de l'utilisation, ou de l'environnement.

L'utilisation d'une telle méthode serait excellente à condition d'imposer des modifications importantes dans l'environnement de recherche et dans les méthodes de

travail des chercheurs. Par conséquent, la mise en place de la norme ou l'application de technique de conception robuste en phase de recherche entraînerait une charge supplémentaire non valorisable pendant la phase de recherche.

Par conséquent, il est nécessaire d'apporter une approche plus orientée vers les besoins actuels des activités de recherche. Les solutions proposées, pour être applicables, ne doivent pas contraindre les personnels de recherche en dehors de leur processus habituel. L'évolution de la technologie doit suivre un modèle de maturité qui permet d'atteindre des objectifs de transfert, mais également ne pas empêcher la réalisation d'autres objectifs typique de la recherche. Cela peut être la réutilisation pour de futures collaborations, la réorientation du projet pour répondre à de nouveaux besoins, cette liste n'est pas exhaustive. Il est donc nécessaire de s'éloigner de l'implémentation d'un nouveau processus en phase recherche.

Pour proposer une approche adaptée, il nous faut comprendre comment la maturité est représentée dans le cadre des technologies logicielles. En adoptant une approche de la maturité du point de vue des systèmes d'information dans un environnement particulier comme celui de la recherche, nous pouvons envisager des solutions pour évaluer la maturité des SdMT. Dans la section suivante, nous allons voir comment se définit la maturité dans les systèmes d'information, en général. Et, nous chercherons à appliquer cette définition en dehors d'une notion de processus fixe et contraignant.

2.5 Les modèles de maturité dans les systèmes d'information

Dans leurs travaux, LASRADO et al. (2015) étudient en détail ce qui caractérise un modèle de maturité dans le cadre de la recherche sur les Système d'Information (SI). Tout d'abord, les auteurs ont cherché à rassembler et à analyser les contributions existantes sur ce sujet pour identifier les tendances, les lacunes et les points forts de la recherche actuelle. Ils ont aussi examiné les problèmes méthodologiques et théoriques que les chercheurs rencontrent lorsqu'ils développent et utilisent des modèles de maturité dans les SI. À l'issue de leur analyse, ils ont proposé des recommandations pour construire des modèles de maturité de manière plus standardisée et efficace, en intégrant des vocabulaires et des pratiques uniformes.

En effet, LASRADO et al. (2015) ont cherché à déterminer, au travers d'une étude systématique, s'il existait une structure générique pour les modèles de maturité. Les résultats de leurs travaux montrent plusieurs aspects intéressants et utiles pour nos propres travaux. Dans un premier temps, il y a une tendance marquée dans la littérature pour le développement de nouveaux modèles de maturité axés sur des problèmes spécifiques à un domaine ou sur de nouvelles technologies d'entreprise. Ce point va dans le sens des remarques faites par HÉDER (2017) sur la nécessité de spécialiser les échelles de maturité.

Un autre point ressort de cette étude (LASRADO et al. 2015) qui est que les modèles de maturité sont vus comme des théories normatives, des guides de meilleures pratiques et des outils de « benchmarking ». Elles doivent permettre d'atteindre un objectif pour lequel le niveau de maturité le plus élevé est défini. Cela était à la base de l'argumentation des chercheurs en chimie pour l'établissement de leur propre échelle (BUCHNER et al. 2019) : proposer un outil permettant d'aboutir à un objectif de préparation spécifique pour intégrer un processus exigeant. Cela

donne une bonne indication des motivations qui mènent à établir ces modèles. Selon les auteurs, les modèles de maturité doivent être composés de cinq éléments principaux (figure 2.4), qui sont :

Les niveaux de maturité : ils représentent différents stades de développement qu'un processus peut atteindre. Par exemple, dans le modèle de maturité des capacités (Capability Maturity Model (CMM) en anglais), les niveaux incluent des stades comme initial, répétable, défini, géré et optimisé.

Les Dimensions : elles sont des catégories ou des domaines d'évaluation qui sont significativement importants pour la maturité. De plus, chaque dimension peut avoir plusieurs sous-catégories. Par exemple, dans le domaine des systèmes d'information, des dimensions pourraient inclure des aspects comme les processus, la technologie et les ressources humaines.

Les sous-catégories : elles sont des composantes plus spécifiques de chaque dimension qui permettent une évaluation détaillée. Par exemple, dans notre item précédent correspondant aux dimensions, dans la dimension "technologie", les sous-catégories pourraient inclure la sécurité, l'infrastructure et l'innovation technologique.

Le chemin vers la Maturité : il s'agit de la description de la progression à travers les niveaux de maturité, illustrant comment passer d'un niveau inférieur à un niveau supérieur. Dans notre cas, par exemple, nous pourrions décrire la transition de processus ad hoc à des processus standardisés et finalement à des processus optimisés.

Questions d'Évaluation : Ici, il s'agit des questions conçues pour aider à évaluer où se situe la technologie par rapport à chaque sous-catégorie de maturité. On peut citer, par exemple, des questions d'évaluation pour la sous-catégorie "sécurité" que nous avons indiquées dans l'item précédent, pourrait inclure : "L'organisation a-t-elle une politique de sécurité clairement définie et appliquée?"

Dans la figure 2.4 tirée de l'article de LASRADO et al. (2015), on retrouve les éléments précédents. Ils sont divisés en deux parties dont la première constitue la structure de conception des modèles de maturité. Et la deuxième montre comment la mesure s'applique, en partant des questions d'évaluation jusqu'à la mise en œuvre d'un score.

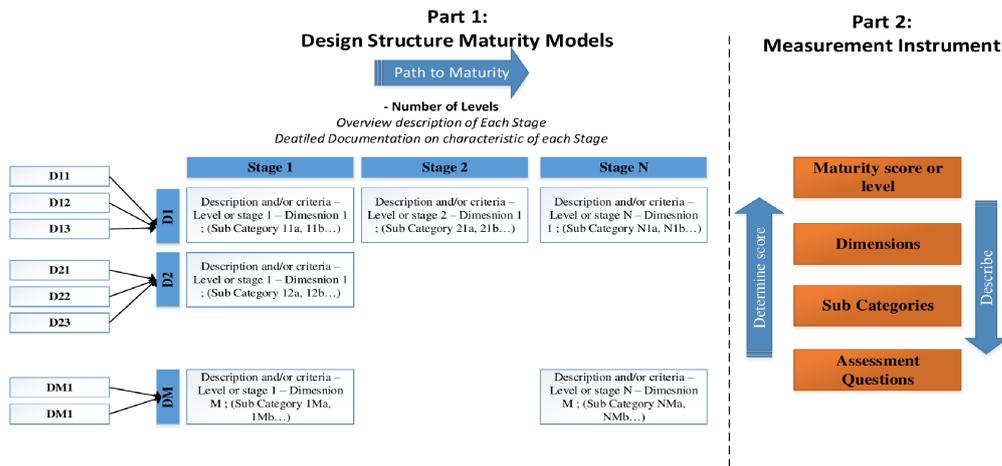


FIGURE 2.4 – Le modèle de maturité générique proposé par LASRADO et al. (2015).

Notre proposition s’appuie sur ces résultats. En effet, nous apportons une évaluation de la maturité dans le seul cadre des SdMT dans un domaine spécifique. Nous avons dans la section 2.2 défini à quoi correspond le plus haut niveau de maturité pour la SdMT. Nous avons ainsi, défini ce qui peut constituer l’objectif vers lequel une progression de maturité de la SdMT peut aller.

Toutefois, dans leur proposition, LASRADO et al. (2015) adopte, principalement, une vision orientée processus. Mais comme nous l’avons vu avec d’autres études (KIM et al. 2019; VARACHIU 2020) ce point de vue n’est pas approprié à la démarche de recherche en laboratoire qui demande une organisation plus flexible des activités.

Cependant, dans ses travaux, LASRADO (2018) montre clairement au moyen d’outils statistiques que les chemins vers la maturité sont multiples ce qui permet d’envisager de multiples adaptations au modèle générique proposé. En outre, cela contredit la linéarité communément adoptée dans les échelles de maturité comme seule approche et basée sur les processus uniquement.

En effet, traditionnellement, nous l’avons déjà évoqué pour les TRL, la linéarité représente la progression. Ce que LASRADO (2018) nous propose de généraliser est dans le même ordre d’idée que ce que BUCHNER et al. (2019) préconisaient. Il s’agit de considérer les niveaux un peu comme des conteneurs d’indicateurs. Une fois ces indicateurs satisfaits, il est alors possible de passer à un autre niveau sans obligations de linéarité. Les indicateurs peuvent même être considérés et travaillés en dehors d’une idée de progression.

3 CONCLUSION

Les sections précédentes ont exploré en détail notre approche pour évaluer la maturité des technologies médicales pilotées par logiciel. En définissant ces technologies et en proposant une méthode d’évaluation adaptée à la phase de recherche, nous avons mis en lumière l’importance de préparer ces technologies aux exigences des processus industriels et réglementaires. Les différents processus que nous avons décrits se rejoignent tous sur une trajectoire qui va de la recherche en

laboratoire vers les équipes R&D. Notre étude des étapes de cette évolution au travers des différents modèles nous a montré l'importance de la continuité entre la recherche fondamentale et l'application clinique. En intégrant ces éléments, nous pouvons à établir un modèle de maturité non linéaire qui peut potentiellement s'intégrer dans l'environnement des chercheurs tout en facilitant le passage des technologies vers des environnements réglementés et industriels.

En effet, cette possibilité d'établir un modèle de maturité non linéaire est un atout pour estimer la progression d'une technologie, de manière décorrélée à tout processus, en phase de recherche comme nous souhaitons le faire. Le modèle de maturité doit, cependant, s'appuyer sur une base solide et théorique comme l'indiquent LASRADO et al. (2015). C'est ce que nous faisons en reprenant les indications que les auteurs fournissent. Il doit donc être constitué d'un certain nombre d'éléments parmi ceux qui sont présentés plus haut. LASRADO (2018) montre que tous les éléments ne doivent pas être forcément présents pour représenter la progression de maturité.

Ainsi, nous avons considéré l'évolution de la **SdMT** au regard de son assimilation potentielle dans un processus sous contrainte réglementaire. Dans les éléments qu'il nous semble important de considérer, nous retenons « les dimensions ». Elles seront représentées par l'état des **SdMT** au fur et à mesure de l'avancement tout le long du chemin. Chaque dimension sera un sous-ensemble des éléments logiciels représentant le mieux le niveau correspondant. Ces éléments logiciels sont les artefacts de génie logiciel qui constituent la **SdMT**. Ils vont permettre, comme nous l'avons évoqué plus haut de mesurer la maturité. Ces artefacts de génie logiciel vont donc servir pour la méthode d'évaluation tout au long du chemin de maturité.

Ce chemin peut être représenté par trois niveaux 1) **Développement initial** est le niveau associé au projet de recherche lui-même, 2) **Préparation aux essais cliniques** correspond au premier contact de la technologie avec l'environnement réglementaire au moment des essais cliniques, 3) **Prétransfert** correspond au moment où la maturité est la plus haute du point de vue de la phase de recherche.

Ce qui sera alors évalué à chacun de ces niveaux sera la **SdMT** elle-même. Nous avons vu plus haut (voir page 40) que cette évaluation sera réalisée sur les artefacts de génie logiciel à tout moment pendant la phase de recherche. Nous allons dans la suite de cette section définir avec précision ce que nous entendons par « artefact de génie logiciel ».

3

ARTEFACTS DE GÉNIE LOGICIEL ET MODÈLE DE MATURITÉ POUR LES SdMT

1	Rôle et évolution des artefacts de génie logiciel dans les SdMT	47
2	Notre proposition : quantification de la maturité des SEA dans les SdMT	60
3	Conclusion	62

Un bon développement logiciel implique la création d'un ensemble d'artefacts qui soient aussi compréhensibles et maintenables que possible.

Steve McConnell

Les artefacts de génie logiciel (ou Software Engineering Artifact (**SEA**) en anglais) sont les briques fondamentales de la construction logicielle, par conséquent, ils sont un élément indispensable au développement des **SdMT**. Ils englobent une variété d'éléments de nature très différente et d'autres formes de données qui ont un rôle important pour la conception, le développement et la maintenance de systèmes logiciels médicaux.

En structurant de manière efficace ces artefacts, les développeurs peuvent améliorer la fonctionnalité des technologies médicales et leur conformité avec les réglementations existantes dans le domaine médical. Ce chapitre explore comment les artefacts façonnent les **SdMT** et jouent un rôle structurant pour parvenir à relever le défi du transfert de technologie avec plus de maîtrise.

Tout au long des différentes sections, nous verrons comment notre raisonnement se construit sur cette notion pour arriver à notre proposition.

1 RÔLE ET ÉVOLUTION DES ARTEFACTS DE GÉNIE LOGICIEL DANS LES SdMT

Cette section examine le rôle essentiel que jouent les artefacts de génie logiciel (**SEA**) dans le développement des **SdMT** et leur évolution au fil du temps. Les **SEA** ont constamment évolué et peuvent maintenant répondre aux besoins croissants de la complexité logicielle et de la rigueur réglementaire dans les **SdMT**.

1.1 Genèse et définition des artefacts en génie logiciel

Le concept d'artefact en génie logiciel (*SEA* en anglais) est apparu dans les années 1970. Dans son article ROYCE, WINSTON (1970), introduit ce qui sera plus tard appelé le modèle en cascade (waterfall model) pour le développement logiciel. Bien qu'il ne mentionne pas explicitement le terme "artefact", il parle de la production de « documents » et « d'éléments de sortie » à chaque phase du processus de développement de logiciels. C'est ce que montre la figure 3.1 tirée de son article.

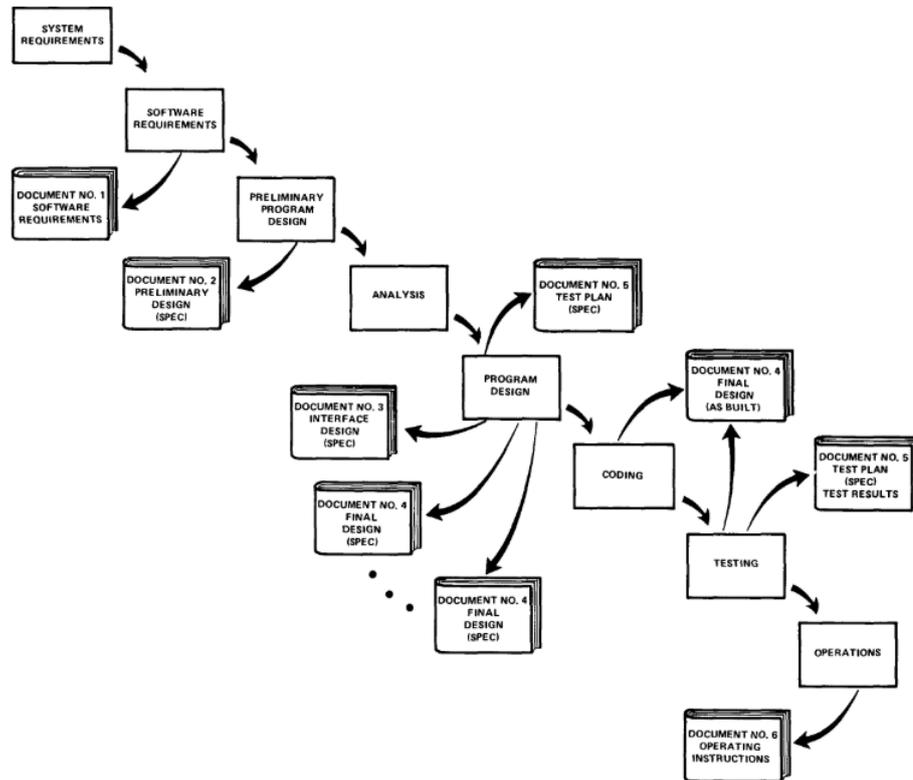


FIGURE 3.1 – Dans son modèle, ROYCE, WINSTON (1970) souligne l'importance des éléments produits en sortie des processus.

Ces documents et éléments de sortie, que l'on pourrait aujourd'hui qualifier de *SEA* dans le langage moderne du génie logiciel, sont utilisés tout au long du cycle de développement logiciel. Ils facilitent la conceptualisation, la création et la maintenance des systèmes logiciels. Ces *SEA* peuvent prendre diverses formes, telles que des documents, des modèles, des codes sources et d'autres types d'informations.

Nous verrons dans la suite de cette section comment le concept de *SEA* a évolué jusqu'à aujourd'hui. Ils jouent de multiples rôles dans et autour des activités de développement. Ils constituent, ainsi, un aspect essentiel à considérer dans le domaine du génie logiciel. Mais comme nous le verrons, la terminologie utilisée dans les *SEA* reste un défi.

1.2 Amélioration de l'intégration des SdMT par les ontologies

Les ontologies fournissent un cadre structuré pour l'unification des terminologies. Elles permettent la clarification des concepts complexes. De plus, elles peuvent se montrer particulièrement utiles dans le contexte des [SdMT](#). Elles peuvent répondre à la nécessité de précision et de cohérence des termes nécessaires pour la conformité réglementaire. Mais elles peuvent aussi apporter des contraintes fixes pour un cadre que l'on voudrait plus flexible. C'est ce que nous allons aborder maintenant.

PROBLÉMATIQUE DE LA TERMINOLOGIE ET SOLUTION PAR ONTOLOGIES

Des efforts ont été déployés pour améliorer l'utilisation des termes dans le milieu professionnel du développement logiciel et mieux caractériser les [SEA](#). À cet égard, il est pertinent de mentionner les travaux de HENDERSON-SELLERS et al. (2013). Ils ont constaté que l'utilisation disparate et souvent contradictoire des termes à travers différentes normes internationales en génie logiciel est monnaie courante. En particulier, ils se sont concentrés sur le problème des "silos" dans le développement des normes par le Sous-comité 7 (Sub-Committee (SC)7) de l'ISO. Cela a pour conséquence que différentes normes utilisent une terminologie similaire, mais avec des définitions variées et des interprétations qui ne sont pas toujours compatibles entre elles. C'est ce que révèle la liste des termes que l'on peut trouver dans le lexique de génie logiciel ([SEVOCAB](#)) qui, pour un même élément, propose souvent plusieurs définitions.

Pour remédier à cela, HENDERSON-SELLERS et al. (2013) proposent l'adoption d'une ontologie des éléments définitionnels (Definitional Elements Ontology (DEO) en anglais) et des ontologies définitionnelles configurées (Configured Definitional Ontology (CDO) en anglais). L'objectif est de créer un cadre unifié pour la standardisation des termes et des définitions à travers les différents standards et pratiques en génie logiciel. Cette approche peut également servir à rationaliser et à préciser la définition des [SEA](#).

Malheureusement, ce travail à notre connaissance n'a jamais abouti. Il n'existe pas à ce jour de systèmes utilisables et qui centralisent les termes de génie logiciel comme le fait [SEVOCAB](#) et qui appliquent les préconisations de HENDERSON-SELLERS et al. (2013).

Pour nos travaux, nous utilisons [SEVOCAB](#) comme une référence des termes de génie logiciel, il sera décrit plus en détail au chapitre 4 à la section 2.2 (voir page 71).

UTILISATION CONCRÈTE DES ONTOLOGIES DANS LE DÉVELOPPEMENT DES SDMT

Les ontologies peuvent aussi jouer un rôle structurant des [SEA](#) pour assurer la cohérence, la gestion des connaissances et la conformité réglementaire tout au long du développement logiciel. En adoptant les ontologies, les acteurs disposent d'une structure rigoureuse qui leur permet de naviguer efficacement à travers les complexités des normes internationales et des réglementations médicales. Dans son approche, RUIZ et al. (2006) montre que les ontologies peuvent être utilisées pour structurer et clarifier les processus de développement logiciel. Elles offrent une représentation explicite et structurée des connaissances du domaine. Elles servent

aussi de pont entre les différentes terminologies et définitions utilisées à travers les normes. Cela facilite l'interopérabilité et la réutilisation des artefacts logiciels au cours des développements lors des phases de développements industriels. Cette structuration contribue à une meilleure communication entre les parties prenantes et améliore les processus de vérification et de validation.

Plus précisément, SANTOS et al. (2012) applique ce concept d'ontologie spécifiquement à l'industrie du dispositif médical. Selon SANTOS et al. (2012), les ontologies permettent également de gérer les informations complexes et interdépendantes caractéristiques des dispositifs médicaux. Là encore, les ontologies peuvent faciliter la traçabilité et la mise en conformité des dispositifs avec les exigences réglementaires en codifiant les relations entre les différentes exigences et les aspects du dispositif. Dans leurs travaux, SANTOS et al. (2012) prennent les ontologies comme base pour définir des processus génériques à suivre à chaque étape de l'évolution d'un dispositif médical, depuis l'idée jusqu'au marché. Ici, les SEA jouent avec les concepts qu'ils représentent un rôle très important.

D. KIM et al. (2019) renforcent encore cette idée en utilisant la notion de SEA conjointement avec celle d'ontologie. Leurs travaux illustrent comment les ontologies peuvent être employées pour intégrer des normes variées dans un cadre unifié. En rationalisant les associations entre les normes, ils réduisent la redondance et optimisent ainsi les efforts de conformité en faisant ressortir les SEA. Ce qui est particulièrement pertinent lors de la gestion de la documentation et des exigences réglementaires dans le développement des SdMT, car les développeurs sont habitués à manipuler ces SEA. Pour D. KIM et al. (2019) il s'agit de faciliter la transition du point de vue des développeurs d'une technologie qui doit se conformer aux exigences de la norme ISO/IEC 62304 et des normes qui lui sont attachées. Les résultats des travaux de D. KIM et al. (2019) s'appliquent dans un contexte industriel, au sein d'équipes qui doivent travailler pour l'élaboration d'un SdMD comme futur produit respectant la réglementation.

D'autres travaux peuvent être cités pour montrer que les ontologies sont utilisées pour leur capacité à structurer le logiciel, les activités pour le produire et les SEA qui le constituent en fonctionnant comme des passerelles entre les domaines. Par exemple, HAGEDORN et al. (2015) mettent en évidence l'utilisation des ontologies pour le raisonnement interdomaines. Cette approche permet aux développeurs de dispositifs médicaux de découvrir des solutions innovantes par la réutilisation des concepts et des fonctionnalités à travers différents domaines. Pour y parvenir, les auteurs utilisent les relations entre les entités décrites dans les ontologies, que l'on peut considérer, ici, comme des artefacts. L'application de leur méthode a permis de créer un framework (Concept Ideation Framework for Medical Device Design (CIFMeDD)) qui intègre dans une ontologie résultante tous les artefacts des différents domaines et pas seulement les SEA, pour aider à la conception. La figure 3.2 montre comment l'utilisation du framework CIFMeDD a permis d'utiliser les connaissances techniques et médicales en une seule représentation dans le cas d'une procédure simplifiée de liposuction.

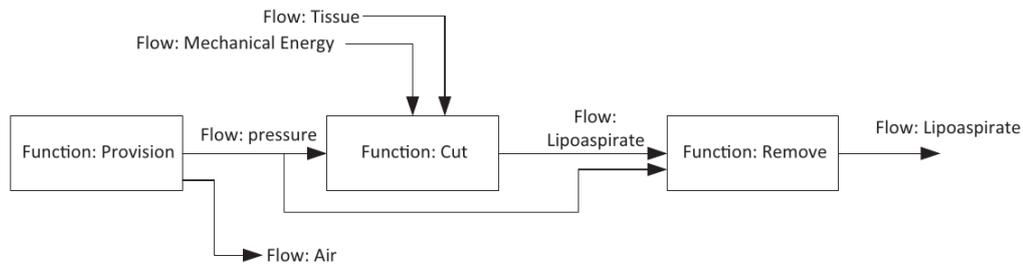


FIGURE 3.2 – Dans HAGEDORN et al. (2015), chaque élément utilisé est un artefact physique ou conceptuel qui a servi à construire l’ontologie pour fournir des aides à la conception.

Cette approche est intéressante pour l’innovation dans les [SdMT](#), où les technologies émergentes peuvent bénéficier de perspectives interdisciplinaires. Elle mixe les différents sous-produits obtenus indépendamment de leur nature, pour fournir des orientations. Là encore, il s’agit d’innovation dans le cadre des premiers pas industriels vers un produit médical innovant en accord avec les besoins du marché.

UTILISATION DES ONTOLOGIES DANS LA STRUCTURATION DES SEA

Ces perspectives montrent clairement le potentiel des ontologies comme outils dynamiques et intégratifs dans les [SdMT](#) par rapport à leurs rôles structurant et multidisciplinaire. Elles offrent une méthode systématique pour aborder les défis du développement logiciel dans un environnement aussi réglementé et complexe que celui des technologies médicales. Toutefois, dans le cadre de la recherche, il est difficile de transposer ces techniques aux projets.

En effet, l’utilisation d’ontologie sous-entend qu’une base d’exigences standardisées soit disponible pour être utilisée pour leur construction. Or, il est rare qu’un tel contexte existe au sein des équipes, même dans le cadre de la recherche translationnelle. Comme nous l’avons vu au début de ce chapitre (voir section 2.5 page 42), la flexibilité dans ce contexte est primordiale pour faire émerger les [SdMT](#). Bien que les ontologies puissent garder un intérêt évident et ne soient pas fondamentalement écartées dans le cadre de notre démarche, il est cependant nécessaire d’adopter une autre base pour structurer le système des [SdMT](#). L’étude de la littérature nous a permis de nous orienter vers un environnement qui permet de définir le logiciel qui constitue les [SdMT](#) sans qu’il soit contraint par un processus. Cette définition repose sur l’approche orientée artefacts. C’est ce que nous présentons dans la suite de cette section. Nous ne considérons que les [SEA](#), et cela, indépendamment d’autres types d’artefacts que l’on peut trouver dans le contexte médical. Mais tout d’abord, nous examinerons plus en détail ce que sont les artefacts.

1.3 Rôles et taxonomies des SEA dans les SdMT

En introduction de cette section, nous avons vu comme les processus menaient à la production de [SEA](#). De plus, les ontologies ont été vues comme des outils particulièrement efficaces en accord avec ces processus pour mieux expliquer les relations et les concepts attachés. Mais on peut considérer les artefacts pour ce qu’ils sont indépendamment des taxonomies et des processus. En fait, on peut même dire que les [SEA](#) définissent à eux seuls le logiciel.

PFEIFFER (2020), dans son étude, a approfondi la notion de « logiciel ». Pour cela, il a examiné 23715 dépôts sur GitHub qui ont été choisis de manière systématique dans 25 langages de programmation. Il a montré que les logiciels comprennent divers types de SEA, y compris du code, des données, et de la documentation. Les fichiers de configuration, les images et les bases de données sont considérés comme des SEA de données, par exemple. La documentation, elle-même, peut inclure, entre autres, des manuels d'utilisation, des licences. Pour réaliser ces travaux, il s'est appuyé sur la définition d'un SEA proposée par l'ISO/IEC 19506 :2012, qui indique qu'un « artefact est un document tangible lisible par machine, créé pendant le développement logiciel. Les exemples sont les documents de spécification des exigences, les documents de conception, le code source et les exécutables ».

En se basant sur ce que sont les SEA qui constituent les logiciels et ce qu'il a observé dans les dépôts étudiés, PFEIFFER (2020) a proposé une taxonomie des catégories de SEA. Pour y parvenir, il a mené une étude semi-automatisée de l'ensemble des dépôts, à l'aide de scripts Python. C'est en examinant le contenu et les extensions des SEA qu'il a déterminé leur type. En étudiant la distribution et la fréquence des différents types de SEA au sein des dépôts, il a mis en évidence leur rôle et leur importance relative dans la composition du logiciel. Cela a permis d'établir une catégorisation des SEA. Il obtient quatre catégories, elles-mêmes subdivisées en sous-catégories, et qui se répartissent de la manière suivante :

Code : il s'agit de code utilisés en programmation, incluant le code source écrit dans divers langages, des scripts interprétables, des fichiers binaires exécutable, ainsi que des scripts et configurations utilisés pour la compilation et la configuration des environnements d'exécution.

Données : il s'agit de fichiers. Ces fichiers comprennent des images, des vidéos, de la musique et des fichiers de configuration qui définissent le comportement des logiciels. On y trouve aussi des bases de données pour le stockage structuré de données, des fichiers de polices pour la typographie, ainsi que des archives qui regroupent plusieurs fichiers compressés. Le texte mentionne également des fichiers de marquage comme le XML ou HTML, des documents tels que les PDF ou Word, et des données d'application spécifiques utilisées dans des contextes propriétaires.

Documentation : ici on distingue deux catégories de documents textuels associés aux logiciels. D'une part, la "Prose", qui inclut des textes écrits tels que les fichiers README, les notes de version et les spécifications, et d'autre part, les documents "Juridiques", comme les licences et les accords de non-divulgation.

Autres : cette catégorie inclut les SEA qui ne sont pas classables de manière simple dans les catégories de code, de données ou de documentation

L'étude de PFEIFFER (2020), fait ressortir le **code**, les **données** et la **documentation** comme étant des composants essentiels du logiciel. Il indique également, combien les **autres** types de SEA peuvent être tout aussi essentiels pour assurer une gestion efficace, une bonne expérience utilisateur et la conformité légale. Il montre que le logiciel est un écosystème de SEA interdépendants, chacun contribuant à la stabilité, la performance et la sécurité du système dans son ensemble.

Dans ses conclusions, PFEIFFER (2020) revient sur le rôle des SEA dans les logiciels. Comme nous l'avons vu, il insiste sur le fait que les logiciels ne se limitent pas au code source et aux programmes. Chaque type de SEA joue un rôle important dans la fonctionnalité, l'utilisation et la maintenance du logiciel. Il souligne aussi l'omniprésence de la documentation dans les dépôts analysés, ce qui renforce l'importance de ce type de SEA.

Enfin, il indique que cette représentation permet une meilleure compréhension de ce qu'est un logiciel et par extension une technologie logicielle, et cela, sur plusieurs niveaux, tant structurel que fonctionnel. Au travers de la taxonomie qu'il propose, il aide à mieux comprendre et à structurer la complexité des composants logiciels. Mais plus encore que cela, il indique que les contrats et les normes industrielles devraient être révisés pour refléter la réalité des SEA. Cela plaide en faveur d'une approche plus orientée sur les SEA que sur les processus.

Cette dernière affirmation s'inscrit directement dans l'idée que considérer le logiciel du point de vue des SEA qui le constituent permet de le définir précisément. Par conséquent, les SEA apportent une définition plus fiable, d'un logiciel ou d'une technologie logicielle, pour servir de fondation à une vision commune pour les parties prenantes. Notre étude est en droite ligne avec cette considération. Cette décomposition en différentes catégories de SEA est centrale pour notre approche. Mais nous avons, en plus, besoin d'une définition qui permette d'observer l'évolution des SEA au cours de la construction d'une technologie. Dans la section suivante, nous considérons d'autres travaux qui permettent d'envisager une dimension dynamique aux SEA et de ce fait à la SdMT elle-même.

1.4 Dynamique et adaptabilité des SEA dans le développement des SdMT

Comme discuté dans les sections précédentes, l'utilisation des artefacts dans le développement logiciel peut prendre plusieurs formes. Les ontologies elles-mêmes peuvent représenter des structures fondamentales et complexes et aussi être des SEA (Ruiz et al. 2006). Dans le contexte des projets de recherche que nous souhaitons couvrir, les spécifications concernant la technologie peuvent évoluer progressivement avec l'avancement du projet. Cette évolution concerne des choix qui peuvent être différents selon les orientations qui s'avèreront pertinentes ou non pour le projet lui-même. Par exemple, il peut être préférable de changer d'approche pour résoudre un problème lié à l'application ou aux données, où il est possible que la technologie nécessite d'autres approfondissements pour aborder une question de recherche. On voit bien qu'il est difficile de fixer les spécifications liées à une technologie dans un contexte où les spécifications, elles-mêmes, peuvent faire l'objet d'investigation. Cela rend très difficile l'application des ontologies de manière prédictive pour les développements futurs.

En revanche, en envisageant les artefacts selon une classification systématique telle que celle proposée par PFEIFFER (2020), nous parvenons à obtenir une vision structurée de la technologie logicielle. Et cela sans qu'il soit nécessaire de disposer d'une définition préalable pour définir le logiciel à produire.

De plus, nous souhaitons pouvoir évaluer l'évolution de la structure logicielle d'une SdMT au cours du temps dans le contexte de la recherche et des projets de recherche qui s'étendent sur plusieurs années. C'est pourquoi il est important

de prendre en compte les dynamiques évolutives des **SEA**. Les contributions de MÉNDEZ FERNÁNDEZ et al. (2019) offrent les fondements pour cette adaptation.

Dans la suite de cette section, nous aborderons les travaux de MÉNDEZ FERNÁNDEZ et al. (2019) et les expliciterons. Ensuite, nous détaillerons comment ils ont servi de base pour fonder notre approche.

UNE DÉFINITION CLAIRE ET STANDARDISÉE DES SEA

Depuis la vision de ROYCE, WINSTON (1970) jusqu'à la définition de cadres sémantiques pour aider à la conception de **DM** innovants (HAGEDORN et al. 2015), les **SEA** sont surtout considérés comme des sous-produits des processus. Cette approche, orientée processus, présente l'inconvénient de créer un lien fort entre la définition des **SEA** et l'existence des processus. Dans ce cas, le rôle des **SEA** peut être négligé au bénéfice des procédures. Une autre approche, orientée-artefacts, replace les **SEA** au centre des développements logiciel. Cette approche, préconisée par MÉNDEZ FERNÁNDEZ et al. (2019), encourage une réflexion plus approfondie sur les produits de travail eux-mêmes, leur qualité et leur interopérabilité. Mais surtout, ce paradigme favorise une approche plus flexible et dynamique qui peut s'adapter aux changements de besoins et aux divers contextes de projet.

Dans leur publication (MÉNDEZ FERNÁNDEZ et al. 2019), les auteurs constatent que, malgré l'utilisation fréquente du terme **SEA** dans la communauté du développement logiciel, il n'existe pas de consensus clair sur sa signification précise. Il propose donc d'établir une définition claire et standardisée des **SEA**. L'objectif est d'améliorer la compréhension et les discussions autour de ces derniers dans le génie logiciel. Pour, *in fine*, obtenir une meilleure conception des processus de développement logiciel et des outils orientés **SEA**.

MÉNDEZ FERNÁNDEZ et al. (2019) proposent de définir un artefact en complétant la définition de base qui consiste à dire que les **SEA** sont le résultat d'un travail. Il la complète en indiquant que ces **SEA** peuvent se décomposer en trois niveaux de perception 1) le contenu sémantique 2) le contenu syntaxique 3) la représentation physique.

Le contenu sémantique représente la signification de l'artefact. Elle est interprétée dans le contexte de la connaissance individuelle de l'observateur ou de l'interpréteur de la machine. Les théories sémantiques et les expériences du monde réel sont les principaux éléments qui influencent ce contenu. Ainsi, le contenu peut être interprété explicitement dans un cadre théorique sémantique ou implicitement en fonction de la compréhension du monde réel de l'observateur.

Ce dernier point constitue le « pragmatisme ». Ce terme désigne le contexte sensible dans lequel le contenu est interprété, fortement influencé par les connaissances individuelles, les expériences, les capacités et les attentes de l'observateur. Le pragmatisme englobe la manière dont un observateur humain comprend et interprète les informations. Il le fait en fonction de sa propre perspective et de son expérience du monde réel, en l'absence d'une théorie sémantique explicite dans laquelle le contenu peut être interprété.

La structure syntaxique décrit la manière dont le contenu des **SEA** est exprimé syntaxiquement. Elle peut comprendre une composition de plusieurs sous-artefacts montrant ainsi qu'un **SEA** peut être construit par plusieurs autres **SEA**. Des grammaires ou des métamodèles, décrivant comment les informations sont organisées

et reliées entre elles sont des éléments constitutifs de cette structure. La représentation physique est la forme tangible ou numérique du SEA à laquelle la définition initiale faisait référence. Par exemple, un document imprimé, un fichier numérique ou une entrée dans une base de données. La figure 3.3, tirée de l'article de MÉNDEZ FERNÁNDEZ et al. (2019), montre cette structuration des SEA en trois niveaux de perceptions.

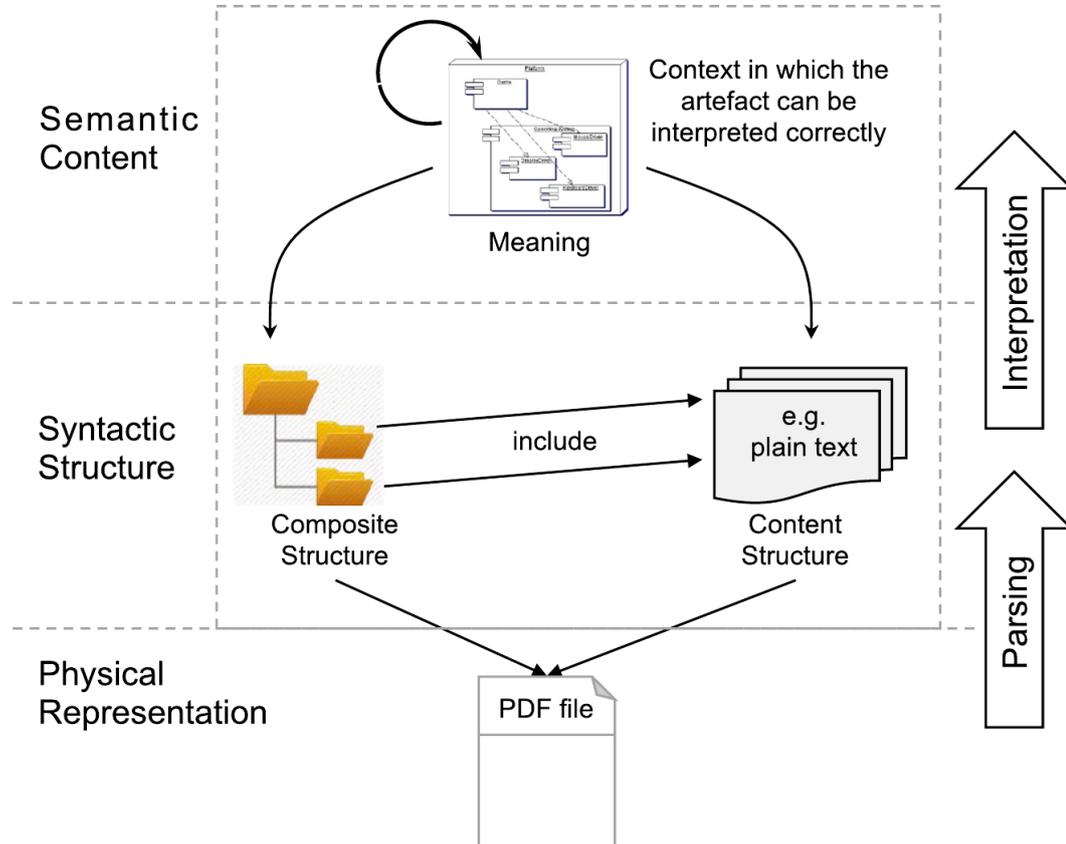


FIGURE 3.3 – Dans MÉNDEZ FERNÁNDEZ et al. (2019), les auteurs définissent un artefact comme un résultat de travail qui peut être décomposé en trois niveaux principaux de perception : le contenu sémantique, la structure syntaxique et la représentation physique.

En plus de ces trois niveaux de perception, MÉNDEZ FERNÁNDEZ et al. (2019) intègrent à leur définition deux étapes de traitement qui permettent d'interconnecter ces niveaux. Il s'agit de l'étape d'analyse (« Parsing ») qui permet l'identification d'une structure à partir de la représentation physique. Cette étape permet de décomposer la représentation physique en une structure organisée et compréhensible. Puis, vient l'étape d'interprétation qui permet l'extraction du contenu (signification) à partir de la structure. Cette étape implique la compréhension du sens et de l'intention derrière la structure syntaxique. Comme nous l'avons vu, elle est influencée, entre autres, par la connaissance de l'observateur. Il découle de cette décomposition en trois niveaux, une définition plus complète des SEA.

Ainsi, MÉNDEZ FERNÁNDEZ et al. (2019) donne la définition suivante :

Définition : Un artefact est un résultat de travail autonome, ayant un but spécifique dans un contexte, et constituant une représentation phy-

sique, une structure syntaxique et un contenu sémantique, formant trois niveaux de perception.

Cette définition est supportée par un métamodèle. La figure 3.4 est issue de la même publication et présente ce métamodèle.

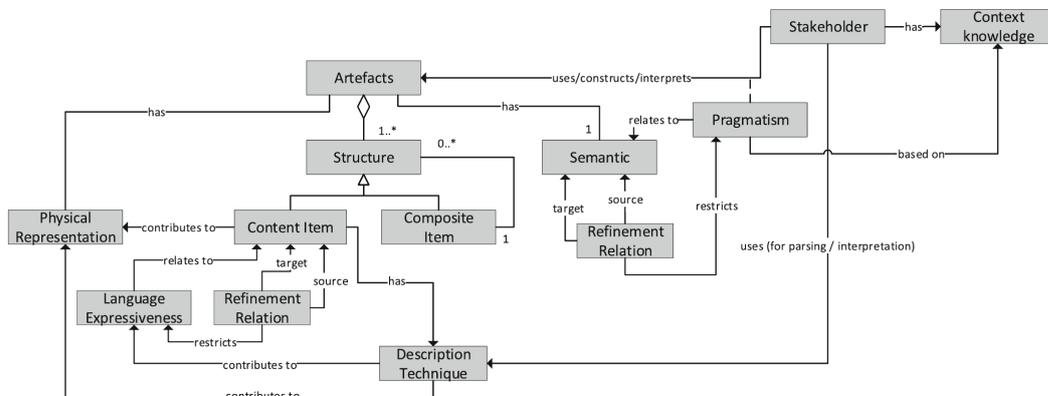


FIGURE 3.4 – MÉNDEZ FERNÁNDEZ et al. (2019) soutiennent leur définition d'un SEA par un métamodèle qui est représenté dans cette figure. On y retrouve notamment les éléments mentionnés et explicités dans ce document. Par exemple, la représentation physique, la structure sémantique ou encore le pragmatisme.

Ce métamodèle contribue à éclairer les relations entre les SEA, notamment par l'introduction de notions telles que le raffinement et l'équivalence. Il s'agit ici des éléments les plus complexes de cette vision orientée SEA que proposent MÉNDEZ FERNÁNDEZ et al. (2019), que nous allons développer ci-dessous.

CONCEPTS DE RAFFINEMENT ET ÉQUIVALENCE DES SEA

La notion d'équivalence entre SEA se réfère à la possibilité qu'ils soient considérés comme tels s'ils partagent la même intention sémantique. Ainsi, même s'ils diffèrent par leur structure syntaxique ou leur représentation physique, ils peuvent être considérés comme équivalents. Cette notion d'équivalence est graduelle. Elle va du contenu à la représentation physique. Cela implique que pour un niveau on retrouve cette notion d'équivalence avec les niveaux de perception supérieurs. Par exemple, cela signifie que pour que deux artefacts soient considérés comme équivalents, ils doivent au minimum partager une équivalence sémantique.

Pour illustrer cette notion d'équivalence, on peut prendre un exemple simple (voir figure 3.5). Par exemple, deux SEA qui peuvent être produits lors de la réalisation d'un SdMT de surveillance de la glycémie chez les personnes diabétiques. Chacun des SEA définit comment la SdMT doit réagir face à des mesures de glycémie anormales.

- Le SEA A est un document détaillé écrit dans un langage naturel. Il explique la logique derrière la réponse du logiciel à des lectures de glycémie élevées.
- Le SEA B est un diagramme de flux réalisé dans un outil de modélisation. Il illustre le processus que le logiciel doit suivre en cas de détection de niveaux de glycémie élevés.

Dans cet exemple, les deux SEA A et B partagent la même intention sémantique (« Intended meaning ») qui est d'assurer une réponse appropriée de la SdMT à des

mesures anormales de glycémie. Cela constitue un premier niveau d'équivalence que l'on peut constater. Mais le modèle de MÉNDEZ FERNÁNDEZ et al. (2019) va plus loin en permettant de vérifier si cette équivalence est complète. L'équivalence complète est définie sur une dimension horizontale des niveaux perception.

Ainsi, en poursuivant avec notre exemple sur la mesure de la glycémie, on peut vérifier si l'équivalence entre les deux SEA est complète en comparant les niveaux de chacun. C'est bien le cas, au niveau de la structure syntaxique où chaque SEA décrit un processus de réponse à une condition médicale, malgré des structures syntaxiques très différentes. Au niveau des représentations physiques, on observe également une équivalence, car malgré la différence qui existe entre les deux représentations physiques, elles sont analysables de la même manière.

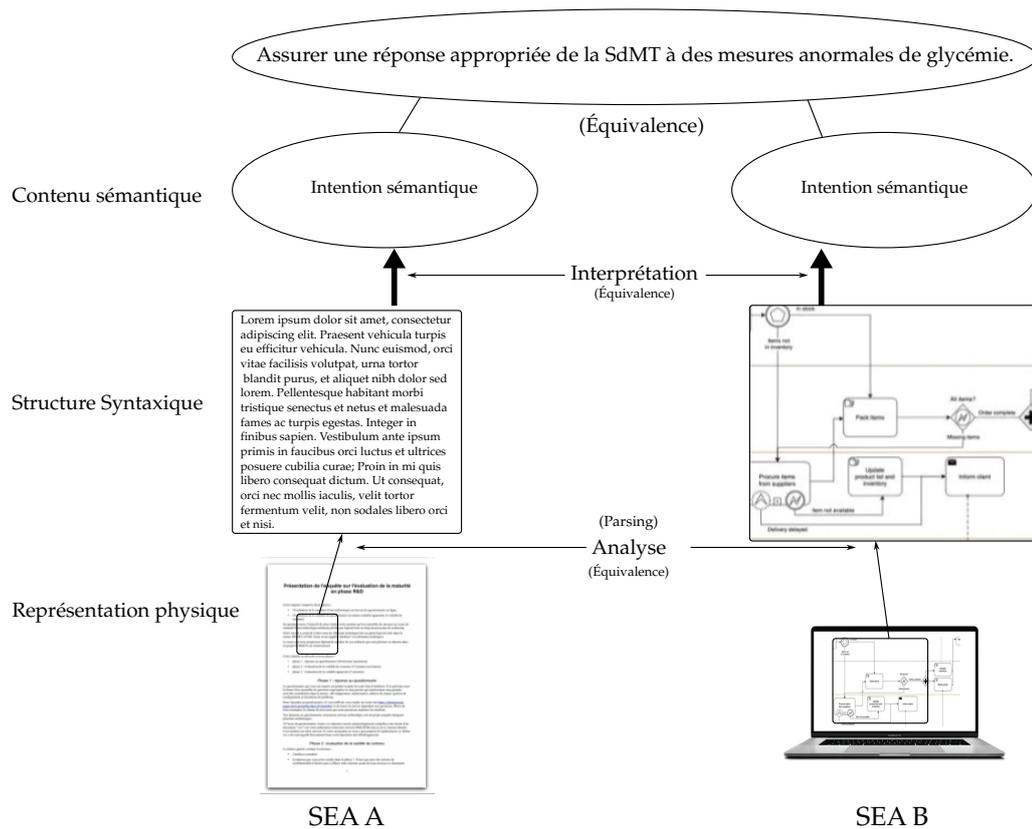


FIGURE 3.5 – La structure en trois niveaux de perception d'un SEA et les relations d'équivalence.

Dans notre exemple, le SEA A, qui est un document écrit, est la représentation physique d'un document textuel qui détaille comment le logiciel doit réagir aux mesures élevées de glycémie. Ce document peut inclure des descriptions verbales de l'algorithme, des protocoles d'intervention et des directives opérationnelles. D'un autre côté, le SEA B, qui est un diagramme de Flux, visualise le flux des opérations que la SdMT doit exécuter en réponse à ces mêmes mesures de glycémie. Ce diagramme inclut des nœuds et des connexions représentant les décisions et actions à prendre. Pour qu'il y ait équivalence sémantique, les deux SEA doivent être capables de guider un acteur, qu'il soit un clinicien, un ingénieur logiciel ou un régulateur, vers la même compréhension de la réponse que le logiciel doit avoir face

à des niveaux élevés de glycémie. Cela implique une compréhension unifiée de la part de tous les acteurs.

Cette notion d'équivalence est importante, en particulier dans les applications médicales où la précision et la clarté doivent, entre autres, permettre d'assurer la sécurité des patients et la conformité aux réglementations strictes. Avec cet exemple on comprend mieux l'importance des étapes entre les niveaux qui sont l'analyse et l'interprétation. Mais il illustre aussi qu'il est nécessaire de considérer le raffinement du SEA au sein d'une SdMT.

Comme MÉNDEZ FERNÁNDEZ et al. (2019) l'expliquent, le principe de raffinement est un processus par lequel un artefact est rendu plus précis et plus restreint dans son contexte sémantique. Pour réaliser ce raffinement, ils proposent d'ajouter des domaines supplémentaires dans le contexte sémantique ou de limiter l'expressivité du langage utilisé pour décrire le SEA. Cela permet de réduire le rôle qu'on les connaissances individuelles dans l'interprétation sémantique du SEA. Par exemple, en utilisant des modèles ou des langages formels cela ajoute des domaines techniques dans le contexte sémantique et le précise. Cela permet d'ajouter un ensemble de connaissance ou de règles et donc de limiter le pragmatisme. Comme nous l'avons vu plus haut, le pragmatisme se réfère ici à la pratique d'adapter la signification d'un SEA selon les intentions des parties prenantes et de la façon dont ces dernières utilisent et comprennent le SEA dans le monde réel.

Les travaux de MÉNDEZ FERNÁNDEZ et al. (2019) apportent une définition robuste de ce que représente un SEA.

Les SdMT sont un ensemble de SEA.

Notre raisonnement repose sur cette définition et nous allons, à présent, montrer comment ils peuvent servir de fondation à notre approche.

1.5 Évaluation de la maturité technologique des SEA dans les SdMT

Cette section illustre comment les SEA évoluent au cours de la transition vers l'environnement industriel et comment la définition selon MÉNDEZ FERNÁNDEZ et al. (2019) peut s'appliquer.

CAS D'USAGE : DE LA RECHERCHE À L'APPLICATION INDUSTRIELLE

Pour appliquer les principes de la définition des SEA par MÉNDEZ FERNÁNDEZ et al. (2019) à nos préoccupations, nous allons prendre un cas d'usage fictif, mais représentatif des activités que nous avons décrites plus haut dans ce chapitre dans le cadre du parcours d'une SdMT.

NAISSANCE D'UNE SDMT AU LABORATOIRE

Ainsi, considérons une équipe de recherche en laboratoire. Elle développe une SdMT. Durant cette phase, l'équipe se concentre sur l'innovation et la fonctionnalité et ne se préoccupe initialement pas des contraintes réglementaires ou des risques. Au cours de ses activités de développement, cette équipe est amenée à produire des SEA. Prenons deux de ses SEA et examinons-les au travers du prisme de la défini-

tion proposée par MÉNDEZ FERNÁNDEZ et al. (2019) ainsi que de leur évolution vers un potentiel transfert.

Considérons que l'équipe de laboratoire a produit au cours de ses recherches un prototype de **SdMT**. Il est souvent constitué par du code expérimental ou du pseudocode, par exemple, il constitue un **SEA** que nous nommerons A. Cette équipe produit également une documentation technique qui peut se présenter sous la forme de note de recherche, de description de résultats expérimentaux ou d'une première publication. Nous nommerons cette production le **SEA** « B ». Il peut s'agir d'une publication.

Dans ce cas, posons-nous la question de savoir ce qui constitue le contenu sémantique de ces deux **SEA**. Ici, les deux **SEA** décrivent ou implémentent la fonctionnalité de la technologie médicale. On peut observer des différences significatives dans les structures syntaxiques et les représentations physiques des deux **SEA**. Le code source est plus formel et structuré, alors que la documentation ou les notes sont potentiellement moins formelles ou plus narratives. Le code interprété par une machine n'a qu'une seule interprétation possible, contrairement au langage naturel qui peut comporter des imprécisions ou des omissions et dont le sens dépend de l'interprétation.

LA SDMT ÉVOLUE VERS L'INDUSTRIE

À présent, considérons une équipe de R&D industrielle dont l'objectif est d'intégrer la **SdMT** que nous venons de présenter. Cette équipe doit développer un produit logiciel conforme à la norme ISO/IEC 62304, un **MDS** par exemple. Dans ce cas, les membres développeront, entre autres, un **SEA** « C » représentant des spécifications détaillées conformes aux exigences réglementaires. Et, un **SEA** « D » représenté par le code source révisé et documenté selon les standards de l'industrie.

Dans le cas des **SEA** « C » et « D » le contenu sémantique est le même que pour les **SEA** « A » et « B ». En effet, ils ont tous pour objectif de décrire ou d'implémenter la même **SdMT** dans son objectif fonctionnel. Sur le plan de la structure syntaxique et de la représentation physique, les **SEA** industriels apparaissent plus structurés et répondants à des contraintes plus formelles pour répondre aux exigences.

On peut, à présent, évaluer l'équivalence entre d'un côté les **SEA** issus du laboratoire et les **SEA** issus de la démarche sous contrainte réglementaire. Sur le plan du contenu sémantique, nous observons une équivalence si les fonctionnalités fondamentales de la **SdMT** restent les mêmes. Quant au niveau des structures syntaxiques et représentations physiques, elles ne sont probablement pas équivalentes. Notamment, car les **SEA** industriels doivent être beaucoup plus structurés et détaillés pour satisfaire les normes réglementaires. Ainsi, les **SEA** de laboratoire ne peuvent pas être utilisés tels quels dans le contexte réglementé des équipes de R&D.

CONCLUSION DE L'EXEMPLE

Toutefois, même si, du fait des fortes différences entre les structures sémantiques et physiques des **SEA** de laboratoires et industriels, on ne peut pas avoir d'équivalence complète, on observe quand même une équivalence incomplète sur le plan du contenu sémantique. Pour rendre équivalents de manière complète les deux groupes de **SEA**, il est nécessaire d'intégrer le « raffinement ». C'est ce processus de raffinement qui permet l'adaptation nécessaire d'un contexte de laboratoire à

un contexte industriel, pour obtenir la conformité réglementaire. Dans ce cas, le processus de raffinement doit préserver l'intention sémantique originale et il doit permettre de reformuler les **SEA** pour répondre aux exigences de structure et de représentation de l'industrie.

Cet exemple montre que, bien que les **SEA** de départ et les **SEA** finaux puissent partager une intention sémantique similaire (ici la fonctionnalité du logiciel médical), ils ne peuvent pas, de prime abord, être considérés comme équivalents. Ils ne pourront l'être que si un travail considérable est fait pour aligner la structure syntaxique et la représentation physique avec les exigences du nouveau contexte de développement. Ce travail est incontournable, car il permet de garantir que le logiciel développé est non seulement fonctionnel, mais aussi sûr et conforme aux normes nécessaires pour une mise sur le marché. On retrouve ici le fondement des guides et autres règlements qui permettent de lancer les travaux et d'évaluer la maturité au démarrage du transfert des **SdMT** vers leur statut de **DM**.

Notre proposition repose sur ce principe d'évolution des **SEA** d'un contexte à un autre. C'est ce changement de contexte qui implique les plus grandes difficultés. En disposant d'un outil permettant de rapprocher les deux contextes sur le point commun des contenus sémantiques, nous abordons une possibilité de rendre plus compréhensibles et quantifiables les types de travaux à mener pour faciliter le transfert. Dans la sous-section suivante, nous montrons comment notre approche utilise la définition des **SEA** selon MÉNDEZ FERNÁNDEZ et al. (2019), pour apporter une solution à ces difficultés.

L'équivalence entre **SEA** permet de clarifier la transition vers le transfert.

2 NOTRE PROPOSITION : QUANTIFICATION DE LA MATURITÉ DES SEA DANS LES SDMT

Suite à cette étude bibliographique, nous considérons qu'une **SdMT** est constituée par un ensemble de **SEA** (voir figure 3.6). Une **SdMT** peut elle-même comporter des sous-unités pouvant elles-mêmes être considérées comme un ensemble de **SEA**. Notre vision se détache de la partie processus, car 1) comme nous l'avons vu, il n'existe pas de processus défini ou adapté dans la phase de recherche, 2) la transition vers des processus industriels semble difficile à obtenir en pratique, contrairement au raffinement des **SEA**.

Comme nous l'avons vu depuis le début de ce chapitre, la transition des **SdMT** du contexte de recherche en laboratoire à un environnement industriel réglementé représente un défi. Cela est principalement dû aux exigences réglementaires imposées par des normes telles que l'ISO/IEC 62304. L'hypothèse que nous formulons est que l'importance relative des **SEA** peut être quantifiée de manière objective en utilisant la fréquence de leur citation dans la norme ISO/IEC 62304 comme indicateur. En cela, nous suivons la méthodologie de PFEIFFER (2020) qui a utilisé la fréquence d'apparition de certains **SEA** dans les dépôts pour déterminer leur importance relative dans la constitution du logiciel. Cette quantification de la maturité

pourrait fournir un moyen systématique d'évaluer où en est la technologie au sein du laboratoire par rapport à l'objectif de maturité.

CLASSIFICATION DES SEA

Tout d'abord, nous proposons de classer les SEA en deux types en fonction de leur besoin de raffinement dès le projet de recherche et pour atteindre la conformité réglementaire (voir la figure 3.6) :

TYPE ① : Ce sont des SEA qui sont déjà dans un format formel ou très proche de la structure syntaxique finale. Les SEA de ce type ne nécessitent pas un raffinement important, car ils sont dans une structure syntaxique qui limite déjà le pragmatisme. C'est le cas par exemple du code source ou des révisions.

TYPE ② : Ce sont des SEA qui comportent des descriptions générales exprimées dans un langage non formel (langage naturel par exemple) et qui nécessiteront un processus de raffinement plus important. C'est le cas par exemple des notes de conception initiales, exigences prises en compte pour élaborer la technologie comme des publications ou d'autres indications en langage naturel.

Cette distinction permet également de rééquilibrer les SEA au cours de leur progression d'évolution, en identifiant une orientation des activités plutôt d'un côté (type ①) que de l'autre (type ②).

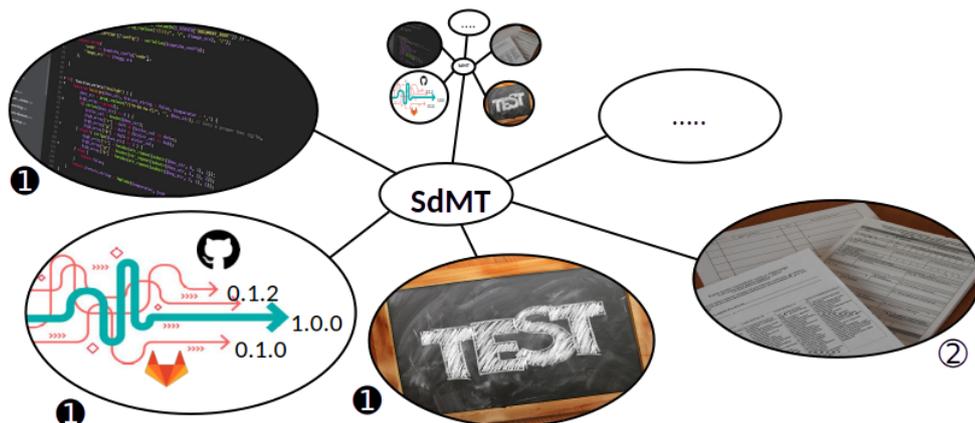


FIGURE 3.6 – Technologie médicale pilotée par logiciel (SdMT)

MÉTHODOLOGIE DE QUANTIFICATION

Pour pouvoir élaborer notre technique de quantification de la maturité technologique d'une SdMT, il faut nous assurer de suivre les étapes présentées ci-dessous.

1. Une étape sera nécessaire pour **extraire et valider** dans le domaine du génie logiciel tous les SEA présents dans la norme ISO/IEC 62304. Cette extraction doit être faite en notant la fréquence de chaque mention. Il sera nécessaire, après validation de l'appartenance au domaine du génie logiciel pour confirmer le terme extrait comme SEA, de procéder à son classement selon le type.
2. L'étape précédente constituera une base valide pour poursuivre avec une étape d'**attribution** à chaque SEA d'un poids relatif. Il sera basé sur sa fréquence de mention dans la norme, ce qui nous permettra d'établir son importance.

3. Ensuite, il sera nécessaire de **déterminer** une méthode de calcul basée sur le poids relatif cumulé des **SEA** pour obtenir l'indice de maturité global de la **SdMT**. Cet indice (ou score) de maturité devrait pouvoir être décliné par **SEA** et par type.

DÉVELOPPEMENT ET UTILISATION DU QUESTIONNAIRE

Enfin, pour être à même d'évaluer avec précision la présence des artefacts dans la **SdMT**, un questionnaire détaillé sera conçu. À l'aide des réponses obtenues au questionnaire, il sera possible d'établir un score de maturité par répondant pour la **SdMT** considérée, selon la méthode de calcul obtenue dans les étapes précédentes. Le score global de maturité technologique est calculé en additionnant les scores individuels des artefacts, fournissant une mesure objective du niveau de préparation réglementaire de la technologie qu'il sera possible de comparer pour chaque répondant.

3 CONCLUSION

À l'issue du présent chapitre, nous avons établi notre raisonnement. Nous considérons que déterminer le niveau de maturité d'une **SdMT** équivaut à déterminer sa **capacité à entrer dans le processus d'industrialisation ou à effectuer une évaluation clinique**. Les **SdMT** doivent répondre aux exigences réglementaires qui sont composées d'artefacts organisés, structurés et articulés par des processus qualité. Cependant, lors de la phase de recherche, les processus qualité n'existent généralement pas.

L'évaluation de la maturité proposée dans nos travaux considère la **SdMT** à n'importe quelle étape de la phase de recherche (voir Fig. 3.7). Cette **SdMT** est composée d'un ensemble d'éléments de génie logiciel que nous appelons des **SEA**. Le score que nous proposons, pour quantifier la maturité technologique, dépend uniquement de la présence des **SEA** dans **SdMT** considérée et qui sont directement cités dans la norme de qualité ISO/IEC 62304.

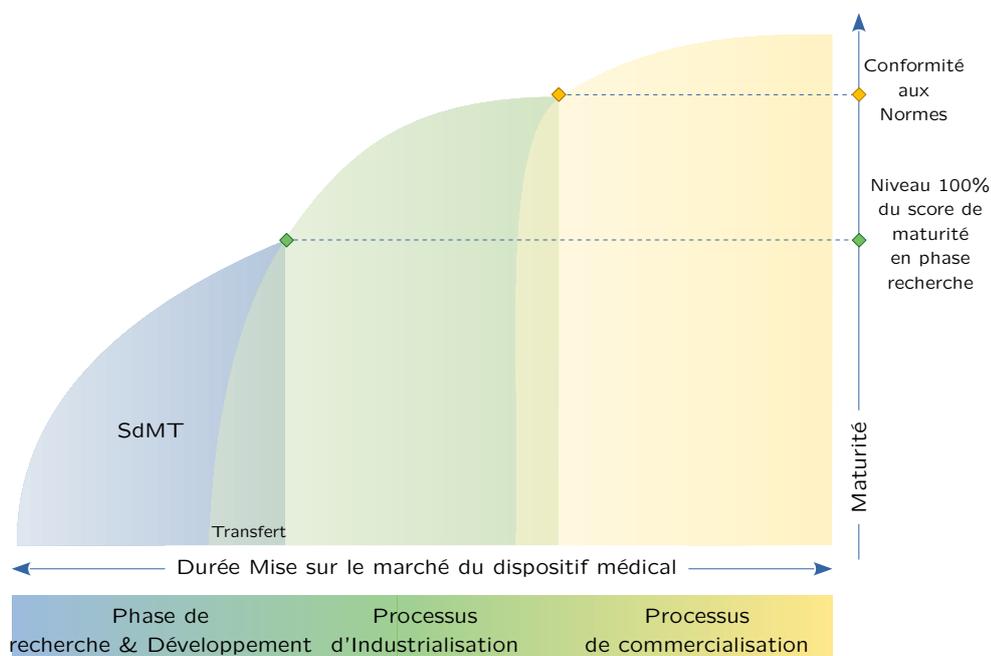


FIGURE 3.7 – La maturité d’un SdMT peut être évaluée à n’importe quel moment de la phase de recherche.

La figure 3.7 montre l’échelle présentant les phases que nous avons déjà vues à la figure 2.2. Elle apporte une dimension supplémentaire dans l’augmentation de la maturité des technologies médicales pilotées par logiciel dans un environnement global des SdMT.

En effet, durant la première phase, la maturité augmente jusqu’au niveau maximum atteignable par les activités de recherche. La fin de la phase de recherche et développement, comme elle est appelée ici, prend en compte les activités de recherche et de mise au point par les équipes de R&D. La zone de recouvrement désigné par le terme « transfert » correspond au moment où la technologie est la plus mature du point de vue de la phase de recherche.

Ensuite, la maturité continue d’augmenter dans les phases suivantes. Elle passera d’un niveau complet pour la phase de recherche à un niveau complet par rapport aux aspects réglementaires.

Ainsi, cette figure montre bien que la pleine maturité réglementaire est différente de la pleine maturité à l’issue de la phase de transfert. Elle montre, aussi, comme les deux niveaux se situent dans la continuité l’un de l’autre, pour aller vers la conformité réglementaire.

Pour pouvoir approfondir notre démonstration, il nous faut maintenant évoquer les étapes qui ont permis de construire le modèle d’artefacts sur lequel nous nous basons pour quantifier la maturité des SdMT. Toute la procédure pour aboutir à ce modèle de référence de maturité est décrite dans le chapitre suivant.

4

ÉVALUATION DE LA MATURITÉ DES TECHNOLOGIES MÉDICALES PILOTÉES PAR LOGICIEL

1	Postulats	65
2	Les outils normatifs pour construire une référence de maturité technologique	68
3	Méthodologie	73
4	Synthèse de la démarche	92
5	Conclusion	93

La créativité consiste à voir ce que tout le monde voit et à penser différemment.

Albert Einstein

1 POSTULATS

La première partie du travail de cette thèse a consisté en la construction d'une méthodologie d'évaluation de la maturité d'un point de vue quantitatif. Pour y parvenir, nous avons formulé des hypothèses, sur la base de nos observations et des éléments fondateurs apportés par la littérature telle que nous l'avons présenté dans le chapitre 3.

Nous avons observé que la démarche qualité est principalement entreprise lors de la phase d'industrialisation d'un dispositif médical (chapitre 2 section 2.3). De plus, dans le chapitre 3, le niveau de maturité d'une SdMT a été défini par rapport à sa capacité à entrer dans le processus d'industrialisation ou à être utilisée dans le cadre d'une évaluation clinique.

Bien qu'une SdMT encore en phase de recherche puisse ne pas inclure de processus ou d'activité de contrôle de la qualité, son développement a déjà produit des SEA (par exemple, le code source, l'exécutable, l'architecture logicielle, la liste des bogues, etc.). Indépendamment des langages, des plateformes et des cadres utilisés pour créer les SdMT, il va de soi que les activités d'ingénierie logicielle sont une partie intrinsèque de leur développement dès les premières étapes.

Par conséquent, certaines formes d'ingénierie logicielle ont nécessairement lieu pendant la phase de recherche, menant à la production de SEA. Certains de ces SEA liés à l'ingénierie logicielle font partie de ceux exigés par la norme ISO/IEC 62304 et constituent la base sur laquelle se construisent les processus sous-jacents dans la norme ISO 13485.

À partir de ces constatations, nous formulons cinq hypothèses.

Hypothèse 1 : les SEA sont le reflet des activités de développement

L'observation des SEA, tels que le code source, les exécutables, l'architecture logicielle, la liste des bogues, etc., au cours de la phase de recherche d'une SdMT suggère l'existence d'activités de développement associées. Cette hypothèse repose sur l'idée que la présence de tels SEA ne résulte pas simplement du hasard, mais plutôt des efforts délibérés pour les produire. Ces activités peuvent varier en matière d'organisation, de structure et de spécification. Il peut exister une diversité dans la manière dont les SEA sont obtenus. Certains peuvent provenir d'activités plus organisées et spécifiées, tandis que d'autres peuvent résulter de processus plus flexibles. L'importance de cette hypothèse réside dans le fait qu'il est nécessaire d'entreprendre des activités de développement pour créer une SdMT. Cette compréhension guide notre façon d'évaluer la maturité technologique. Elle transforme les SEA du génie logiciel en preuves tangibles et concrètes que nous pouvons utiliser dans une méthodologie d'évaluation quantitative.

Hypothèse 2 : il y a corrélation entre le nombre et la diversité des SEA et la maturité technologique

Nous postulons que le niveau de maturité d'une SdMT est lié au nombre de SEA de natures différentes, présents à un moment donné de son développement. Il s'agit de considérer le nombre de SEA produit à différents stades de la vie d'une SdMT, en envisageant une progression vers la maturité technologique. Cette progression est basée sur l'accumulation de SEA de natures différentes.

Ces principes établissent que la production et la diversité des SEA sont des indicateurs tangibles du niveau d'avancement de la SdMT. De plus, une augmentation du nombre de SEA reflète un engagement continu dans les activités de développement, signalant un stade de maturité technologique plus élevé. Cet engagement s'effectue vers un objectif qui peut être défini comme une référence de maturité.

Cependant, il est important de ne pas seulement considérer la quantité de SEA, mais aussi leur pertinence pour obtenir une évaluation complète de la maturité technologique. La diversité des types de SEA est plus importante que la simple quantité d'un même type. Par exemple, la richesse en fonctionnalités d'une SdMT ne la rend pas mature uniquement en raison de cette richesse ou du simple volume du code implémenté.

Hypothèse 3 : la présence de SEA peut être évaluée par des questions fermées

L'hypothèse 3 avance que la présence de SEA liés aux activités de génie logiciel dans le développement d'une SdMT peut être évaluée de manière simplifiée à l'aide de questions fermées dont la réponse est oui ou non.

Ces questions peuvent être conçues pour déterminer la présence ou l'absence d'éléments spécifiques de génie logiciel tels que le code source, la documentation, ou d'autres SEA pertinents.

Cette approche offre une méthode rapide et accessible pour évaluer la nature des activités de génie logiciel sans nécessiter une analyse approfondie.

Ces questions doivent être conçues de manière à capturer la diversité des **SEA**, garantissant ainsi une évaluation complète de la présence d'activités de génie logiciel.

Cette hypothèse suggère que l'utilisation de questions fermées peut fournir une première évaluation de la maturité technologique, en identifiant la base de **SEA** existants, et peut être intégrée dans une méthodologie d'évaluation quantitative de la maturité.

Hypothèse 4 : la norme ISO/IEC 62304 peut être utilisée comme référence de maturité

La norme ISO/IEC 62304 intègre des références aux **SEA** de l'ingénierie logicielle dans son contenu. Durant la phase de recherche (cf. chapitre 2, section 2.5 page 42), les processus de qualité liés à cette norme sont absents.

L'hypothèse 4 avance que la norme ISO/IEC 62304, en détaillant les **SEA** nécessaires pour assurer la conformité, peut être utilisée comme indicateur de maturité pour évaluer le développement d'une **SdMT**. En d'autres termes, la présence et la qualité des **SEA** spécifiés par la norme peuvent servir de critères pour évaluer la maturité technologique d'une **SdMT**.

On peut ainsi mesurer de quelle manière les **SEA** définis par la norme sont présents dans la phase de recherche. Dans l'ensemble des **SEA** produits lors des développements, un certain nombre étant produit pendant cette phase peut être comparé aux éléments logiciels équivalents de la norme.

Cette hypothèse rejoint l'hypothèse 2 quant à l'augmentation de la maturité en fonction du nombre de **SEA**, lesquels sont caractérisés par comparaison avec la norme.

Cette hypothèse suggère que la norme ISO/IEC 62304 peut servir de référence pertinente pour évaluer la maturité technologique en identifiant les **SEA** nécessaires à la conformité réglementaire.

Hypothèse 5 : La présence ou l'absence des SEA permet de définir la maturité d'une SdMT

Cette hypothèse suggère que la simple existence de **SEA**, tels que le code source, la documentation, ou d'autres éléments identifiés, peut fournir des informations sur la maturité technologique de la **SdMT**. En évaluant chaque **SEA** individuellement, indépendamment des autres, cette approche simplifiée offre une compréhension initiale de la présence des composants clés du génie logiciel par rapport à la capacité d'une **SdMT** à entrer dans un processus sous contrainte réglementaire.

Cette hypothèse met en avant la valeur intrinsèque des **SEA** en tant qu'éléments tangibles du développement. Nous proposons de considérer les **SEA** dans leur capacité à définir le logiciel de manière détachée du processus, comme expliqué au chapitre précédent.

En considérant cette hypothèse, l'évaluation de la maturité technologique peut se baser sur des critères spécifiques comme l'absence ou la présence offrant un choix binaire. Cela rejoint l'hypothèse 3, facilitant alors une approche quantitative et accessible.

Cette hypothèse sous-entend également que les [SEA](#) soient considérés de la même manière par tout évaluateur. Par conséquent, chaque [SEA](#) doit être clairement défini et compréhensible.

Dans la suite de ce chapitre, nous explorerons en détail les moyens et ressources sélectionnés pour étayer les hypothèses énoncées précédemment. Nous mettrons en lumière les choix que nous avons fait, derrière chaque source utilisée, en soulignant comment ces ressources contribuent à la robustesse de notre cadre d'évaluation. De plus, nous exposerons les méthodes spécifiques que nous avons développées à partir de ces ressources, décrivant de manière approfondie leur application dans l'établissement de notre proposition d'évaluation de la maturité technologique des [SdMT](#).

2 LES OUTILS NORMATIFS POUR CONSTRUIRE UNE RÉFÉRENCE DE MATURITÉ TECHNOLOGIQUE

Cette section présente les deux références utilisées pour générer et consolider la liste des [SEA](#).

Il s'agit de la norme internationale ISO/IEC¹ 62304 (section [2.1](#)), intitulée « Logiciels pour dispositifs médicaux - Processus du cycle de vie du logiciel » (« Medical device software – Software life cycle processes » en anglais), et le [SEVOCAB](#) (section [2.2](#)). [SEVOCAB](#) est étroitement lié à la norme ISO/IEC/IEEE 24765 :2017 qui poursuit le même but qui est de fournir des définitions standardisées des termes utilisés dans le génie des systèmes et du logiciel. Cependant, ces deux sources se distinguent par quelques aspects. La norme ISO/IEC/IEEE 24765 :2017 est un document officiel publié par l'ISO et l'IEC. Le [SEVOCAB](#) est une ressource ou un outil en ligne qui facilite l'accès aux définitions de la norme ISO 24765 et à d'autres sources similaires. Pour nos travaux nous avons utilisé [SEVOCAB](#) pour la facilité d'accès à sa base de données.

Comme nous allons le voir dans la suite, nous utilisons ces deux sources pour obtenir les [SEA](#) dont nous voulons mesurer la présence dans une [SdMT](#) en cours de développement. Comme l'illustre la figure [4.1](#), la norme ISO/IEC 62304 est utilisée pour créer une liste des termes de génie logiciel. Ces termes seront validés comme étant des [SEA](#) par comparaison avec les termes présents dans [SEVOCAB](#).

1. International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)

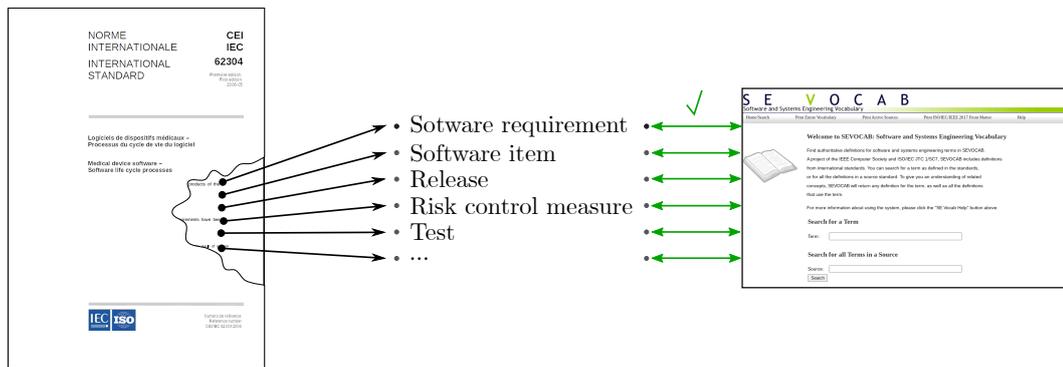


FIGURE 4.1 – Les termes de génie logiciel sont extraits de la norme et comparés aux termes présents dans SEVOCAB. La liste obtenue est ainsi consolidée. On peut ensuite en déduire les SEA.

Ce processus d'extraction à partir de la norme ISO/IEC 62304 se fonde sur nos hypothèses et est décrit dans la suite de ce chapitre. Il mènera à la création d'un score et à la production d'un questionnaire à même d'évaluer la maturité technologique d'une SdMT. Nous considérons la norme ISO/IEC 62304 comme notre référence de maturité et SEVOCAB comme une référence pour le génie logiciel. Pour éviter les éventuelles erreurs de traduction, nous utilisons la version anglaise de ces deux documents de références pour nos travaux.

2.1 La norme ISO/IEC 62304 : un cadre pour le développement logiciel médical

Comme nous venons de le voir, le texte de la norme internationale ISO/IEC 62304 (ISO/IEC 2006) est utilisé comme référence pour révéler les SEA produits lors d'activités d'ingénierie logicielle sur lesquelles le processus de normalisation est construit. Mais avant d'aller plus loin dans la description de ce processus, voyons une brève présentation de cette norme.

Un consortium de professionnels du monde entier et de diverses entreprises a contribué à l'élaboration de cette norme publiée pour la première fois en 2006 (IEC 62304 :2006) et amendée en 2015 (IEC 62304 :2006/Amd 1 :2015).

La norme ISO/IEC 62304 est structurée en neuf clauses et quatre annexes. Nous verrons que seules certaines parties de cette norme seront utilisées lors de notre étude. Les quatre premières clauses détaillent le champ d'application, les références normatives, les termes et définitions, et les exigences générales. Les annexes présentes dans le document de la norme apportent des guides de mise en œuvre et des informations supplémentaires pour utiliser la norme dans un contexte opérationnel. Notamment, ces annexes expliquent les relations de la norme avec d'autres normes.

Les deux premières clauses et la quatrième clause ne concernent pas le domaine du génie logiciel et décrivent le contexte normatif de l'application de cette norme. Nos travaux sont focalisés sur le génie logiciel, par conséquent il n'est pas opportun de les analyser du point de vue du génie logiciel.

La clause 3, intitulée « Terms and definitions », de la norme ISO/IEC 62304, qui traite des termes et définitions, définit un total de 34 termes et expressions pour faciliter la compréhension et l'interprétation des concepts spécifiques dans le contexte normatif.

Pour notre travail, nous utilisons les clauses 5 à 9, qui contiennent toutes les exigences spécifiques et les références aux éléments de génie logiciel qui sont l'objet de cette étude. Nous les détaillons ci-dessous.

- *Clause 5 : PROCESSUS DE DÉVELOPPEMENT LOGICIEL* : elle établit les principes et les exigences pour le développement du logiciel dans le contexte des dispositifs médicaux. Cette clause fournit des directives détaillées sur les activités et les processus associés à la conception, à la mise en œuvre, à la vérification et à la validation du logiciel médical. Elle est constituée de 8 sections, de 53 sous-sections qui donnent 53 exigences sur 9 pages en 3205 mots.
- *Clause 6 : PROCESSUS DE MAINTENANCE LOGICIELLE* : elle traite des principes et des exigences liés à la maintenance et aux modifications du logiciel dans le contexte des dispositifs médicaux. Elle assure ainsi que les activités de maintenance logicielle sont gérées de manière structurée et conforme aux exigences de qualité et de sécurité pour les dispositifs médicaux. Cette clause est constituée de 3 sections, de 7 sous-sections qui donnent 10 exigences sur 2 pages en 576 mots.
- *Clause 7 : PROCESSUS DE GESTION DES RISQUES LOGICIELS* : elle est consacrée à la gestion systématique des risques associés au logiciel des dispositifs médicaux. Cette clause s'aligne sur la norme ISO 14971, qui fournit des principes généraux de gestion des risques pour les dispositifs médicaux. Les principaux éléments abordés incluent la planification de la gestion des risques, l'identification, l'analyse, l'évaluation et le contrôle des risques logiciels. Cette clause est constituée de 4 sections, de 13 sous-sections qui donnent 13 exigences sur 2,5 pages en 762 mots.
- *Clause 8 : PROCESSUS DE GESTION DE LA CONFIGURATION LOGICIELLE* : elle a pour objectif d'établir des principes et des exigences pour la gestion systématique de la configuration du logiciel dans le contexte des dispositifs médicaux. Cette clause vise à garantir la maîtrise, l'identification unique, le contrôle des modifications, et la traçabilité des éléments de configuration du logiciel tout au long de son cycle de vie. Cette clause est constituée de 3 sections, de 6 sous-sections qui donnent 7 exigences sur 2 pages en 480 mots.
- *Clause 9 : PROCESSUS DE RÉOLUTION DES PROBLÈMES LOGICIELS* : elle établit des principes et des exigences pour la résolution systématique des problèmes liés au logiciel des dispositifs médicaux. Elle met l'accent sur l'identification, la documentation, l'évaluation des impacts, la mise en œuvre de correctifs, et la vérification et validation de ces correctifs. L'application de cette clause assure une gestion proactive et efficace des problèmes logiciels, contribuant à maintenir la sécurité et la performance des dispositifs médicaux tout au long de leur cycle de vie. Cette clause est constituée de 8 sections, 8 exigences sur 9 pages en 447 mots.

Ces cinq clauses regroupent ainsi 91 exigences qui sont décrites en 5470 mots. Des notes peuvent compléter les informations données. La figure 4.2 montre

l'exemple de l'exigence décrite à la section 5.6.8 de la clause 5. Chaque exigence a un titre et contient des paragraphes d'explication en replaçant l'exigence dans le contexte de la classe de sécurité concernée. C'est de ce point de vue que nous avons considéré la structure de la norme ISO/IEC 62304. La structure d'une norme en général peut varier selon son domaine d'application et sa nature. La manière d'appréhender les exigences peut, ainsi, varier d'une norme à l'autre. De plus, il est possible qu'une exigence soit potentiellement considérée au même niveau qu'une section. Il est également possible de la considérer au niveau d'un paragraphe comme nous le faisons dans le cadre de cette étude. Notre appréhension de l'exigence dans l'analyse de la norme ISO/IEC 62304 se fait donc au niveau du paragraphe pour toute l'étude.

5.6.8 Use software problem resolution PROCESS

The MANUFACTURER shall enter ANOMALIES found during software integration and integration testing into a software problem resolution PROCESS.

FIGURE 4.2 – Exigence 5.6.8 de la norme ISO/IEC 62304

Le texte des exigences comprend les termes pertinents qui décrivent les activités de développement du logiciel gérées par les processus qualité. La norme ISO/IEC 62304 contient à la fois les activités d'ingénierie logicielle et les activités de contrôle de la qualité. Plus précisément, la norme décrit les activités de génie logiciel de manière régulée. Notre objectif est d'extraire les éléments de génie logiciel du contexte qualité. C'est pourquoi un lexique de référence comme [SEVOCAB](#) est nécessaire. Il nous permet d'isoler les termes spécifiques au génie logiciel des éléments qui sont spécifiquement orientés processus. Nous l'avons donc utilisé comme un guide terminologique, dont on se sert pour la traduction de l'environnement normatif vers un environnement purement lié aux activités du génie logiciel. La section suivante présente cet outil.

2.2 SEVOCAB : un guide terminologique

[SEVOCAB](#) est un projet de l'IEEE Computer Society et de l'ISO/IEC Joint Technical Committee (JTC) 1/SC7. Il compile un lexique largement accepté dans le domaine du génie logiciel. Il se concentre sur des termes et des définitions provenant d'un large éventail de sources reconnues. Il fournit des définitions faisant autorité et regroupe plus de 5000 termes provenant de 188 normes internationales ISO/IEC ou IEEE. Chaque terme dans [SEVOCAB](#) a une ou plusieurs définitions provenant d'une ou plusieurs de ces normes. La table 4.1 donne quelques exemples du type de termes et leurs définitions que l'on peut trouver dans [SEVOCAB](#).

Termes	Définitions
Anomaly	« Anything observed in the documentation or operation of a system that deviates from expectations based on previously verified system, software, or hardware products or reference documents. »
Software Requirement	« Software capability needed by a user to solve a problem or to achieve an objective. Software capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document. »
Software Unit	« Atomic-level software component of the software architecture that can be subjected to standalone testing. »
Specification	<p>① « Detailed formulation, in document form, which provides a definitive description of a system for the purpose of developing or validating the system; information item that identifies, in a complete, precise, and verifiable manner, the requirements, design, behavior, or other expected characteristics of a system, service, or process; »</p> <p>② « A document that specifies, in a complete, precise, verifiable manner, the requirements, design, behavior, or other characteristics of a system, component, product, result, or service and, often, the procedures for determining whether these provisions have been satisfied. Examples are : requirement specification, design specification, product specification, and test specification; concrete representation of a model in some notation. »</p>

TABLE 4.1 – Quelques exemples de termes et leurs définitions tels qu'on peut les trouver dans le lexique SEVOCAB. Certains termes ont plusieurs définitions issues de sources différentes.

Certains termes ont été délibérément exclus du référencement dans [SEVOCAB](#) par ses auteurs. C'est le cas des termes considérés comme spécifiques à un groupe ou à une organisation, appartenant à une entreprise ou à une marque, ou jugés insuffisamment spécifiques. Un exemple concret de cette approche est observé dans le choix de ne pas inclure le terme générique « bug » dans [SEVOCAB](#). Les raisons de cette exclusion résident dans la volonté de privilégier des termes plus précis et spécifiques. En lieu et place, [SEVOCAB](#) propose des alternatives telles que « Error », « Fault », « Failure » ou « Anomaly », qui offrent une granularité plus fine dans la description des problèmes logiciels.

Cette démarche répond à une aspiration à favoriser une utilisation précise du vocabulaire dans le domaine de l'ingénierie logicielle. Prenons l'exemple des termes

« Error », « Fault », et « Failure » bien connu dans les activités de test. Chacun de ces termes, bien que liés à des aspects problématiques du logiciel, revêt une signification distincte dans le contexte de [SEVOCAB](#).

Ainsi, « Error » correspond à une erreur de programmation, c'est-à-dire une action incorrecte ou une mauvaise décision prise par le développeur. Cela peut être une faute de frappe, une erreur de logique, ou un oubli. Tandis que « Fault » se réfère à la manifestation d'une erreur dans le code. C'est une partie du code qui est incorrecte et qui peut potentiellement causer une défaillance.

D'un autre côté, « Failure » est employé pour décrire une dégradation du système dans son ensemble. Lorsque le système ne fonctionne pas comme prévu ou ne remplit pas une ou plusieurs de ses fonctions attendues. Dans le langage courant, on retrouve souvent l'utilisation du mot « bug », un terme unique pour décrire ces concepts.

Le terme « Anomaly » en génie logiciel est utilisé de manière plus générale pour désigner toute sorte de déviation ou d'irrégularité par rapport à ce qui est attendu ou spécifié dans le système. Il peut s'agir d'erreurs, de fautes, de défaillances, ou de toute autre non-conformité qui affectent le fonctionnement, la performance ou le comportement prévu du logiciel.

Ainsi, la sélection méticuleuse des termes dans [SEVOCAB](#) vise à offrir une richesse sémantique tout en éliminant les ambiguïtés associées à des termes génériques tels que « bug ». Cela favorise une communication précise et un échange d'informations plus clair au sein de la communauté de l'ingénierie logicielle.

D'autres termes ont été délibérément exclus du référencement dans [SEVOCAB](#) lorsque leur signification dans le domaine des technologies de l'information pouvait être directement déduite de leur sens ordinaire en anglais. Un exemple concret de cette démarche est illustré par l'exclusion de termes tels que « email » ou « server ». Ces termes, bien que couramment utilisés en anglais, possèdent des significations spécifiques et largement acceptées dans le contexte des technologies de l'information, ne nécessitant pas de définition supplémentaire au sein de [SEVOCAB](#).

Cette approche vise à optimiser la pertinence et la concision du vocabulaire inclus dans [SEVOCAB](#). En évitant d'inclure des termes dont la signification informatique est largement intuitive pour les utilisateurs du domaine, [SEVOCAB](#) se concentre sur des concepts plus spécialisés et sujets à des interprétations variées. Cela facilite également la tâche du mappage, que nous verrons plus loin, des termes entre [SEVOCAB](#) et d'autres référentiels, en réduisant les risques d'ambiguïté ou de redondance lors de l'alignement des terminologies entre différentes sources normatives ou organisations.

Dans cette étude, [SEVOCAB](#) sera utilisé comme référence pour s'assurer qu'un terme relève du domaine technologique du génie logiciel.

3 MÉTHODOLOGIE

Notre démarche s'articule autour d'un processus décrit par la figure 4.3. Il consiste en quatre étapes utilisées pour établir le score de maturité technologique d'une [SdMT](#) à partir de la norme ISO/IEC 62304.

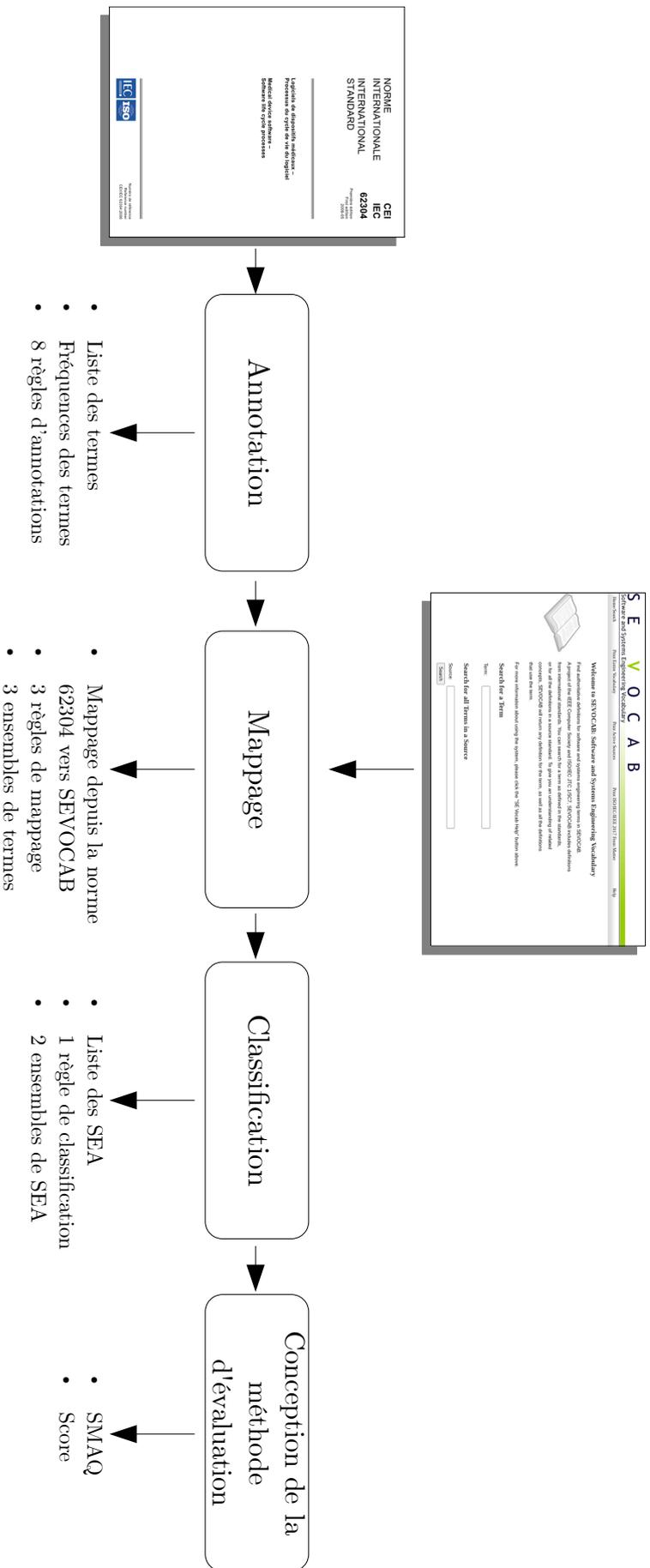


FIGURE 4-3 – Les quatre étapes utilisées pour construire un score de maturité à partir de la norme ISO/IEC 62304 sont les suivantes : annotation manuelle, mappage des termes lexicaux SEVOCAB en liste de SEA, classification en SEA de classe ① ou classe ②, et configuration de l'évaluation qui génère le score de maturité à partir du nombre de citations. À chaque étape, des règles sont appliquées pour garantir la répétabilité des résultats.

- La première étape consiste à annoter tous les mots clés et les termes qui sont pertinents et susceptibles de nous informer sur les activités et les **SEA** contenus dans l'énoncé des exigences de la norme (c'est-à-dire ISO/IEC 62304), comme cela est détaillé dans la section 3.1.
- La deuxième étape consiste à comparer et à mapper les termes annotés avec le lexique du génie logiciel (c'est-à-dire **SEVOCAB**), comme expliqué dans la section 3.2. En utilisant cette référence, il est possible de consolider l'extraction des termes en vérifiant leur appartenance à **SEVOCAB**.
- La troisième étape classe les termes obtenus en deux ensembles correspondant à deux niveaux de **SEA** (classe ❶ ou classe ❷), comme indiqué dans la section 3.3.
- La dernière étape met en place l'évaluation en établissant un score de maturité basé sur la présence des **SEA** classés et leur importance dans la norme mesurée par le nombre de citations, comme détaillé dans la section 3.5.

3.1 Annotation du texte de la norme ISO/IEC 62304

Le processus d'annotation consiste à annoter manuellement et directement le texte de la norme ISO/IEC 62304 à l'aide d'un système de balisage non destructif, comme illustré dans le listing 4.1. Il est important de souligner que, dans cette démarche, nous avons délibérément choisi de ne pas recourir à des outils d'annotation préexistants tels que « BRAT Annotation » ou « Doccano ». Cette décision découle de plusieurs considérations spécifiques à notre approche de recherche.

Tout d'abord, la nature complexe et spécifique du langage utilisé dans le domaine du génie logiciel médical, ainsi que les particularités des termes contenus dans la norme 62304, ont rendu l'utilisation d'outils d'annotation génériques moins adaptée. Ces outils peuvent parfois manquer de la finesse nécessaire pour capturer les subtilités des concepts propres au génie logiciel dans le contexte médical.

En outre, notre processus d'annotation nécessitait une compréhension approfondie des enjeux liés aux **SdMT** et aux normes associées. L'annotation manuelle a permis une immersion directe dans le contenu, favorisant une interprétation précise et contextuelle des termes et concepts.

Enfin, le caractère spécifique de notre étude nécessitait une flexibilité et une adaptabilité constantes tout au long du processus d'annotation. Les outils d'annotation existants peuvent parfois imposer des structures rigides qui limitent la liberté nécessaire pour explorer de nouveaux angles ou pour adapter l'approche en fonction des découvertes faites au cours de l'annotation.

Par conséquent, notre choix délibéré de l'annotation manuelle a été motivé par la nécessité d'une compréhension profonde des nuances du génie logiciel médical, la recherche de flexibilité dans le processus et la capacité à saisir les subtilités contextuelles spécifiques à notre domaine d'étude.

3.1.1 La technique d'annotation

Dans le code présenté dans le listing 4.1, l'article 5.6.8 (Fig. 4.2) de la norme ISO/IEC 62304 est traduit au sein d'un document \LaTeX . De la ligne 1 à 8, une balise `article` est définie en s'appuyant sur la structure proposée par le code \LaTeX :

`\begin{}` et `\end{}`. Son utilisation a été possible grâce à la définition d'un nouvel environnement comme le montre le code fourni par le listing en annexe A.1.

```

1 \begin{article}{5.6.8 Use software \term{problem resolution}
   PROCESS }
2   The MANUFACTURER shall enter \term{ANOMALIES}[ANOMALY] found
   during \term{software integration} and \term{integration
   testing}[INTEGRATION TEST] into a software \term{problem
   resolution} PROCESS.
3   [Class B, C]
4
5   \notebox{
6   NOTE See Clause 9.
7   }
8 \end{article}

```

Listing 4.1 – Balisage des termes pour l'annotation dans le texte de la norme

Pour réaliser l'annotation des termes de génie logiciel, nous avons utilisé la forme suivante :

$$\text{\term{A}[B]}. \quad (4.1)$$

Dans l'expression 4.1, la partie A est le terme faisant partie du texte de la norme ISO/IEC 62304. Il est la partie obligatoire de la commande. La partie B est le terme A auquel on a appliqué les règles d'annotation RA1 à RA8, le cas échéant, que nous décrivons ci-après. La partie B est optionnelle. La figure 4.4 apporte des précisions sur ce point en montrant le passage entre la forme textuelle d'une exigence dans la norme ISO/IEC 62304 et sa forme annotée.

Enfin, le listing en annexe A.2 montre la création de cette commande pour pouvoir indexer les termes de génie logiciel dans le texte de la norme ISO/IEC 62304.

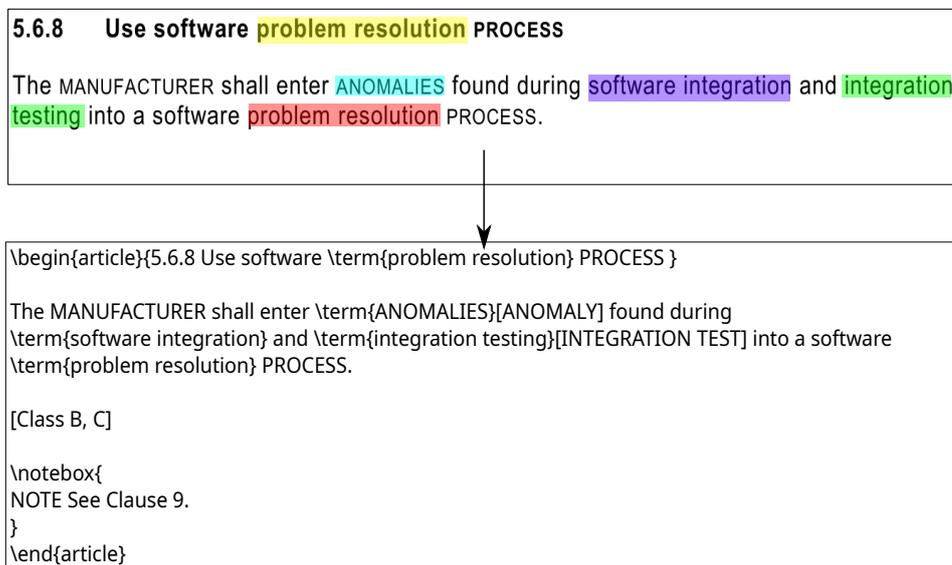


FIGURE 4.4 – Annotation de l'exigence 5.6.8 de la norme ISO/IEC 62304. Pour le terme « ANOMALIES » la règle RA1 est appliquée (voir ci-dessous).

L'utilisation du système d'encodage fourni par \LaTeX nous permet la mise en place et l'utilisation de ce balisage autorisant un traitement ultérieur des termes

retenu, en conservant une référence au texte original. De plus, grâce à un script Python ad hoc nous pouvons extraire ces annotations et les stocker pour une exploitation ultérieure en base de données.

L'objectif de cette première étape est d'annoter tous les termes de génie logiciel cités dans les phrases de titre et les énoncés des exigences. Par exemple, les termes suivants ont été annotés dans la section 5.6.8 (voir Fig. 4.2) : « Software Integration », « Integration Test » et « Problem Resolution ». Afin de consolider l'annotation et de s'assurer que cette étape manuelle est reproductible, sept règles ont été appliquées au cours du processus d'annotation.

Règle d'annotation RA₁ : utilisation de la forme singulière.

Tous les termes sont annotés dans leur forme au singulier. Par exemple, dans le texte de l'ISO/IEC 62304, le terme « Software requirement » et sa forme plurielle « Software requirements » ont la même signification. Nous utilisons « Software requirement » pour annoter les occurrences, comme c'est illustré dans la figure 4.4, où le terme « ANOMALIES » est annoté selon cette règle.

Règle d'annotation RA₂ : différenciation des termes ayant des significations distinctes, même avec des mots communs.

Les mêmes termes sont annotés de manière distincte lorsque différentes significations sont impliquées, même s'ils partagent un sous-ensemble commun de mots. Par exemple, « Integration Test Plan » et « Integration Test Report » partagent deux mots, mais en raison de leurs significations distinctes, ils sont annotés comme deux termes différents. Cette règle est illustrée dans la figure 4.4, où le terme « integration » dans « Software integration » et « integration testing » est annoté dans son contexte complet.

Règle d'annotation RA₃ : termes ignorés.

Certains termes ne sont pas annotés pour les raisons suivantes :

- **RA_{3.1}** les termes donnés en exemple : dans certaines exigences de la norme, notamment dans les notes à la fin des sections, des termes sont illustrés par des exemples afin de clarifier leur signification dans un contexte technique. Par exemple, l'exigence 5.2.2 comporte une note illustrant les termes « functional and capability requirements » (« Exigences fonctionnelles et de capacité ») avec des exemples. Les termes faisant partie de ces exemples ne sont pas annotés. Cette décision repose sur le fait que ces exemples sont des illustrations. Cependant ces exemples sont utilisés lors du mappage et de la classification pour fournir une meilleure compréhension du terme.
- **RA_{3.2}** les termes se référant à l'ensemble du système ou à l'objet de l'étude : « Medical Device », « Software System », « Software », « Hardware », « Software System Development », « Hardware Development », et « Software Development ». Ces termes représentent des concepts trop vagues dans le contexte de la norme pour être interprétés de manière univoque en tant que [SEA](#).

- **RA3.3** les termes relatifs au développement dans le cadre de l'assurance qualité et n'entrant pas dans le champ d'application de la présente étude : « Process », « Activity », « Task », et « Milestone » (« Jalon » en français). Tous ces termes décrivent la manière de mener des activités de management. Ces termes ne sont pas spécifiques au génie logiciel, mais plutôt issus d'activités de planification.
- **RA3.4** les termes décrivant les acteurs et les parties prenantes : « Manufacturer » et « User ». Les acteurs décrits dans la norme ne sont pas pertinents pour notre étude, qui se concentre sur les [SEA](#) de la norme ISO/IEC 62304.

Règle d'annotation RA4 : lorsque des termes non annotables, car exclus (RA3), sont utilisés comme qualificatifs, seul le terme qualifié est annoté.

Par exemple, « Medical Device Software Architecture » est annoté « Software Architecture ».

Règle d'annotation RA5 : lorsqu'une déclaration utilise la structure « doit + verbe + objet », la forme substantive du verbe doit être annotée.

Par exemple, dans la phrase « *the MANUFACTURER shall specify the SYSTEM hardware and software...* », les termes « System Hardware Specification » et « System Software Specification » sont annotés. Cette annotation permet d'identifier les [SEA](#) requis par la norme.

Règle d'annotation RA6 : forcer l'annotation qualifiée.

Les termes « Traceability », « Interface », « Verification », « Documentation », et « Test » sont trop génériques et peuvent faire référence à trop de [SEA](#) s'ils sont utilisés seuls. L'annotation de ces mots doit inclure la description de leur cible. Par exemple, dans l'énoncé : « *The MANUFACTURER shall test the integrated SOFTWARE ITEMS* », le contexte fournit les informations permettant de qualifier la cible du test, le terme annoté est donc « Software Item Test ».

Règle d'annotation RA7 : pour une activité simple, non désignée par les mots processus/tâche/activité dans le texte, le SEA résultant de cette activité est annoté

Par exemple, dans le titre « *5.1 Software development planning* », « Software Development Plan » est annoté.

Règle d'annotation RA8 : les références explicites aux activités de gestion et qui ne sont pas des termes de planification sont annotées

Par exemple dans la déclaration suivante : « *The plan shall address :*

- ...

- *software configuration and change management, including SOUP CONFIGURATION ITEMS and software used to support development ; and*
- ...

» , « Change Management » est annoté comme une spécification.

3.1.2 Remarques sur l'annotation

Un script permet l'insertion automatique dans une base de données des informations associées suivantes : les références aux numéros de paragraphe, la section où l'annotation a été effectuée et le texte source du terme annoté. Cette approche garantit une conservation exhaustive des données contextuelles, permettant ainsi une plus grande facilité d'accès pour des analyses ultérieures.

La cohérence de chaque terme annoté a également été vérifiée. Ainsi, un terme est annoté de la même manière pour toutes les occurrences dans les différentes sections de la norme. De même que tous les éléments de texte identiques sont annotés par les mêmes termes. Cette étape d'annotation fournit une liste de termes normalisés qui peuvent être regroupés pour chaque section.

On peut remarquer que les règles en elles-mêmes sont complexes et que l'utilisation d'un système manuel est adaptée. L'utilisation d'un système automatisé (comme l'utilisation de technologies d'intelligences artificielles) entrainerait l'émergence de nouvelles problématiques qui n'ont pas d'intérêt directement lié avec les travaux menés pour cette étude. Toutefois, elles pourraient être une piste en soi pour la généralisation de notre méthodologie, à d'autres domaines. Elle ne semble pas simple à mettre en œuvre facilement, même si l'émergence des LLM depuis un peu plus d'un an laisse envisager cette possibilité.

3.1.3 Les résultats de l'annotation

Au total, 165 termes différents ont ainsi été annotés dans le texte de la norme ISO/IEC 62304. Ces termes sont présentés dans l'annexe A.2 à la table A.1. Certains termes ont été cités plus d'une fois, le total représente 537 annotations (ou citations) cumulées, avec 131 citations dans les titres et 406 citations dans le corps des articles.

3.2 Mappage

L'étape de mappage consiste à rechercher une correspondance entre les termes annotés et les termes de génie logiciel tels qu'ils figurent dans SEVOCAB. L'objectif de cette étape est de mettre en correspondance les termes trouvés dans la norme avec des termes qui représentent des SEA et qui sont donc susceptibles d'être trouvés dans une SdMT. Il existe trois façons différentes de mapper le terme annoté avec un terme SEVOCAB, ce qui donne trois règles de mappage et trois ensembles différents (voir la figure 4.14 page 92).

3.2.1 Règles de mappage

- **Règle de mappage RM1 : mappage direct** la première règle repose sur une mise en correspondance terme à terme avec la base de données SEVOCAB.

Cette règle fournit l'ensemble \mathcal{D} des termes en correspondance directe avec les termes fournis par SEVOCAB.

Si un terme annoté est défini tel quel dans SEVOCAB, il est inclus dans l'ensemble \mathcal{D} . Dans le cas où aucun mappage direct ne peut être effectué pour un terme donné, la règle RM2 est appliquée.

- **Règle de mappage RM2 : mappage des combinaisons.** pour les termes annotés composés de plus d'un mot (termes annotés multimots), une correspondance avec SEVOCAB est recherchée pour chaque sous-combinaison de mots, en donnant la priorité aux combinaisons les plus longues. Cette règle fournit l'ensemble \mathcal{C} des termes annotés qui peuvent être trouvés en combinant deux ou plusieurs termes trouvés dans SEVOCAB.

Si les définitions *combinées* dans SEVOCAB ont le même sens, alors le terme annoté est mappé avec la combinaison de termes SEVOCAB.

Par exemple, le terme annoté « Software System Requirement » n'est pas un terme définis dans SEVOCAB, mais « Software », « System », « Requirement », « Software System » et « Software Requirement » sont des termes définis dans SEVOCAB. La combinaison des définitions de « Software System » et de « Requirement » aboutit à une signification équivalente à celle du terme annoté, telle qu'on peut la déduire du contexte du terme dans la norme ISO/IEC 62304 ou de la définition donnée dans la clause 3 de la même norme. Dans cet exemple, une correspondance de combinaison est établie.

Si un terme annoté peut être mappé par une combinaison de termes et de définitions, il a été inclus dans l'ensemble \mathcal{C} . Sinon la règle RM3 est appliquée.

- **Règle de mappage RM3 : mappage manuel** tous les termes annotés qui n'appartenaient pas à \mathcal{D} ou \mathcal{C} ont été pris en compte lors du mappage manuel. Ces termes restants étaient soit liés au contexte du contrôle de la qualité, soit spécifiques à la norme. L'application de cette règle fournit l'ensemble \mathcal{M} des termes annotés qui ont une forme généralisée ou une équivalence de sens dans SEVOCAB obtenue par une évaluation manuelle.

Cette action manuelle a nécessité de comparer une à une les définitions de toutes les sources disponibles (normes et SEVOCAB) de chaque terme pour être à même de proposer une équivalence afin de choisir le bon terme.

Par exemple :

- le terme annoté « Problem Correction » n'a pas de correspondance directe ou combinée (« Problem » est un terme SEVOCAB, mais « Correction » ne l'est pas). Ce terme est donc mappé manuellement avec le terme « Problem Repair » défini dans SEVOCAB pour lequel on trouve une signification équivalente à celle donnée dans la norme.
- Le terme annoté « SOUP »² est mappé avec « Library ». Nous procédons, ici, par généralisation. En effet, « SOUP » entre bien dans le champ sémantique de « Library » dans le contexte de la recherche.

2. (sigle pour l'anglais « Software Of Unknown Provenance ») logiciel de provenance inconnue : élément logiciel qui est déjà développé, et généralement disponible, et qui n'a pas été développé pour être incorporé dans le dispositif médical (également appelé « logiciel de série ») ou logiciel précédemment développé pour lequel des enregistrements suffisants des processus de développement ne sont pas disponibles.

Tous les termes annotés mappés manuellement sont inclus dans l'ensemble \mathcal{M} .

À la fin du processus de mappage, tous les termes annotés se sont retrouvés dans l'un des trois ensembles qui mettent en correspondance leur signification avec un **SEA**. Certains termes mappés, en particulier dans l'ensemble \mathcal{M} , peuvent correspondre à plus d'un terme annoté. Par exemple, « SOUP Designator » et « SOUP Identification » sont tous les deux mappés avec « Library Identifier ».

L'étape de mappage permet 1) l'extraction de tous les **SEA** inclus dans les exigences de la norme sans tenir compte des exigences de la réglementation en matière de contrôle de la qualité, et 2) la traduction des termes d'un vocabulaire spécifique au contrôle de la qualité ou à la norme en un vocabulaire d'ingénierie logicielle plus susceptible d'être compris par les acteurs de la phase de recherche.

Finalement, à l'issue de cette étape de mappage, chaque terme mappé est un **SEA** pouvant être produit par une activité de génie logiciel. À partir de maintenant, nous utiliserons donc le terme « **SEA** » au lieu du terme « **terme mappé** ».

3.2.2 Résultats du mappage

L'étape de mappage a permis d'obtenir un ensemble de 156 **SEA**, ils sont présentés dans la table A.2 de l'annexe A.3.

La figure 4.5 présente la répartition des **SEA** selon les trois ensembles de mappage \mathcal{D} , \mathcal{C} et \mathcal{M} . La plus grande proportion de **SEA**, soit 57,2 % (95 **SEA**), a été obtenue grâce au mappage **Combiné**. Les proportions de mappage **Direct** avec 21,7 % (36 **SEA**) et le mappage **Manuel** avec 21,1 % (35 **SEA**) sont égales.

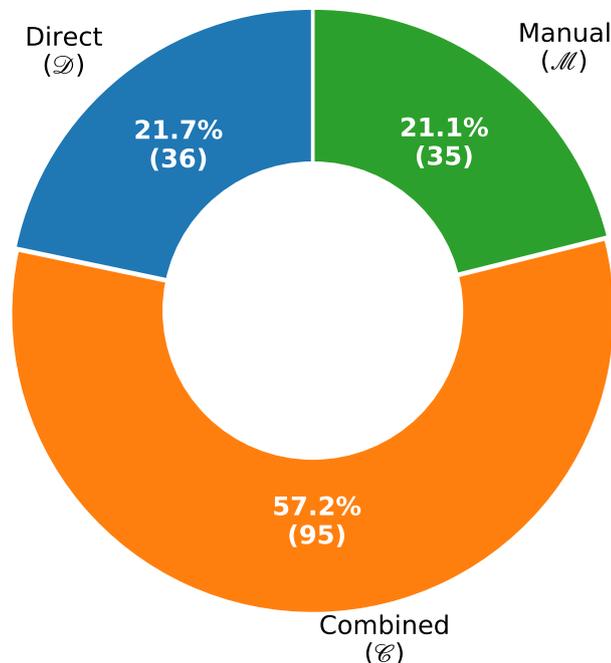


FIGURE 4.5 – Répartition des SEA parmi les ensembles de mappage \mathcal{D} , \mathcal{C} et \mathcal{M} .

3.2.3 Nombre de citations

À partir des annotations et de leur fréquence de citations, il est possible de déduire la fréquence d'apparition des **SEA** dans les exigences de la norme ISO/IEC 62304. Le nombre total de citations qui est le nombre cumulé de toutes les citations des annotations est de 537.

Pour un **SEA** donné, son nombre de citations s'interprète comme étant la somme du nombre de citations (occurrences) du *terme* annoté correspondant. La figure 4.6 montre la distribution de ces nombres de citations pour les **SEA**. On peut remarquer que 69 **SEA**, représentant 44,2 % du nombre de **SEA** total, ne sont cités qu'une seule fois dans le texte des clauses de la norme que nous avons considérées. 60 **SEA**, représentant 38,5 % des **SEA**, ont un nombre de citations compris entre 2 et 4. Ces deux groupes représentent ensemble 82,7 % des **SEA**. 17 **SEA**, représentant 10,9 % des **SEA**, ont un nombre de citations compris entre 5 et 10, tandis que seulement 10 **SEA**, représentant 6,4 % des **SEA**, ont un nombre de citations supérieur à 10. Ensemble, ces 27 **SEA** les plus cités représentent 17,3 % des **SEA**.

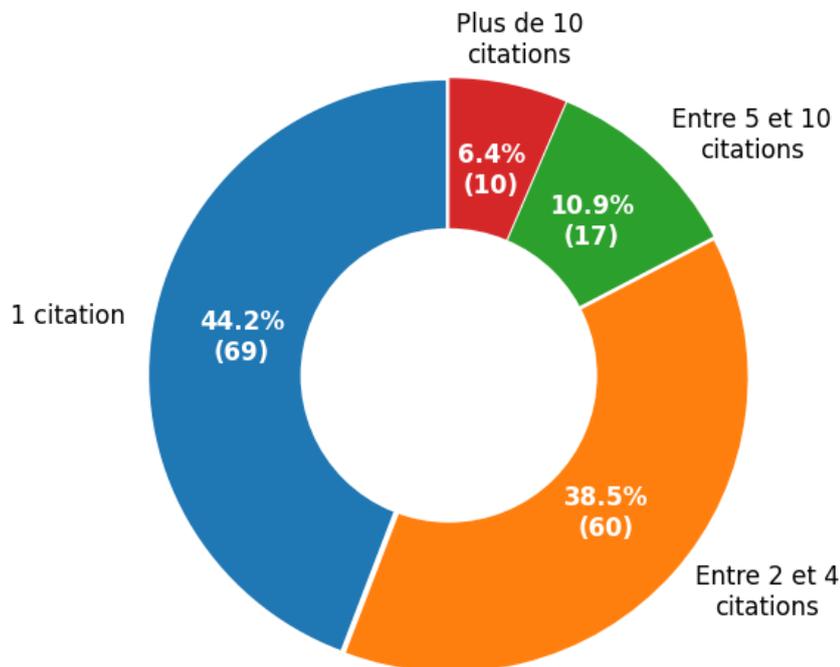
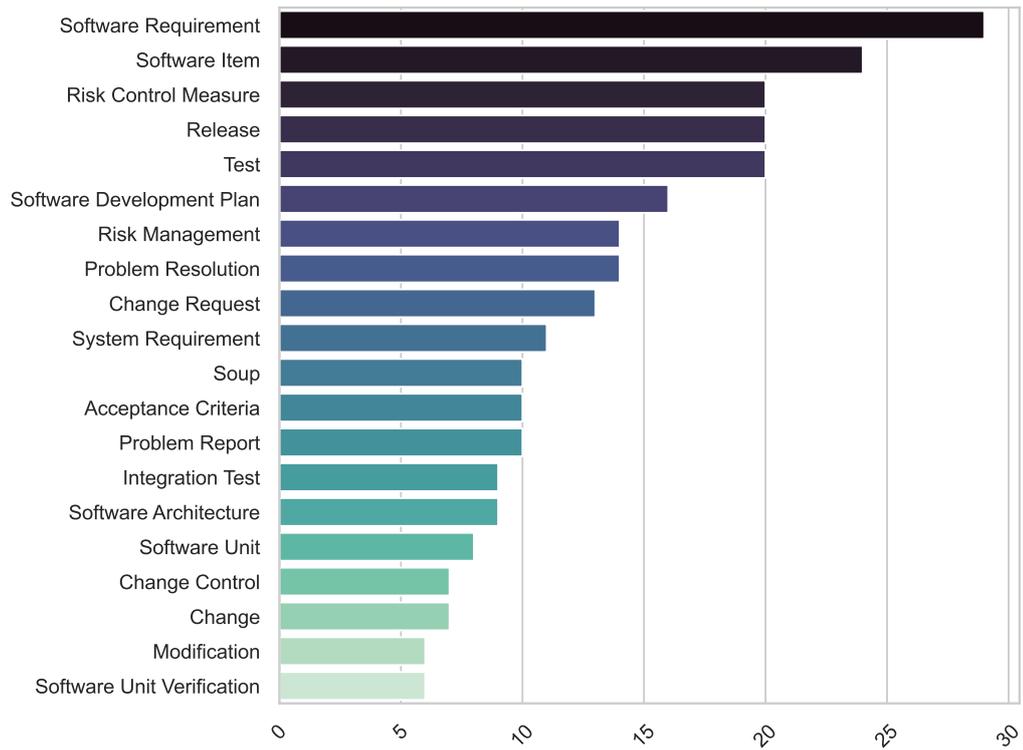
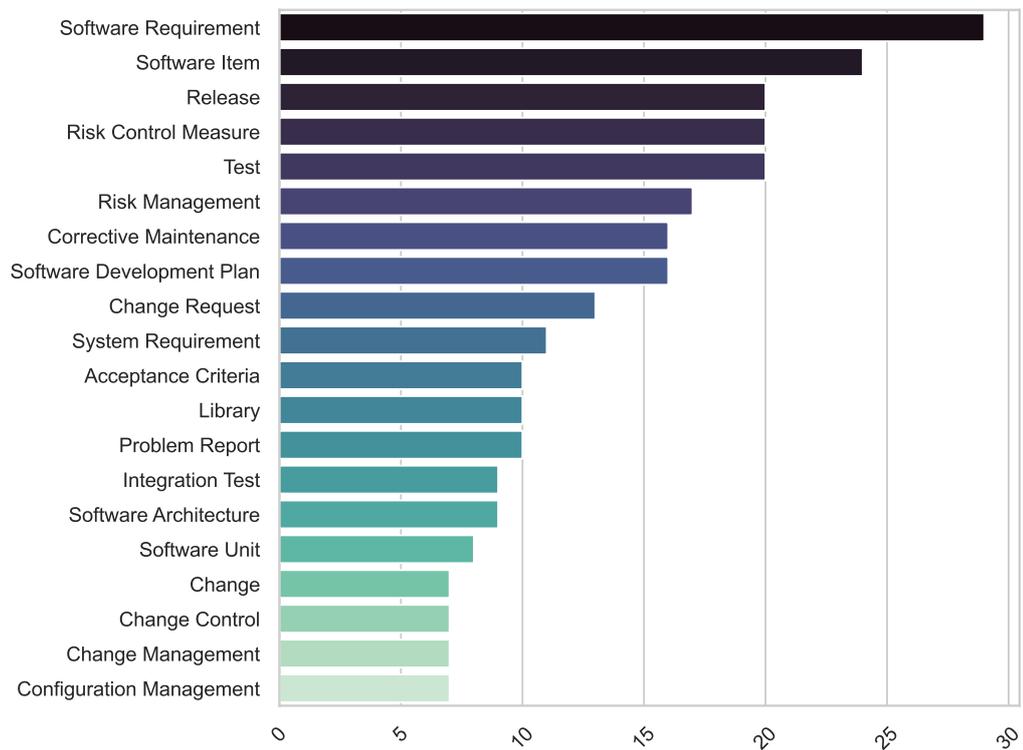


FIGURE 4.6 – Répartition des SEA en fonction de leur nombre de citations.

La figure 4.7a présente les 20 termes annotés les plus cités, tandis que la figure 4.7b présente les 20 **SEA** les plus cités. Ces 20 **SEA** les plus cités représentent 50,3 % du nombre total de citations, avec un nombre total de citations égal à 270. Nous voyons dans cette figure que la liste des **SEA** est pratiquement équivalente à la liste des termes annotés les plus cités.



(a) Les 20 termes annotés les plus cités



(b) Les 20 SEA les plus cités

FIGURE 4.7 – Termes annotés et SEA avec le plus grand nombre de citations.

3.3 Classification des SEA

La phase de classification des SEA représente une étape importante. Cette classification a pour objectif de distinguer les SEA qui apparaissent tout au long du développement logiciel et à éclairer leur évolution au fil de ce processus pendant le projet de recherche.

Comme exposé précédemment dans la section 2 du chapitre 3 (voir page 60), ces SEA sont regroupés en deux classes fondamentales, à savoir les SEA ① et les SEA ②. Ces catégories sont définies selon des critères distincts reflétant la nature de leur élaboration et de leur formalisme.

La classe ① englobe les SEA qui ont une représentation syntaxique plus formelle plus proche de la forme syntaxique finale. Ils sont dans une structure syntaxique qui limite déjà le pragmatisme. Ces SEA constituent une première strate dans le processus de développement.

En contraste, la classe ② regroupe les SEA dont la représentation syntaxique est plus proche du langage naturel. Cette classe de SEA se trouve plus proche de la documentation, des aspects réglementaires et de la spécification sans toutefois entrer dans un formalisme proche de celui de la norme au moment de leur production en phase recherche. Ils nécessiteront un processus de raffinement plus important. Au cours du projet de recherche, ces SEA apparaissent progressivement sous une forme de plus en plus structurée. In Fine, ils représentent une phase avancée du processus de développement, exigeant une formalisation claire des spécifications en cohérence avec les exigences (réglementaires par exemple).

3.3.1 Règles de classification

Nous proposons d'utiliser des règles de classification systématique. Nous voyons, ainsi, apparaître deux règles :

- **Règle de classification RC1** : les SEA se terminant par des termes tels que « Documentation », « Evaluation », « Analysis », ou « Management » sont catégorisés comme SEA de classe ②. Ils ont un lien étroit avec des processus nécessitant une expertise spécialisée et une documentation rigoureuse. Les SEA se terminant par des mots décrivant une forme liée à une spécification ou à une documentation structurée. Par exemple, la mention « Software Architecture Documentation » ou « Software Risk Analysis » guide cette classification vers les SEA de classe ②.
- **Règle de classification RC2** : Les autres SEA sont classés dans le type ①.

3.3.2 Résultats de la classification

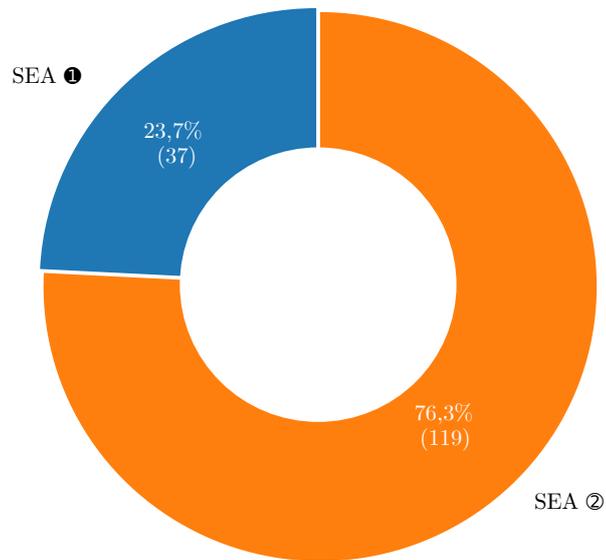
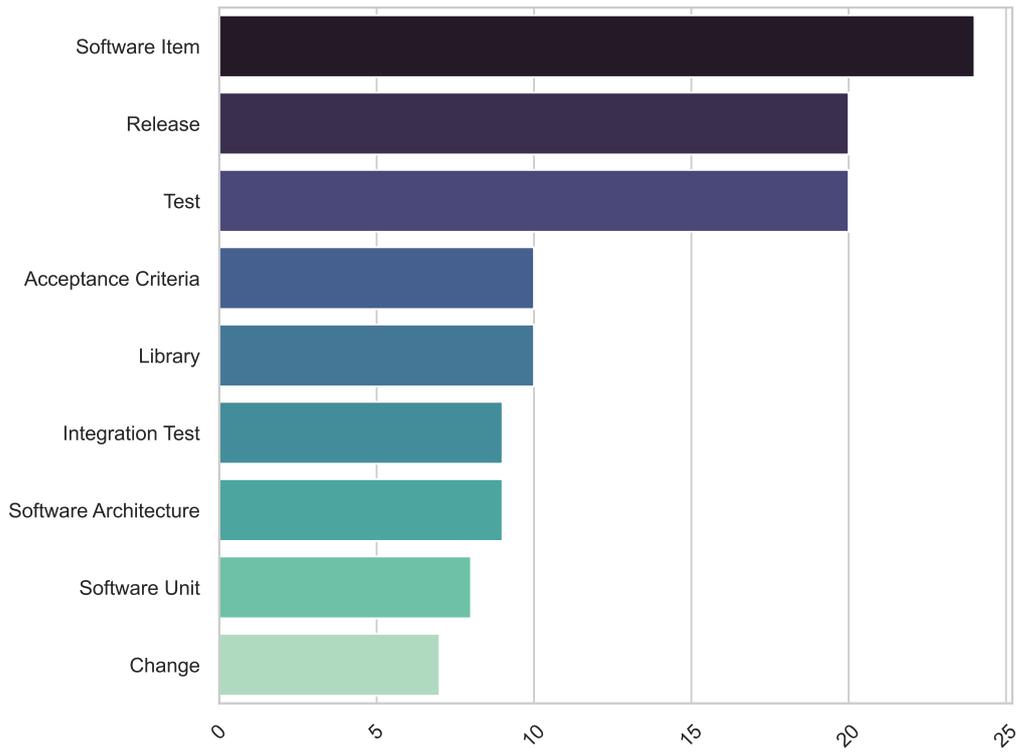


FIGURE 4.8 – Répartition des SEA entre les classes ① et ② .

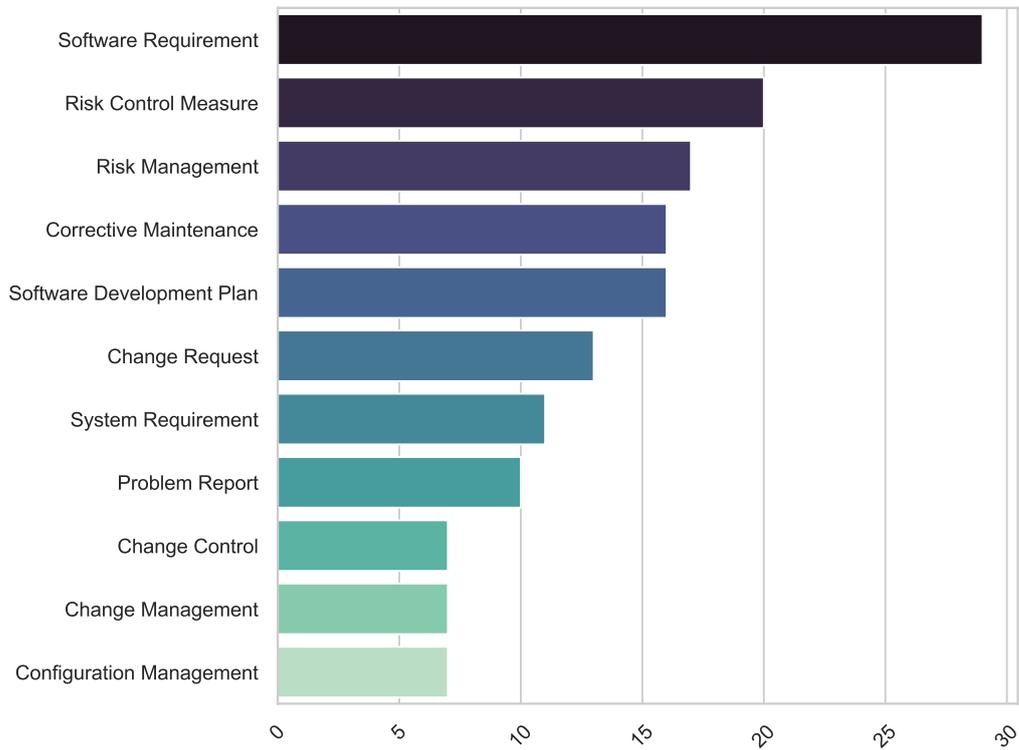
Comme le montre la figure 4.8, l'étape de classification a subdivisé la liste des SEA issus de l'étape de mappage en deux groupes formés par la classe ① contenant 37 éléments ($|①| = 37$), représentant 31,1% des SEA et la classe ② contenant 119 éléments ($|②| = 119$), représentant 76,3% des SEA.

3.4 Synthèse des étapes d'annotation, de mappage et de classification

La figure 4.9 présente les SEA de classe ① et ② parmi les 20 SEA les plus cités. La figure 4.9a montre que 9 des SEA les plus cités correspondent à des SEA de la classe ①, avec un nombre de citations atteignant presque 25 pour le plus cité (« Software Item »). Comme le montre la figure 4.9b, il y a deux SEA de plus pour la classe ② parmi les 20 SEA les plus cités, avec 11 SEA et un nombre de citations dépassant 25 pour le plus cité « Software Requirement ».



(a) Les SEA les plus fréquemment cités de la classe ① parmi les 20 SEA les plus cités.



(b) Les SEA les plus fréquemment cités de la classe ② parmi les 20 SEA les plus cités.

FIGURE 4.9 – Liste des SEA de classe ① et de classe ② correspondant aux 20 SEA les plus cités, toutes classes confondues.

La figure 4.10 montre la répartition des nombres de citations des SEA en fonction de l'emplacement des termes annotés qui leur correspondent dans les clauses de la norme ISO/IEC 62304 ainsi que leurs répartitions par classe. La clause 5, la clause la plus longue de la norme, contient le plus grand nombre de citations, avec un total de 326 citations, ce qui représente 61% du nombre total de citations.

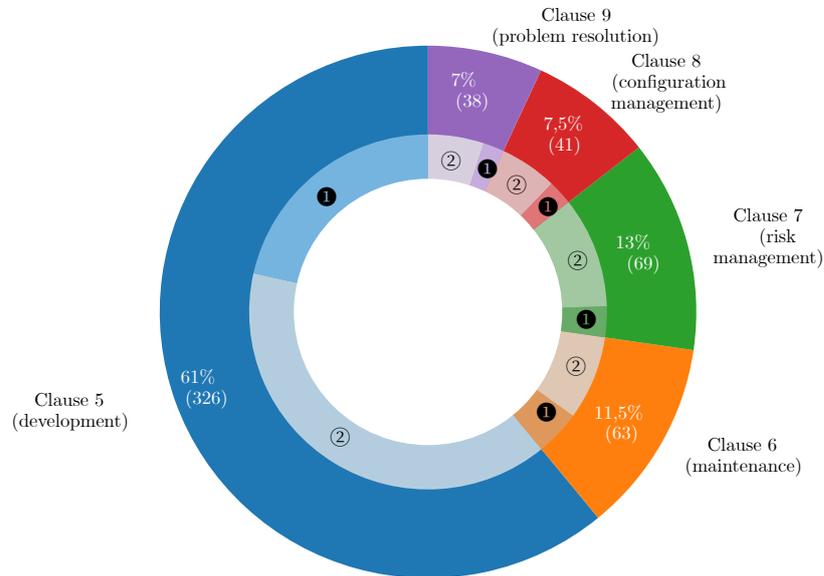


FIGURE 4.10 – Proportion de citations par clause (diagramme circulaire externe) et répartition entre les SEA 1 et les SEA 2 pour chaque clause de la norme (diagramme circulaire interne).

La figure 4.11 montre la répartition du nombre de citations des 156 artefacts parmi les cinq clauses décrites dans la norme ISO/IEC 62304.

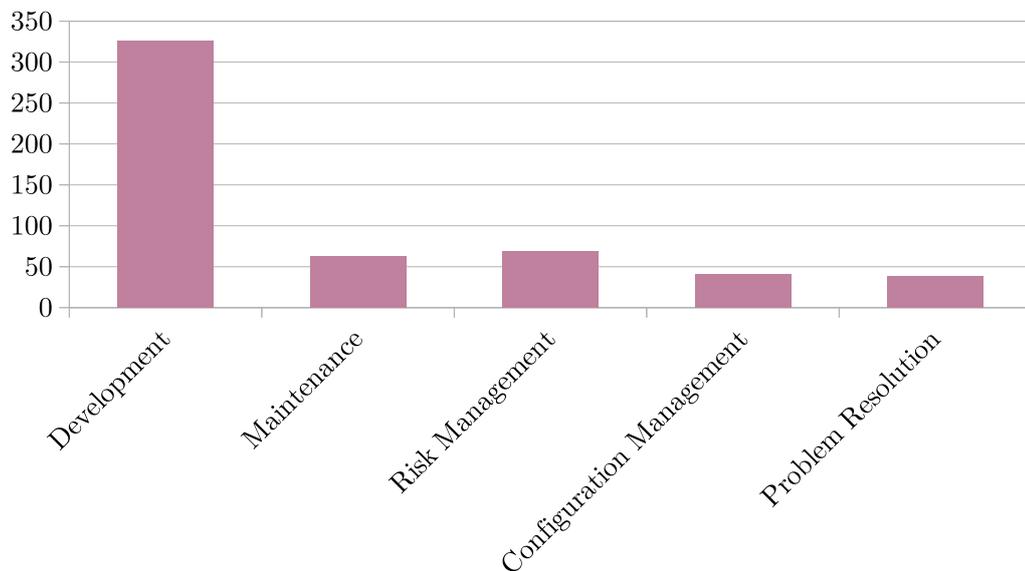


FIGURE 4.11 – Nombre et pourcentage de citations pour chaque clause de la norme.

3.5 Conception de la méthode d'évaluation du score de maturité

Notre méthode d'évaluation du score de maturité repose sur un score calculé selon les réponses données à un questionnaire d'évaluation. Nous expliquons ici, la conception de ces deux outils.

3.5.1 Le score

Pour calculer le score de maturité technologique d'une **SdMT** nous nous basons sur le nombre de citations des termes extraits de la norme. Au cours de l'étape d'annotation, le nombre de citations de chaque terme annoté a été mesuré. Le nombre de citations des termes annotés correspond au nombre de fois où un même terme est annoté. Le nombre de citations d'un **SEA** est la somme des citations de chaque terme annoté avec lequel il est en correspondance. En soutien de notre hypothèse 4 (voir section 1 page 67) le nombre de citations d'un **SEA** est utilisé comme indicateur de l'importance du **SEA** correspondant pour la norme : plus le **SEA** est cité dans la norme, plus le **SEA** est considéré essentiel pour pouvoir entrer dans le processus guidé par la norme ISO/IEC 62304. On note $\mathbf{1}_i$ le i^e **SEA** de la classe $\mathbf{1}$ et $\mathbf{2}_i$ le i^e **SEA** de la classe $\mathbf{2}$.

Le score de maturité *Score* peut être exprimé comme suit :

$$\text{Score} = \frac{1}{N} \left(\sum_{i=1}^{|\mathbf{1}|} w_{\mathbf{1}_i} [\mathbf{1}_i] + \sum_{i=1}^{|\mathbf{2}|} w_{\mathbf{2}_i} [\mathbf{2}_i] \right)$$

où

$$[\mathbf{1}_i] = \begin{cases} 1 & \text{si } \mathbf{1}_i \text{ présent dans la SdMT} \\ 0 & \text{si } \mathbf{1}_i \text{ non présent dans la SdMT} \end{cases} \quad (4.2)$$

et

$$[\mathbf{2}_i] = \begin{cases} 1 & \text{si } \mathbf{2}_i \text{ présent dans la SdMT} \\ 0 & \text{si } \mathbf{2}_i \text{ non présent dans la SdMT} \end{cases}$$

Avec $w_{\mathbf{1}_i}$, respectivement $w_{\mathbf{2}_i}$, désigne le nombre de citations de $\mathbf{1}_i$, respectivement le nombre de citations de $\mathbf{2}_i$; $[\mathbf{1}_i]$, respectivement $[\mathbf{2}_i]$, prend la valeur 1 si $\mathbf{1}_i$, respectivement $\mathbf{2}_i$, est un **SEA** existant à l'état courant de développement de la **SdMT** étudiée, ou 0 s'il n'existe pas au moment de l'évaluation; N est le nombre total de citations ($N = \sum w_{\mathbf{1}_i} + \sum w_{\mathbf{2}_i}$); et $|\mathbf{1}|$, respectivement $|\mathbf{2}|$, le nombre de **SEA** de type $\mathbf{1}$, respectivement de type $\mathbf{2}$.

3.5.2 Application du score dans le contexte de la norme ISO/IEC 62304

Dans cette étude, l'étape d'annotation a donné lieu à un nombre total de 537 termes annotés contenant plusieurs occurrences de mêmes termes. La totalité de ces occurrences constitue le nombre total de citations $N = 537$. Le score de maturité est calculé en utilisant le poids de chaque **SEA** ($w_{\mathbf{1}_i}$ et $w_{\mathbf{2}_i}$) présent dans la **SdMT** évaluée.

Dans la table 4.2 on présente un exemple de calcul pour une SdMT fictive, ne contenant que certains des 20 SEA les plus cités (pour plus de clarté, tous ne sont pas visibles). On observe que le score obtenu est de 46,5%.

Classe	SEA	Nb de citations w_i	Poids $\frac{w_i}{N}$	Dans la SdMT [i]	Score
①	Software Item	24	0,045	1	4,5%
①	Release	20	0,037	0	0%
①	Test	20	0,037	1	3,7%
①
②	Software Requirement	29	0,054	1	5,4%
②
Total					46,5%

TABLE 4.2 – Exemple du calcul du score pour une SdMT fictive contenant certains des 20 SEA les plus cités.

Par exemple, le SEA « Release » de la classe ① a un poids de $w_{\text{①}_i} = 20$. Par conséquent, si une SdMT a un SEA correspondant à une version logicielle (c'est-à-dire, « une version logicielle mise formellement à la disposition d'une communauté plus large » comme l'indique la définition de SEVOCAB), le score de maturité de la SdMT est augmenté de $w_{\text{①}_i}/N = 20/537 = 3.7\%$.

Lorsque les 156 SEA sont présents dans la SdMT, le score de maturité est de 100%. Lorsque tous les 20 SEA les plus cités sont présents seuls, le score de maturité obtenu est de 50,2%. Pour déterminer si un SEA est présent, nous pourrions utiliser une question fermée (voir section 3.5.3).

Par exemple, le SEA le plus cité, « Software Requirement », est associé à la question fermée suivante : « Avez-vous spécifié et documenté les exigences logicielles de votre SdMT? », dont on déduira de la réponse la présence ou l'absence dans la SdMT.

REMARQUE À PROPOS DU NOMBRE DE CITATIONS

Dans la figure 4.12, l'évolution du score de maturité est tracée en fonction du nombre de SEA présent dans la SdMT en les classant par ordre de poids. Sur l'axe des abscisses, les SEA sont disposés dans l'ordre décroissant du nombre de citations, de sorte que les SEA les plus cités sont positionnés en premier et les moins cités en dernier. L'axe des ordonnées décrit le pourcentage de maturité que la SdMT peut atteindre en fonction du nombre de SEA présents.

La courbe bleue correspond à l'évolution de la maturité en ne considérant que les SEA de la classe ①. Le marqueur bleu indique les neuf premiers SEA de la classe ①, correspondant à ceux de la figure 4.9a.

La courbe orange correspond à l'évolution de la maturité en ne considérant que les SEA de la classe ②. Le marqueur orange indique les onze premiers SEA de la classe ②, correspondant à ceux de la figure 4.9b.

Enfin, la courbe verte correspond à l'évolution de la maturité en considérant à la fois les SEA des deux classes. Le marqueur vert indique les vingt premiers SEA, comme indiqué dans la figure 4.7b. Il montre que les vingt SEA les plus cités représentent environ 50% du score de maturité (50,2%).

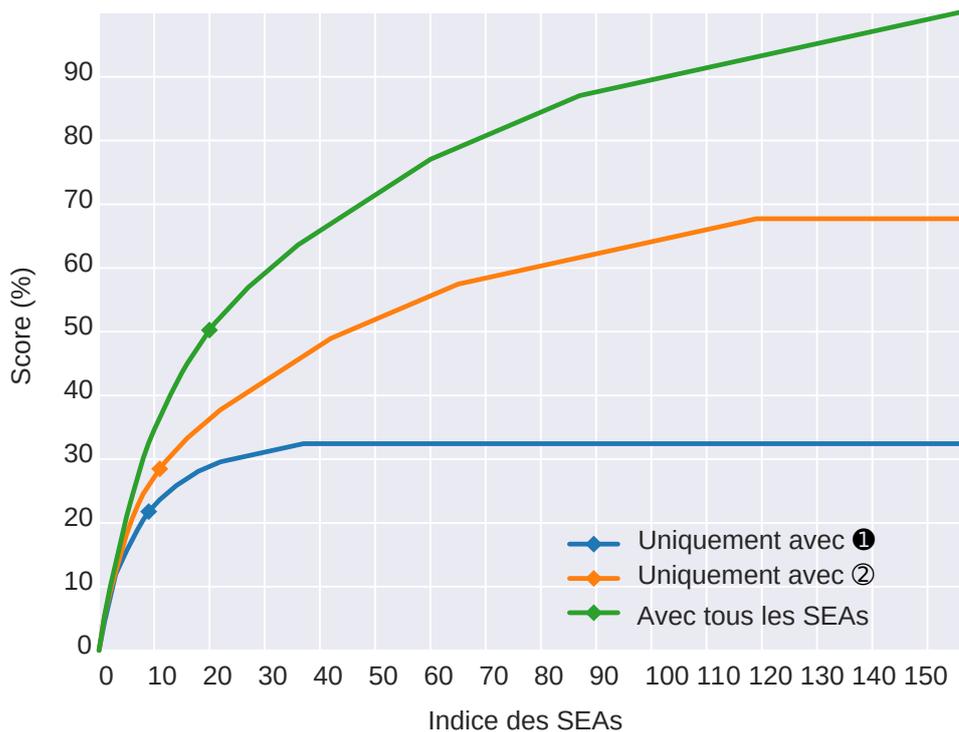


FIGURE 4.12 – Score de maturité en fonction du nombre de SEA de classe ① et de SEA de classe ② présent dans la SdMT, classé par ordre décroissant du nombre de citations.

La figure 4.12 fait apparaître clairement la disproportion entre les SEA à faible fréquence et les autres qui sont minoritaires.

3.5.3 Le questionnaire

Un questionnaire appelé SdMT Maturity Assessment Questionnaire (SMAQ)³ peut être dérivé directement à partir des SEA en construisant des questions fermées pour évaluer la valeur de la présence ou de l'absence des SEA (représentée par $[①_i]$ et $[②_i]$ dans l'équation 4.2).

Lors de la phase de conception de la méthode d'évaluation de la maturité technologique d'une SdMT, nous avons développé un questionnaire reprenant l'ensemble des 156 questions. Il s'agit d'un questionnaire (figure 4.13) spécifique à l'évaluation de ce type de projet. Le SMAQ repose sur un ensemble de questions fermées proposées en anglais et fondées sur la vérification de la présence ou de l'absence d'un SEA donné. Chacune des questions est accompagnée d'une définition construite à partir des définitions proposées dans SEVOCAB. Le nombre de SEA extraits de la norme détermine le nombre de questions. En répondant « oui » à une question, la personne sollicitée indique que le SEA concerné est présent dans la SdMT évaluée. La table 4.3 présente l'exemple de la question pour le SEA « Test ». Le répondant

3.  Le Questionnaire SMAQ

doit se prononcer par « oui » ou par « non » ou par « Je ne sais pas ». La réponse « Je ne sais pas » permet de ne pas bloquer le répondant s'il ne connaît pas le SEA et sera considérée comme une réponse négative.

Software-driven Medical Technology: Maturity Check List

The questions of this survey helps you to evaluate the maturity of your Software-driven Medical Technology.

UGA University Grenoble Alpes U3O

Development Maintenance Risk Management Configuration Man... Problem Resolution

Development
Questions related to the Development activities (mainly covers chapter 5 of the ISO/IEC 62304 standard)

1. Please enter your name *

John Doe

Have you performed any tests? *
Test is an activity in which a system or component is executed under specified conditions, the results are observed or recorded, and an evaluation is made of some aspect of the system or component.

Yes No I don't know

→ Questions about test

Have you performed any integration test? *
An integration test is a progressive linking and testing of programs or modules in order to ensure their proper functioning in the complete system.

Yes No I don't know

Have you performed regression tests? *
Regression testing is a retesting activity aimed at detecting defects introduced by a change.

Yes No I don't know

Have you documented the tests? *
Test documentation is the documentation describing the plans or results of testing a system or component.

Yes No I don't know

Have you done any test reports? *

FIGURE 4.13 – Questionnaire d'évaluation de la maturité technologique d'une SdMT (SMAQ)

Have you performed any tests ?

Test is an activity in which a system or component is executed under specified conditions, the results are observed or recorded, and an evaluation is made of some aspect of the system or component.

Yes

No

I don't know

TABLE 4.3 – Question type présente dans le questionnaire d'évaluation de la maturité. La définition provient de SEVOCAB

4 SYNTHÈSE DE LA DÉMARCHE

La figure 4.14 illustre la transformation des termes annotés en questions, selon quatre étapes. Elles ont permis de créer le score de maturité à partir de la norme ISO/IEC 62304. Elle montre comment de simples termes de génie logiciel pris dans un contexte qualité (les termes annotés) sont regroupés en sous-ensembles (\mathcal{D} , \mathcal{C} et \mathcal{M}) selon le mode de mise en correspondance possible avec les termes logiciels du lexique de référence. Puis, comment ces termes sont transformés en SEA qui sont alors listés pour être vérifiés au moyen d'un questionnaire.

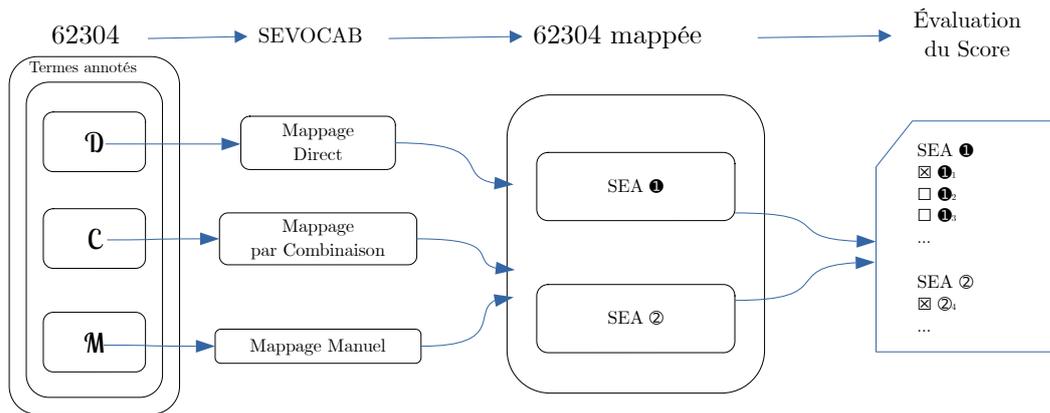


FIGURE 4.14 – De la norme ISO/IEC 62304 au score de maturité d'une SdMT. Une transformation des termes de génie logiciel présents dans la norme en SEA.

4.1 L'approche

En construisant notre méthodologie d'évaluation de la maturité, nous avons formulé des hypothèses basées sur nos observations et la littérature présentée dans le chapitre 2 et 3. La première étape a consisté en l'annotation manuelle de la norme ISO/IEC 62304, le mappage des termes annotés vers les SEA décrits dans SEVOCAB, et la classification de ces SEA.

Annotation et mappage : l'hypothèse 4 a été validée en mappant 165 termes annotés de la norme vers 156 SEA dans SEVOCAB. La plupart des SEA a été obtenue soit en utilisant l'approche par mappage direct, soit en utilisant la méthode par mappage combiné (78,8 %), montrant ainsi la forte relation entre les termes utilisés dans l'ISO/IEC 62304 et le vocabulaire proposé dans SEVOCAB. Seuls 21,2 % des SEA sont obtenus par une méthode de mappage manuelle. Cette répartition entre les ensembles de mappage \mathcal{D} , \mathcal{C} et \mathcal{M} indique une forte relation entre la norme ISO/IEC 62304 et le vocabulaire d'ingénierie logicielle. Cela vient du fait que les concepts de génie logiciel sont sous-jacents à la norme ISO/IEC 62304.

Classification : l'étape de classification a permis de distinguer les SEA de classe ① des SEA de classe ②. La proportion plus élevée de SEA ② dans toutes les clauses confirme la nécessité d'activités de spécification pendant le cycle de vie des logiciels médicaux. Cela met en lumière le fait que la norme ISO/IEC 62304 accorde une attention significative à cet aspect.

Score : le modèle de maturité évalue le niveau de maturité en comparant les **SEA** présents dans une **SdMT** avec ceux de la norme. Le score, variant de 0 à 100%, représente le niveau de maturité atteint. Les résultats montrent que quelques **SEA** bien choisis peuvent rapidement augmenter le score, tandis que la progression vers un score de 100 % nécessite la production d'un nombre significatif de **SEA**. Cette observation souligne l'importance de certains **SEA** clés dans l'évaluation de la maturité, permettant d'atteindre rapidement des niveaux de maturité significatifs.

5 CONCLUSION

Dans ce chapitre, nous avons montré qu'il est possible de construire un score basé sur des termes technologiques extraits de la norme ISO/IEC 62304 pour évaluer la maturité technologique d'une **SdMT** pendant la phase de recherche. Cette méthode permet une certaine évaluation de la maturité de manière précise et quantifiée d'une **SdMT**.

À la fin de la première étape, un total de 165 termes différents ont été annotés dans le texte de la norme ISO/IEC 62304. L'étape de mappage a abouti à 156 **SEA**.

En termes d'importance pour le score de maturité, l'activité principale est décrite dans le chapitre 5 « Développement » avec 326 citations. Les vingt **SEA** les plus cités sont présentés dans la Fig. 6.5a et représentent 50.2% du score de maturité.

Le chapitre suivant présente une expérimentation de l'utilisation du **SMAQ** par des experts du domaine sur une **SdMT**.

La suite du document abordera les travaux réalisés pour l'amélioration du questionnaire **SMAQ** dérivé de la liste de **SEA**.

5

APPLICATION À UN PROJET DE RECHERCHE DÉVELOPPANT UNE SDMT

1	Présentation du projet « FollowKnee »	96
2	Objectif de l'étude sur FollowKnee	97
3	Méthodologie d'expérimentation et collecte des données pour FollowKnee	98
4	Présentation des Résultats	98
5	Implications et ajustements de la méthodologie	105
6	Conclusion sur l'étude FollowKnee	106

Tester, c'est voir l'innovation dans le miroir de la réalité. La confrontation révèle la force des fondations et guide les ajustements nécessaires.

Inconnu

Au chapitre 4, nous avons détaillé les fondements méthodologiques permettant l'extraction et l'obtention des SEA ainsi que la formulation du score de maturité technologique pour les SdMT. À présent, il est important de mettre à l'épreuve notre approche au sein d'un cadre expérimental concret. Ce chapitre concis s'inscrit dans une démarche d'application et de validation. Il s'attelle à l'essai de notre système d'évaluation de la maturité (le SMAQ) sur un projet de recherche réel : « FollowKnee ».

« FollowKnee » représente une initiative de pointe dans le domaine des SdMT. Ce projet a pour objectif à développer une prothèse de genou pilotée par logiciel. Il incarne les défis et les exigences typiques auxquels les SdMT sont confrontées, notamment en matière de conformité à la norme ISO/IEC 62304. Avec l'expérimentation de l'utilisation de notre score de maturité sur « FollowKnee », nous cherchons à vérifier la pertinence du SMAQ pour l'évaluation de la maturité technologique au sein d'un projet de recherche.

Il s'agira d'analyser la présence et l'absence des SEA identifiés précédemment, dans la SdMT développée par l'équipe du projet « FollowKnee » puis de calculer le score de maturité technologique du projet. Au moyen de cette étude de cas, nous cherchons à déterminer l'utilité pratique de notre méthode et sa capacité à fournir une évaluation fiable et objective de la maturité technologique d'une SdMT en phase de recherche. Nous débuterons par une description du projet « FollowKnee », suivie par une présentation de la méthodologie expérimentale. Enfin, nous procéderons à l'analyse des résultats ainsi qu'à l'examen des implications et des ajustements méthodologiques.

1 PRÉSENTATION DU PROJET « FOLLOWKNEE »

Ce projet est axé sur l'amélioration de la chirurgie de remplacement du genou. Avec une augmentation prévue de 673% de ces procédures aux États-Unis et en Europe d'ici à 2030, « FollowKnee » s'attaque aux défis urgents posés par cette croissance¹.

1.1 Axes d'innovation de « FollowKnee »

« FollowKnee » se distingue par ses trois axes d'innovation, formant trois sous-projets :

1. **Prothèses et instrumentation personnalisées** : l'objectif est de pouvoir concevoir automatiquement à partir d'un scanner une prothèse de genou personnalisée qui pourra être fabriquée par impression 3D. Cet axe aligne « FollowKnee » sur les tendances de personnalisation en médecine. Cette démarche permet d'assurer des solutions sur mesure pour chaque patient.
2. **Réalité augmentée en chirurgie** : l'utilisation de la réalité augmentée pour la pose des prothèses devrait permettre d'augmenter la précision et l'efficacité des interventions.
3. **Capteurs pour le suivi postopératoire** : l'intégration de capteurs dans les prothèses pour le suivi postopératoire devrait permettre d'assurer la surveillance continue et la prévention des complications.

« FollowKnee » s'inscrit pleinement dans une démarche de recherche translationnelle. Les découvertes scientifiques sont directement reliées aux applications cliniques comme le montre la figure 5.1.

1.  Une vidéo de présentation du projet « FollowKnee »

Une thérapie personnalisée pour répondre aux changements démographiques des 20 prochaines années

Vers une approche globale guidée par les données



FIGURE 5.1 – Les objectifs du projet « FollowKnee » (PTG : Prothèse Totale de Genou) (image tirée du projet).

2 OBJECTIF DE L'ÉTUDE SUR FOLLOWKNEE

Le projet « FollowKnee », avec un fort potentiel à évoluer vers la prise en compte des contraintes réglementaires, offre un terrain d'application idéal pour notre méthode d'évaluation de la maturité. De plus, les activités d'ingénierie dans ce projet reposent pour une grande part sur des développements logiciels. Ces développements étant réalisés dans le cadre de la recherche dans les technologies médicales, les activités sont destinées à produire des [SdMT](#).

FOCUS SUR LA PARTIE 'PLANNING' DU PROJET

Notre évaluation s'est concentrée sur la partie désignée comme le « planning » dans le projet « FollowKnee »². Il s'agit d'un projet extérieur à nos travaux et l'analyse détaillée de la technologie est de nature confidentielle. Ainsi, notre étude s'est appuyée uniquement sur les réponses fournies au [SMAQ](#), sans nécessiter un accès à des informations confidentielles sur le projet lui-même.

Notre objectif est d'abord de valider notre démarche du point de vue de sa faisabilité et d'observer les résultats pour ajuster notre proposition.

2.  Une vidéo du logiciel du projet.

3 MÉTHODOLOGIE D'EXPÉRIMENTATION ET COLLECTE DES DONNÉES POUR FOLLOWKNEE

PROCÉDURE DE COLLECTE DES DONNÉES

Une méthodologie spécifique a été mise en place, elle s'est déroulée en trois temps. Dans un premier temps, nous avons sollicité les ingénieurs et responsables du développement logiciel du projet « FollowKnee » au laboratoire [LaTIM](#). Pour cela, ils ont reçu un courrier électronique contenant une description détaillée de l'étude, les instructions pour répondre au [SMAQ](#) et le type de réponses attendues. Une démarche CNIL a également été réalisée pour la déclaration du recueil de données et les répondants potentiels ont été informés de leurs droits vis-à-vis de la réglementation en vigueur.

Après avoir obtenu l'accord explicite des participants, nous avons retourné les instructions spécifiques au [SMAQ](#). Ces instructions portaient sur la manière de remplir le [SMAQ](#). Des indications ont été données sur la signification des questions et comment faire le choix entre les trois possibilités qui étaient « Yes », « No », « I don't know ». Ces 3 réponses possibles font partie du [SMAQ](#) qui a été conçu pour recueillir des informations précises sur les [SEA](#) produits au cours des activités de développement présentées au chapitre 4. Les informations recueillies permettent le calcul du score de maturité technologique. Ici, il s'agit de la partie « planning » du projet « FollowKnee ». Les ingénieurs travaillant sur cette partie ont reçu un lien vers la version électronique du [SMAQ](#) qui est présenté à la figure 4.13. Lors de l'évaluation, les ingénieurs étaient au nombre de deux. Ces deux ingénieurs sont expérimentés, chacun ayant plus de cinq ans d'expérience dans le secteur du développement logiciel en technologies pour la santé. À la fin du traitement du [SMAQ](#), ils devaient également indiquer le temps mis pour traiter la totalité du [SMAQ](#). Une fois le [SMAQ](#) complété, un fichier automatiquement généré après validation du répondant et contenant les réponses nous a été retourné via courrier électronique (aucune donnée n'est stockée sur nos serveurs pour garantir la confidentialité des informations). Ensuite, nous avons réalisé l'analyse des retours des deux ingénieurs. Cette analyse a fourni les résultats que nous décrivons dans la suite de ce chapitre.

Enfin, les deux professionnels ont participé à une interview conjointe pour leur permettre d'exprimer des commentaires sur les réponses qu'ils ont données. Cette interview nous a également permis de recueillir leurs avis sur le [SMAQ](#) lui-même.

4 PRÉSENTATION DES RÉSULTATS

Les réponses obtenues grâce au [SMAQ](#)³ sont présentées dans la table 5.1. Elle donne la liste des différents [SEA](#) évalués et des réponses fournies par les ingénieurs de l'équipe « FollowKnee ». Quand il n'y a pas de valeur, il s'agit d'une question pour laquelle le répondant correspondant n'a pas fourni de réponse. En effet, le [SMAQ](#) présente les questions de manière regroupées. De telle sorte que, si le répondant ne répond pas positivement à une réponse « chapeau » les questions qui

3.  Le Questionnaire [SMAQ](#)

en découlent ne lui sont pas posées. Par exemple, si pour le SEA « Test » la réponse est non, alors les questions concernant les SEA « Integration Test » ou « Unit test » ne sont pas posées. L'ensemble des réponses des deux ingénieurs portent sur 63 SEA différents sur un total de 156 SEA présents initialement dans la liste. L'ingénieur 1 a répondu pour 59 SEA et l'ingénieur 2 a répondu pour 57 SEA.

TABLE 5.1 – Réponses au SMAQ pour la partie « Planning » de FollowKnee

SEA	Classe	Réponse Ingénieur 1	Réponse Ingénieur 2
Acceptance Criteria	①	No	No
Anomaly Documentation	②	No	No
Change	①	No	Yes
Change Control	②		No
Change Management	②		No
Change Request	②		Yes
Change Risk Management	②		No
Change Verification	②		No
Configuration Documentation	②	No	No
Configuration Identification	②	No	No
Configuration Item	①	No	No
Configuration Management	②	No	No
Configuration Tool Documentation	②	No	No
Corrective Maintenance	②	No	No
Deliverable	①	Yes	No
Deliverable Verification	②	No	
Documentation Information	②	No	No
External Component	①	Yes	No
Fault Isolation	②	No	No
Hardware Item Integration	②	No	Yes
Integration Plan	②	No	No
Integration Test	①	Yes	Yes
Integration Test Report	②	No	No
Issue Documentation	②	Yes	Yes
Issue Evaluation	②	No	No
Issue Report	②	No	No
Library	①	Yes	Yes
Library Configuration Item	①	No	No
Library Functional Requirement	②	No	No
Library Hardware Requirement	②	No	No
Library Identifier	②	No	No
Library Performance Requirement	②	No	No
Library Risk Analysis	②	No	No
Library Software Requirement	②	No	No
Problem Report	②	No	No
Problem Safety Evaluation	②	No	No
Regression Test	①	No	No

Suite sur la page suivante

Table 5.1 – suite de la page précédente

SEA	Classe	Réponse Ingénieur 1	Réponse Ingénieur 2
Release	①	No	No
Requirement Traceability	②	No	Yes
Risk	①	No	No
Software Architecture	①	Yes	No
Software Architecture Documentation	②	Yes	
Software Architecture Verification	②	No	
Software Detailed Design	②	No	No
Software Development Life Cycle	①	Yes	No
Software Development Life Cycle Model	②	Yes	
Software Development Plan	②	No	No
Software Integration	②	Yes	Yes
Software Item	①	Yes	Yes
Software Item Integration	②	Yes	Yes
Software Item Interface	①	Yes	No
Software Maintenance	②	No	Yes
Software Requirement	②	Yes	Yes
Software Safety Classification	②	No	No
Software Tool	①	Yes	Yes
Software Unit	①	Yes	No
Software Unit Implementation	②	Yes	
Software Unit Integration	②	Yes	
Software Unit Verification	②	Yes	
Software Verification	②	No	No
System Requirement	②	Yes	Yes
Test	①	Yes	Yes
Test Documentation	②	No	No
Test Report	②	No	No

CALCUL DU SCORE

Selon les réponses obtenues, nous avons procédé au calcul du score de maturité technologique de la partie planning du projet « FollowKnee ».

Les résultats de ces calculs sont présentés en détail à l'annexe A.4. La table A.3 présente le score du projet « FollowKnee » en mettant en commun les réponses des deux ingénieurs. Les tables A.5 et A.7 de la même annexe A.4 présente le calcul du score selon les réponses (table A.4 et table A.6) de chaque ingénieur individuellement.

Pour obtenir ces scores, nous avons appliqué la méthode décrite à la section 3.5.1 page 88.

La table 5.2 présente la répartition des réponses données par les deux ingénieurs lors de l'évaluation de la maturité de leur SdMT. Pour chaque ingénieur, la table indique le nombre de SEA évalués comme présents (« Yes ») et absents (« No »). Il indique également le score de maturité technologique qui résulte de cette évaluation, exprimé en pourcentage. Ces scores donnent une mesure relative de la matu-

	Réponse	Nombre	Score
Ingénieur 1	Oui	21	31%
	Non	38	
Ingénieur 2	Oui	15	28%
	Non	42	

TABLE 5.2 – Répartition des données de réponse par ingénieur.

rité actuelle de la technologie par rapport à un référentiel représenté par la liste des 156 SEA extraits de la norme.

Lors de l'interview, les deux ingénieurs se sont accordés sur un ensemble de points que nous détaillerons dans une sous-section dédiée dans la suite de ce chapitre. Tout d'abord nous analysons les résultats issus directement des réponses fournies au SMAQ.

4.1 Analyse des Résultats de l'Expérimentation sur FollowKnee

Les réponses recueillies par l'intermédiaire du SMAQ montrent des différences entre les ingénieurs interrogés. Dans les scores calculés, on observe une légère différence de 3% entre le score le plus bas et celui le plus élevé. Pour comprendre cette différence de points de vue, nous détaillons ces résultats.

Pour approfondir, nous ajoutons le calcul des scores en version optimiste et en version pessimiste. Ils sont présentés dans la table 5.3. Il s'agit de scores communs. Pour les établir, nous considérons l'ensemble des SEA pour lesquels les deux ingénieurs ont répondu. Si pour un SEA donné l'un des deux ingénieurs n'a pas répondu, alors le SEA n'est pas pris en compte pour le calcul des scores communs. Le score optimiste est obtenu en gardant la réponse positive lors d'une réponse commune différente. Par exemple, si l'ingénieur 1 a répondu « Yes » à la présence d'un SEA et que l'ingénieur 2 a répondu « No » pour le même SEA, la réponse « Yes » est conservée. Le score pessimiste fait le contraire en ne conservant que la réponse « No » en cas de différence.

	Optimiste	Pessimiste	Ingénieur 1	Ingénieur 2
Scores	36 %	22 %	31 %	28 %

TABLE 5.3 – Calcul de scores communs optimiste et pessimiste et comparaison avec les scores individuels.

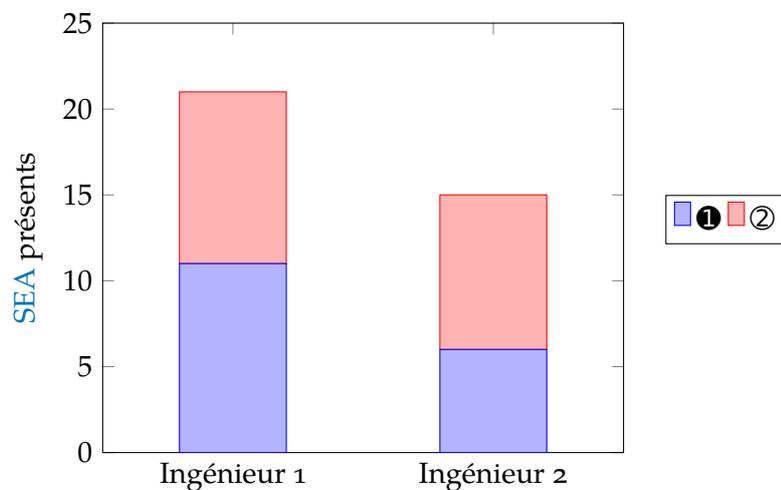
Logiquement les scores optimiste et pessimiste encadrent les scores des deux ingénieurs. La différence entre les scores optimistes et pessimistes met en évidence le niveau de désaccord ou de variabilité dans les réponses des ingénieurs. Ainsi, un grand écart entre les scores optimistes et pessimistes indique que les ingénieurs ont des avis différents sur la présence de certains SEA. Cela indique aussi que ces désaccords concernent potentiellement des SEA importants pour la maturité de la technologie compte tenu de l'influence des poids sur le score de maturité.

Cela signifie qu'il pourrait y avoir des éléments clés de la technologie, essentiels à sa maturité globale, sur lesquels les ingénieurs ne sont pas alignés.

La figure 5.2 permet d'observer la répartition des SEA présents selon leur classe et par ingénieur. La table 5.2a fournit les valeurs précises du nombre des SEA ① et ② présents selon les réponses des 2 ingénieurs. La figure 5.2b montre visuellement que cette répartition des SEA ① et des SEA ② est presque équilibrée dans les réponses de l'ingénieur 1 et légèrement en faveur des SEA ② dans les réponses de l'ingénieur 2.

	①	②
Ingénieur 1	11	10
Ingénieur 2	6	9

(a) Nombre de SEA présents selon les ingénieurs et les classes.



(b) Répartition des SEA selon les ingénieurs et les classes.

FIGURE 5.2 – Répartition des SEA dans les réponses des ingénieurs selon leur classe.

Les SEA ainsi comptabilisés peuvent également être représentés en fonction des clauses de la norme ISO/IEC 62304 (voir chapitre 4 page 69). La relation du SEA avec la clause de la norme au travers du terme annoté correspondant ayant été conservée, il est possible de dresser un tableau de cette répartition. La table 5.4 montre cette répartition en pourcentage des SEA présents pour chaque ingénieur.

	Clause 5	Clause 6	Clause 7	Clause 8	Clause 9
Ingénieur 1	90,4%	4,2%	3%	1,2%	1,2%
Ingénieur 2	74,3%	12,2%	4%	5,4%	4%

TABLE 5.4 – Répartition des SEA en fonction des clauses de la norme selon les réponses par ingénieur.

La table 5.4 montre bien le déséquilibre dans la répartition des SEA selon les clauses de la norme ISO/IEC 62304. Nous observons une proportion très importante des SEA de la clause 5 dans les réponses des 2 ingénieurs.

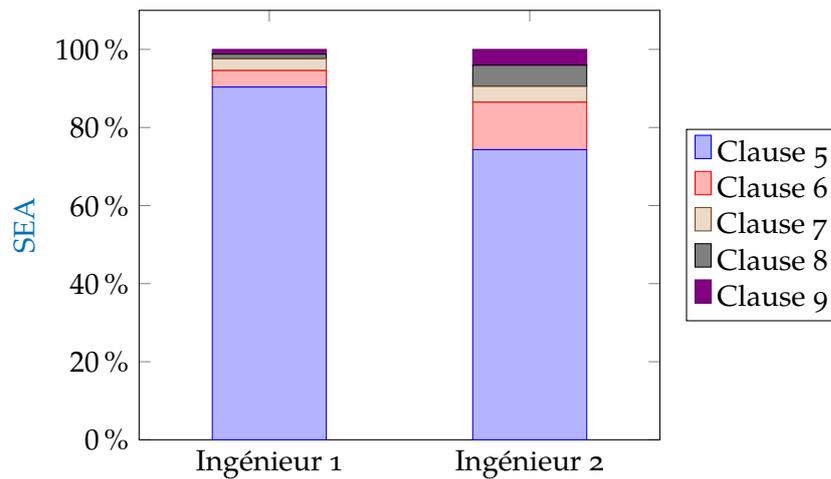


FIGURE 5.3 – Répartition des clauses par ingénieur.

Plus précisément, à partir de ces résultats, nous pouvons voir que les deux ingénieurs ont couvert un grand nombre de citations de la clause 5. Cela indique que les deux ingénieurs perçoivent bien la couverture de cette partie de la norme. L'ingénieur 1 a couvert plus de citations dans la clause 6 que l'ingénieur 2. Cela permet de caractériser la différence de point de vue. Cette différence peut-être due à une perception spécifique de la [SdMT](#) de la part de chacun des deux ingénieurs. Cela peut être fonction des tâches spécifiques qui leur sont confiées ou plusieurs de ces facteurs en même temps. Nous avons abordé ce point qui touche au pragmatisme lors de la description des [SEA](#) (chapitre 3 à la section 1 page 54).

Les clauses 7, 8 et 9 sont relativement peu couvertes par les deux ingénieurs, mais l'ingénieur 2 fournit des réponses permettant une meilleure couverture et notamment pour ce qui concerne la clause 6.

Pour la technologie elle-même, ces résultats pourraient suggérer des domaines où il pourrait être nécessaire de revoir la documentation ou la technologie pour assurer une meilleure conformité avec la norme. Ils pourraient également indiquer des domaines où la formation ou la clarification pourraient être nécessaires pour les ingénieurs.

4.2 L'interview

Lors de cette interview, les questions posées étaient basées sur un échange ouvert et ont suivi le déroulement du [SMAQ](#) lui-même en insistant sur les points suivants :

- la méthode d'interrogation,
- la lisibilité de l'Interface,
- la pertinence des [SEA](#),
- la pertinence du [SMAQ](#) pour l'évaluation de la maturité.

Nous résumons, ci-dessous les réponses et commentaires des deux ingénieurs.

Un premier constat que l'on peut faire après discussions concernant les remarques formulées par les ingénieurs est que leurs interactions avec la norme

ISO/IEC 62304 ont été différentes durant leur vie professionnelle antérieure. Tandis que l'ingénieur 2 a été impliqué dans l'application détaillée de la norme, l'ingénieur 1 a bénéficié d'une expérience plus proche du développement et moins détaillée dans sa mise en œuvre. Malgré cette différence, les répondants ont unanimement perçu une forte dimension réglementaire dans le SMAQ. Ils ont attribué cette perception à la manière dont les définitions étaient formulées et intégrées aux questions. De plus, ils ont indiqué que les définitions n'étaient pas toujours très claires. Cela a sans doute influencé la différence observée dans les réponses des deux ingénieurs.

Ils ont également émis des remarques sur la présentation méthodique des SEA liés à la norme, mise en évidence par une liste de questions. Selon eux, cette présentation leur a permis de prendre conscience de l'ampleur du travail restant. Ils ont concordé sur la pertinence de cette liste, tout en soulignant que l'importance relative de chaque question varie selon le stade d'avancement du projet de recherche. Plus particulièrement dans le contexte des GMCAO. Cette variation souligne la dynamique des exigences réglementaires, qui diffèrent significativement entre les phases pré et post-essais cliniques.

L'ingénieur 1 a spécifiquement mentionné que certains SEA seraient inévitablement révisés dans la suite du processus. En effet, les politiques d'intégration de la SdMT par les entreprises impliquent des méthodologies spécifiques à ces entreprises. Toutefois, des artefacts tels que le « Software Management Plan » ont été identifiés comme essentiels lors de l'évaluation, bien qu'ils soient sujets à des révisions futures. Cela indique ainsi l'importance de l'adaptabilité et de la conformité continue aux standards réglementaires dans le développement de logiciels médicaux y compris en phase recherche.

Enfin, les deux répondants confirment la pertinence du SMAQ comme outils d'évaluation de la maturité technologique d'une SdMT. De même, il confirme l'appartenance des éléments constitutifs du SMAQ au domaine du génie logiciel.

4.3 Discussion sur l'expérimentation FollowKnee

L'analyse des réponses au SMAQ indique des variations dans la présence des SEA. Par exemple, les réponses divergentes entre les ingénieurs sur des SEA tels que « Change Control » et « Integration Test » suggèrent des différences dans la perception ou la mise en œuvre de ces éléments au sein du projet.

Les différences que l'on peut observer peuvent être expliquées selon le modèle de maturité que nous proposons. Ce dernier repose sur la nature même des SEA. Comme nous l'avons montré dans le chapitre 2 et selon MÉNDEZ FERNÁNDEZ et al. (2019) les SEA ont un contenu sémantique qui peut être interprété de manière différente selon l'individu qui le considère en fonction de ses propres expériences et connaissances. Cela se traduit dans le SMAQ par des réponses différentes aux mêmes questions pour un même SEA d'une même SdMT, cet effet, étant renforcé par le manque de clarté dans les définitions. Cela montre l'importance de considérer cet aspect sémantique qui dépend aussi fortement du contexte et par conséquent de l'étape dans laquelle se situent les développements du logiciel, comme le font remarquer les deux ingénieurs.

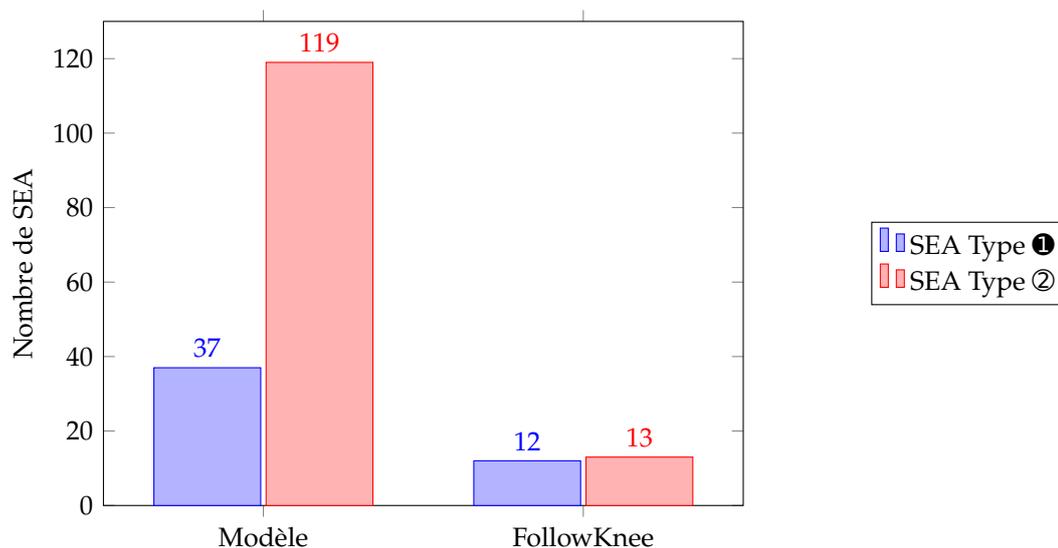


FIGURE 5.4 – Répartition des types de SEA développés pendant les phases de Modèle et FollowKnee.

De plus, on peut remarquer dans les représentations graphiques et notamment celles traitant de la répartition des SEA par rapport à la norme (figure 5.3), le déséquilibre entre les SEA de la clause 5, plus orientés vers le développement, par rapport aux autres clauses qui sont plus orientées sur les aspects de spécifications, d'analyse du risque et de maintenance. Ce déséquilibre semble logique dans une phase précoce du projet de recherche, là où on se concentre sur la mise au point de la technologie.

La figure 5.4, montre la comparaison entre la répartition des types de SEA annoncés comme présent dans la SdMT en cours de développement au sein du projet « FollowKnee » et la répartition des types de SEA dans l'ensemble des SEA de notre modèle. On observe que la proportion de SEA de type ① par rapport à celle des SEA de type ② est différente de la répartition de cette proportion pour les SEA extraits au moment de l'évaluation.

Les données obtenues fournissent un aperçu de l'état actuel de la maturité technologique des SdMT dans le projet « FollowKnee ». Elles soulignent l'importance de l'harmonisation des pratiques et des perceptions des différents acteurs impliqués dans le développement des SdMT.

5 IMPLICATIONS ET AJUSTEMENTS DE LA MÉTHODOLOGIE

La récolte des données et leur analyse confirment la faisabilité de notre méthode d'évaluation basée sur le SMAQ. Cependant, les divergences dans les réponses soulèvent des questions sur l'interprétation uniforme des SEA de génie logiciel par différents développeurs. Cela implique de mesurer cette maturité technologique auprès de professionnels qui ont une connaissance globale de la SdMT évaluée. Ainsi, les résultats de l'expérimentation sur « FollowKnee » ont des implications pour notre méthode d'évaluation de la maturité technologique des SdMT. En effet,

compte tenu des réponses fournies par les deux ingénieurs lors de l'interview il semble nécessaire de procéder aux ajustements suivants :

- Proposer de nouvelles définitions plus accessibles pour chaque [SEA](#) et ayant un aspect moins théorique.
- Réaliser une validation de la liste des 156 [SEA](#) pour valider la pertinence de chaque [SEA](#). Cela permettra de ne garder que les [SEA](#) faisant consensus pour caractériser la maturité technologique d'une [SdMT](#).
- Faire valider le [SMAQ](#) auprès d'une audience d'experts plus large.

Implications pour le projet « FollowKnee »

L'étude met en évidence des domaines spécifiques au sein du projet « FollowKnee » qui pourrait bénéficier d'une attention accrue. Notamment en ce qui concerne la documentation et la gestion des changements.

Les réponses « Yes » et « No » des ingénieurs reflètent le niveau d'accord entre eux et envers un référentiel de maturité basé sur la norme ISO/IEC 62304 pour chaque [SEA](#) spécifié. Tirer des conclusions sur une démarche à entreprendre au travers de la considération des scores de manière individuelle donne une idée de l'état dans lequel se trouve la technologie. Pour être plus précis, on peut considérer qu'un score moyen entre les deux scores optimiste et pessimiste fonderait une idée plus fiable de la maturité et montrerait l'état actuel du travail réalisé et restant à réaliser.

Le niveau de désaccord quant à lui, peut indiquer, par exemple, qu'une concertation de l'équipe technique est à prévoir pour harmoniser les avis sur les points clés à développer. Les différences observées entre les réponses des deux ingénieurs peuvent indiquer des domaines nécessitant une attention particulière pour améliorer la standardisation et la documentation au sein du projet dans le contexte d'une démarche en direction d'une mise en conformité réglementaire.

On peut supposer que plus les scores sont différents et plus l'harmonisation sera nécessaire pour obtenir une progression coordonnée et efficace. Dans le cas de « FollowKnee », les deux ingénieurs ont une perception de la norme relativement proche, il serait intéressant d'en vérifier la réalité.

6 CONCLUSION SUR L'ÉTUDE FOLLOWKNEE

Cette étude sur le projet « FollowKnee », bien que limitée à deux ingénieurs, a fourni des entrées précieuses sur l'application de notre méthode d'évaluation de la maturité technologique des [SdMT](#).

Cette expérimentation sur un cas réel nous a indiqué sur quelles pistes porter notre attention pour de futures améliorations, notamment en termes d'optimisation des méthodes d'évaluation et d'élargissement de leur application à d'autres projets dans le domaine des [SdMT](#).

Cela nous a permis également d'initier une phase de validation du [SMAQ](#), qu'il est nécessaire maintenant d'élargir auprès d'un plus grand nombre d'experts. Le

chapitre suivant présente de manière détaillée une deuxième étape de ces travaux de validation.

6

VALIDATION DU MODÈLE DE MATURITÉ

1	Validation statistique	109
2	Application des techniques de validation au SMAQ	120
3	Validation de nos résultats	123
4	Synthèse de la validation du SMAQ	129
5	Conclusion	131

Si vous ne pouvez pas le
mesurer, vous ne pouvez pas
l'améliorer.

Lord Kelvin

Dans le chapitre 4, nous avons décrit l'élaboration d'un questionnaire et d'un score de maturité basé sur les concepts d'ingénierie logicielle extraits de la norme ISO/IEC 62304 (ISO/IEC 2006). Cette extraction répertorie les SEA qui sont ensuite utilisés pour construire le questionnaire que l'on nomme le SMAQ. Ce dernier a pour objectif de permettre l'évaluation de la maturité technologique d'une SdMT.

La validation statistique du SMAQ est une étape importante. Elle a pour objectif de soumettre le SMAQ à un test de fiabilité et de validité pour mesurer ce qu'il prétend mesurer. Pour réaliser cette validation, nous avons procédé en vérifiant la validité apparente et la validité de contenu. Ce chapitre présente ce que sont ces deux types de validité et comment ils ont été vérifiés.

Mais avant d'entrer plus en détail dans notre démarche, examinons en quoi consiste la validation de questionnaires.

1 VALIDATION STATISTIQUE

Les questionnaires sont très utilisés dans la recherche médicale, la pratique clinique, la psychologie et les études comportementales, entre autres domaines. Ils servent à recueillir des données via un ensemble structuré de questions. La conception du questionnaire lui-même est très importante, il est désigné par le terme d'« instrument ». C'est-à-dire, un objet qui concrétise le construit. Le terme de « construit » représente, quant à lui, le concept faisant l'objet de la mesure.

Dans notre cas (voir la table 6.1)

- le construit est le concept représenté par tous les items du SMAQ, c'est-à-dire la maturité technologique d'une SdMT,

- les items sont représentés par les énoncés ou les questions individuelles du SMAQ
- l'instrument est le SMAQ lui-même.

Comme l'indique TAHERDOOST (2016), l'instrument permet l'opérationnalisation du construit en le transformant en une variable mesurable. Ce qui est critique est alors de vérifier que l'instrument capte véritablement l'essence du construit ciblé, c'est en cela que consiste l'étape de validation. Cette étape de validation est souvent menée auprès d'un panel d'experts de tous niveaux. Leur expertise se situe dans le domaine du construit à valider. La table 6.1 représente les termes et les définitions pour la réalisation de la validation de nos outils de mesure de la maturité d'une SdMT. Dans la suite de ce document, nous utilisons le terme générique « instrument » ou le terme spécifique « questionnaire » qui est le type d'instrument que nous étudions.

Outils d'évaluation de la maturité des SdMT	
Construit	L'évaluation de la maturité technologique des SdMT
Instrument	Il s'agit du SMAQ
Items	Chaque item est représenté par chacune des 156 Questions du SMAQ
Mesure	Le score de maturité technologique d'une SdMT
Juges	Ce sont les experts que l'on va interroger lors de la validation

TABLE 6.1 – Termes utilisés pour désigner les éléments de nos outils d'évaluation de la maturité technologique des SdMT.

Selon la littérature, les techniques de validation comprennent plusieurs étapes et différentes approches pour s'assurer de la fiabilité et de la validité de ces instruments. Dans leur article Cook et al. (2006) plaident pour une idée nouvelle de la validité d'un instrument. Elle repose sur le fait que cette notion de validité ne doit pas simplement être considérée comme une propriété intrinsèque de l'instrument. En effet, selon eux, le concept de validité d'un instrument est plus nuancé et dynamique. La validité dépend de la manière dont les scores obtenus grâce à cet instrument sont interprétés.

En effet, la validité des scores générés par un instrument peut être conditionnée par le contexte spécifique de son utilisation ou par l'interprétation que l'on en fait. Un instrument conçu pour évaluer la dépression peut fournir des scores valides lorsqu'il est utilisé dans des études cliniques sur cette même condition. En revanche, ces scores pourraient ne pas être appropriés si l'instrument était employé pour mesurer un autre construit psychologique proche, tel que l'anxiété. Ici, on voit bien comment la validité, telle que redéfinie par Cook et al. (2006), est intuitive et dynamique. Il y a donc une exigence pour une analyse approfondie afin de garantir que l'instrument reste valide dans le contexte d'application.

Dans cette optique, les auteurs reconnaissent que la validité est liée au but pour lequel les scores sont utilisés. Ils soulignent, également, l'importance de recueillir des preuves. Ils indiquent cinq sources de preuves pour soutenir l'interprétation spécifique des scores. Ces sources sont le contenu, le processus de réponse, la structure interne, les relations avec d'autres variables et les conséquences de l'utilisation des scores.

Dans son article YUSOFF (2019) étend ces concepts en se concentrant spécifiquement sur la validité de contenu auquel il donne une place primordiale au sein du processus de validation. Le contenu est une des cinq sources de preuves de validité mentionnées par COOK et al. (2006). YUSOFF (2019) propose une méthodologie détaillée en six étapes pour la validation de contenu :

- la préparation d'un formulaire de validation de contenu,
- la sélection d'un panel d'experts,
- la conduite de la validation de contenu (soit en face à face, soit de manière non présenteielle),
- la révision des domaines et des items,
- la notation de chaque item par les experts,
- le calcul du Content Validity Index (CVI).

Nous avons suivi ces six étapes pour réaliser notre propre validation de contenu. Elles sont détaillées dans la suite de ce chapitre, ainsi qu'une explication de ce que représente le CVI et comment nous l'avons utilisé dans notre cas.

Avec l'article de TAHERDOOST (2016), nous étendons cette discussion. Dans son article, il couvre de manière exhaustive les divers types de validité et de fiabilité.

En effet, il propose des méthodes spécifiques pour les tester. Cela permet de garantir que les instruments de recherche sont bien conçus et capables de collecter des données fiables et valides. Nous en avons retenu deux qui sont les principales et les plus utilisées. Dans la suite nous proposons une brève présentation des techniques de conception et de validation retenues et qui nous ont permis de mener à bien nos travaux de validation.

1.1 Méthodologie systématique pour la conception et validation du SMAQ

Dans un article plus récent, YUSOFF et al. (2021) approfondissent les discussions précédentes sur le développement et la validation des instruments et plus spécifiquement, des questionnaires, pour la recherche par enquête. Pour cela ils fournissent une approche systématique étape par étape pour la conception et la validation d'instruments du type questionnaires. Même si cet article ne couvre pas explicitement les cinq sources de preuves présentées par COOK et al. (2006), il n'en est pas moins un guide pratique détaillé de l'élaboration et de la validation de questionnaires. Dans cette section, nous voyons comment le développement du SMAQ correspond aux préconisations de ce guide et les directions qu'il indique.

En particulier, YUSOFF et al. (2021) donnent des indications précises sur le développement systématique du questionnaire. Cela comprend la définition claire des objectifs du questionnaire, la création d'items et leur arrangement logique. Ces éléments permettent d'assurer la compréhension et une réponse facile de la part des répondants. Mais aussi, de déterminer ce que le questionnaire mesurera et comment il contribuera au domaine d'étude. Dans le cas de notre étude, le SMAQ est prévu pour évaluer la présence ou l'absence de SEA dans la SdMT étudiée. Il permet ainsi de calculer le score de maturité en appliquant la formule 4.2 du chapitre 4 (voir page 88). L'objectif du SMAQ est donc bien clairement défini.

De plus, le questionnaire doit établir les qualités et caractéristiques des attributs qu'il doit mesurer. Pour le SMAQ cette étape a été satisfaite lors de l'extraction des

termes, de leur consolidation en [SEA](#) et de leur classification. L'adjonction de définitions a permis de rendre ces attributs plus compréhensibles en les remplaçant dans le contexte du génie logiciel. Nous avons ainsi diminué le risque d'ambiguïté concernant la nature des éléments dont nous cherchons à vérifier la présence ou l'absence dans la [SdMT](#).

YUSOFF et al. (2021) indiquent également l'importance de planifier le contenu du questionnaire, la population cible, le nombre et les types d'items pertinents, les instructions d'administration, le temps estimé pour le compléter et la manière dont les scores sont calculés et interprétés. Comme nous venons de le voir, dans notre cas, le score a fait l'objet d'un travail spécifique. Note questionnaire lui-même satisfait à ces indications et leur mise en œuvre est décrite dans le chapitre 5 à la section 3 (page 98). Cette méthodologie est établie dans le respect des préconisations faites par YUSOFF et al. (2021).

Néanmoins, le travail sur les items et leur présentation, bien que nécessaire, n'est pas suffisant. En effet, YUSOFF et al. (2021) préconisent de déterminer, au moyen de prétests sur le terrain ou de validation de contenu et d'entrevues, les items de manière plus précise. L'expérimentation de notre méthode sur le projet « FollowKnee » (chapitre 5) représente un prétest (ou étude pilote), à la suite de laquelle nous avons mené une interview (voir section 4.2, page 103). Cette étape de prétest a permis d'avoir une première approche des actions à mener sur les items en particulier. La validation que nous décrivons dans ce chapitre nous permettra de compléter ce que l'expérimentation sur le projet « FollowKnee » a permis d'aborder. Il sera notamment possible de définir plus précisément les items pertinents pour permettre l'évaluation de la maturité technologique d'une [SdMT](#).

Dans la suite du document, nous présentons les techniques de validation de questionnaires tels que décrits dans les travaux de TAHERDOOST (2016) et de YUSOFF (2019) notamment. Il s'agit des tests de validité qui ont été intégrés à notre démarche de validation.

VÉRIFICATION DE LA CONFORMITÉ AUX PRÉCONISATIONS DE YUSOFF

Afin d'assurer l'adéquation de notre méthode d'évaluation avec les standards de validation établis, nous avons procédé à une vérification a posteriori de notre approche en fonction des critères détaillés par YUSOFF et al. (2021). Cette revue a permis de vérifier la présence des éléments que nous venons de lister et qui sont préconisés par YUSOFF et al. (2021) dans le [SMAQ](#). Mais aussi, de mettre en place chaque étape de notre processus de validation pour se conformer aux six étapes de validation de contenu recommandées par Yusoff.

Dans la suite du document, nous présentons les techniques de validation de questionnaire que nous avons abordées plus haut par rapport aux travaux de TAHERDOOST (2016) et de YUSOFF (2019) notamment. Il s'agit des tests de validité qui ont été intégrés à notre démarche de validation.

1.2 Les tests de validité

TAHERDOOST (2016) montre l'importance de la validité et de la fiabilité dans les instruments de recherche, tels que les questionnaires et les enquêtes. Son article

présente techniquement divers aspects de cette validation et la manière de la mettre en œuvre pour un instrument donné.

La validation d'un instrument de mesure peut impliquer plusieurs niveaux de vérification, chacun ayant son importance et sa spécificité :

- **Validité apparente** : c'est la plus élémentaire forme de validation. Elle examine si, à première vue, l'instrument semble mesurer ce qu'il est censé mesurer. Ce type de validation est souvent basé sur les opinions subjectives des utilisateurs ou des experts qui évaluent la pertinence et l'apparence des items.
- **Validité de contenu** : suit la validité apparente en importance. Cette validité assure que tous les aspects du construit à mesurer sont couverts par les items de l'instrument. Elle requiert souvent un consensus d'experts qui évaluent chaque question pour sa pertinence et sa représentativité du construit.
- **Validité de construit** : ce niveau supérieur de validation teste si l'instrument mesure effectivement le construit théorique qu'il prétend mesurer. Elle inclut souvent des analyses statistiques pour examiner les relations entre les items et le construit global.
- **Validité de critère** : c'est la plus exigeante, elle évalue la performance de l'instrument en le comparant à un autre critère connu ou un « gold standard ». Elle peut être divisée en validité concurrente et prédictive.

La figure 6.1 tirée de TAHERDOOST (2016), illustre les différentes catégories de validité et leurs relations.

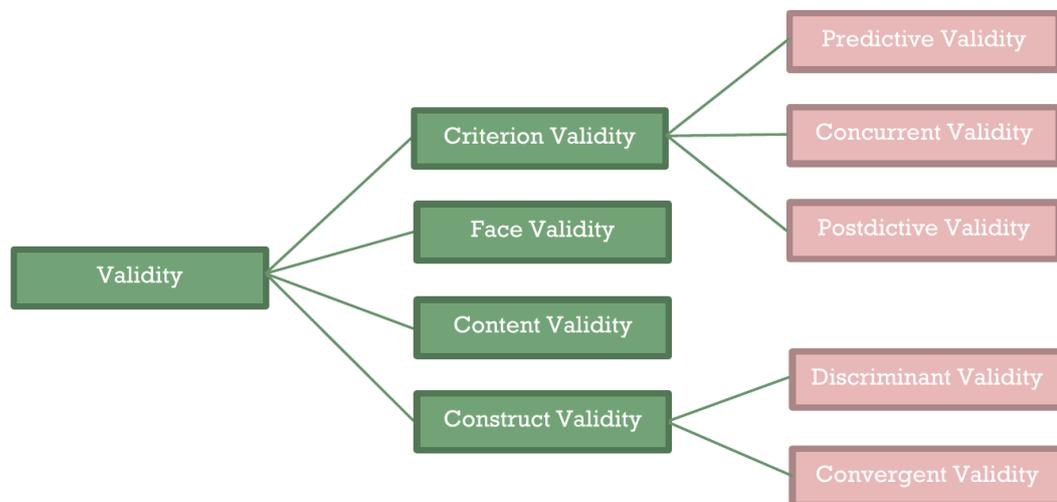


FIGURE 6.1 – Types de validité et leurs relations selon TAHERDOOST (2016).

Ainsi, plusieurs tests peuvent être menés à bien pour réaliser des travaux de validation. Nous pouvons illustrer cela au travers d'autres études (SESSA et al. 2018; FIARD et al. 2020) présentant la validation d'instrument de nature différente d'un questionnaire comme le SMAQ.

Dans l'étude de SESSA et al. (2018), le construit fait référence à la compétence ou à l'aptitude que le simulateur Xperience™ Team Trainer (XTT) est supposé développer chez l'assistant de chirurgie robotique. Le simulateur constitue l'instrument

lui-même et des formulaires sont développés pour vérifier la validation apparente et de contenu du simulateur.

Le formulaire permettant l'évaluation de la validité apparente se compose d'une série de questions conçues pour évaluer la perception des participants quant à divers aspects du simulateur. Il peut s'agir de la facilité d'utilisation de la plateforme XTT, du réalisme des exercices simulés par la plateforme, de la pertinence du « XTT » comme outil de formation pour les compétences ciblées. Pour la validité de contenu, d'autres questions permettent d'évaluer si le « XTT » enseigne effectivement ce qu'il est censé enseigner. Ces questions sont spécifiques à chaque item qui doit faire l'objet d'un « consensus » sur son niveau de pertinence au regard de ce que le simulateur est censé proposer.

Le « XTT » n'est pas un questionnaire et ne se manipule pas de la même manière. Néanmoins, il s'agit bien d'un instrument qui représente le construit visé dans cette étude et qui peut faire l'objet, au même titre qu'un questionnaire, d'une étape de validation.

Dans leurs travaux, FIARD et al. (2020), a également examiné la validation d'un simulateur, le « Biopsym ». Il est utilisé pour l'entraînement aux biopsies prostatiques. Cette étude a mis l'accent sur la validation de construit, en démontrant la capacité du simulateur à différencier les compétences des utilisateurs novices des utilisateurs confirmés. Le simulateur « Biopsym » a été évalué à la fois pour la validité de contenu et la validité apparente. Les utilisateurs ont évalué la validité de contenu en vérifiant si le simulateur enseignait effectivement les compétences nécessaires pour effectuer des biopsies prostatiques. La validité apparente, quant à elle, a été testée par des évaluations de l'interface utilisateur et du réalisme des procédures simulées. Elle a permis de vérifier que les instruments de formation doivent enseigner correctement, mais aussi refléter fidèlement la réalité de la procédure qu'ils sont censés simuler.

Ces deux exemples de validation montrent que la validité apparente et la validité de contenu sont les deux premières étapes incontournables pour la validation d'un instrument.

Dans la suite, nous présentons ce que sont validité apparente et la validité de contenu, en s'appuyant sur les données collectées à partir de ces études pour renforcer notre compréhension et application de ces concepts de validation dans le développement du SMAQ.

1.3 Validité apparente

La validité apparente est une évaluation superficielle de l'instrument. En effet, la validité apparente, ou « face validity » en anglais, est un jugement subjectif portant sur l'opérationnalisation d'un construit. Selon TAHERDOOST (2016) : « La validité apparente est la mesure dans laquelle les items de l'instrument de mesure ressemblent, sur le plan linguistique et analytique, à ce qui est censé être mesuré ». OLUWATAYO (2012) ajoute qu'elle évalue l'apparence de l'instrument en termes de faisabilité, de lisibilité, de cohérence du style et de la mise en forme, ainsi que de la clarté du langage utilisé. Ainsi, un test a une validité apparente si son contenu semble simplement pertinent pour la personne passant le test par rapport à l'objectif de mesure visé.

Pour examiner la validité apparente, une échelle dichotomique peut être utilisée avec les options catégoriques « Oui » et « Non » en guise de réponses pour chacune des questions. Elles indiquent respectivement un élément favorable et défavorable. Il peut s'agir d'une échelle de Likert comme cela a été utilisé dans l'étude de Sessa et al. (2018). Ensuite, les données collectées sont analysées pour déterminer la validité apparente de l'instrument. Landis et al. (1977) recommandent un indice Kappa (κ) acceptable de 0,60 pour l'accord interjuges, comme le montre la table 6.2.

Valeur de Kappa	Interprétation de l'accord
$\kappa < 0$	Accord inférieur à celui attendu par hasard
$0.01 \leq \kappa \leq 0.20$	Accord léger
$0.21 \leq \kappa \leq 0.40$	Accord faible
$0.41 \leq \kappa \leq 0.60$	Accord modéré
$0.61 \leq \kappa \leq 0.80$	Accord substantiel
$0.81 \leq \kappa \leq 1.00$	Accord presque parfait

TABLE 6.2 – Seuils de Kappa pour l'interprétation de la force de l'accord selon Landis et Koch (1977).

Malheureusement, la validité apparente est probablement la forme la plus faible de validité, et beaucoup suggèrent qu'il ne s'agit pas d'une forme de validité au sens strict du terme.

Alors que Cook et al. (2006) voulaient l'écartier, Taherdoost (2016) souligne que bien que la validité apparente soit considérée comme la forme de validité la plus élémentaire elle représente quand même une étape préliminaire importante dans le processus de validation d'un instrument. Elle permet, bien souvent, d'identifier rapidement les premiers problèmes liés à la pertinence et à la clarté des items et de les rectifier rapidement. Ainsi, nous avons nous-mêmes procédé à la vérification de la validité apparente avant de procéder à des activités de validation plus poussées.

1.4 Validité de contenu

Selon Taherdoost (2016), la validité de contenu est définie comme « la mesure dans laquelle les items de l'instrument de mesure sont pertinents et représentatifs du concept cible ». En d'autres termes, la validité de contenu concerne la mesure dans laquelle les items d'un questionnaire, d'un test ou d'une autre forme d'instrument de mesure (comme nous l'avons vu avec Sessa et al. (2018) ou Fiard et al. (2020)) sont représentatifs et pertinents pour le construit spécifique qu'ils sont censés représenter. Cette validation implique l'évaluation de chaque item. La table 6.3 donne un aperçu des termes utilisés pour l'évaluation de la validité dans notre cas.

Outils d'évaluation de la validité du SMAQ	
Formulaire pour la validité apparente	Un questionnaire spécifiquement développé pour évaluer la validité apparente du SMAQ.
Formulaire pour la validité de contenu	Un questionnaire spécifiquement développé pour évaluer la validité de contenu du SMAQ. Il y a 156 items (une question pour chaque question du SMAQ).
Items	Les questions des deux formulaires spécifiquement développés pour évaluer la validité apparente ou la validité de contenu du SMAQ.
Instrument	Objet dont la validité apparente ou de contenu est évaluée (SMAQ).

TABLE 6.3 – Termes utilisés pour désigner les éléments de nos outils d'évaluation de la validité du SMAQ

Dans notre cas, cette évaluation systématique de chaque item du SMAQ passe par l'élaboration d'un formulaire spécifique comme indiqué dans la table 6.3. Il permet de recueillir, la réponse de l'expert concernant la pertinence de chaque item du SMAQ. Ensuite, les données de tous les experts consultés sont collectées et analysées pour calculer un index qui permet de représenter quantitativement la validité de contenu.

LAWSHE (1975) a initialement introduit le concept de CVI pour définir cet index de validité de contenu. Pour cela, il utilise la notion de Content Validity Ratio (CVR) pour chaque item de l'instrument évalué.

Dans ses travaux, LAWSHE (1975) utilise le CVR en tant que mesure statistique pour évaluer la validité de contenu des items d'un test psychométrique ou d'évaluation professionnelle (l'instrument). Le CVR doit alors permettre de quantifier le degré de consensus parmi un panel d'experts sur l'importance d'un item spécifique du test pour un poste donné. Dans son article, le CVR est calculé à partir des jugements de ce panel d'experts qui évaluent si les compétences ou connaissances testées par chaque item sont :

- Essentielles,
- Utiles, mais pas essentielles,
- Non nécessaires.

Chaque membre du panel répond indépendamment, et le CVR pour un item spécifique est calculé comme suit :

$$\text{CVR} = \frac{(N_e - \frac{N}{2})}{\frac{N}{2}} \quad (6.1)$$

où N_e est le nombre de membres du panel qui considèrent l'item comme essentiel, et N est le nombre total de membres du panel.

INTERPRÉTATION DU CVR

Une fois la valeur du CVR calculée, elle peut être interprétée selon deux niveaux.

Le premier niveau d'interprétation, le niveau « basique », reflète la direction de l'accord entre les juges et s'établit selon trois valeurs :

- Un **CVR** positif indique qu'une majorité du panel trouve que l'item est essentiel.
- Un **CVR** de zéro signifie que 50% des membres du panel trouvent l'item essentiel.
- Un **CVR** négatif signifie que moins de la moitié du panel considère l'item comme essentiel.

Le deuxième niveau d'interprétation repose sur un ensemble de seuils statistiques qui fournissent un jugement plus robuste sur l'item. Ces seuils, basés sur la distribution binomiale, tiennent compte du nombre de membres du panel et d'un niveau de signification prédéfini (typiquement 0.05) qui correspond à un niveau de confiance de 95%. Ils aident à déterminer si le consensus parmi les membres du panel sur l'importance des items du test est statistiquement significatif et non le résultat du hasard, comme le montre le tableau présenté par LAWSHE (1975), référencé dans la table 6.4.

Nombre de Panélistes	Seuil de CVR
5	0.99
6	0.99
7	0.99
8	0.75
9	0.78
10	0.62
11	0.59
12	0.56
13	0.54
14	0.51
15	0.49
20	0.42
25	0.37
30	0.33
35	0.31
40	0.29

TABLE 6.4 – Seuils de CVR en fonction du nombre de panelistes pour un niveau de signification de 0.05.

Ainsi, si pour un item donné, la valeur calculée du **CVR** est en dessous de la valeur minimale donnée par le tableau 6.4 pour le nombre d'experts du panel correspondant, alors l'item évalué est rejeté. À la suite de ce tri, le **CVI** peut être calculé.

CALCUL DU CVI

Le CVI est la moyenne des CVR des items retenus dans l'instrument évalué. Elle est calculée comme suit :

$$CVI = \frac{\sum \text{CVR des items retenus}}{\text{Nombre d'items retenus}} \quad (6.2)$$

INTERPRÉTATION GÉNÉRALE DU CVI

Le CVI varie entre 0 et 1, où :

- un CVI proche de 1 indique un fort consensus parmi les experts que la majorité des items du test sont essentiels, ce qui indique une bonne validité de contenu,
- un CVI faible suggère un consensus moins important. Ce qui indique que moins d'items sont considérés comme essentiels par les experts et ce résultat peut remettre en question la validité de contenu de l'instrument.

Dans le cadre de ses travaux, LAWSHE (1975) a proposé une règle de base qui peut être utilisée pour évaluer le CVI. Si la valeur du CVI est de 0.78 ou plus, la validité de contenu peut être considérée comme excellente. Cette valeur de 0.78 n'est pas dérivée de calculs statistiques complexes, mais est plutôt une norme empirique. Elle provient d'une règle heuristique, l'heuristique de Lynn (LYNN 1986), développée pour aider les chercheurs à déterminer si un instrument possède une validité de contenu adéquate. Elle est censée refléter un niveau où la majorité substantielle des experts s'accorde sur l'importance des items d'un instrument. Ce qui suggère que l'instrument soit valide par rapport au construit qu'il est censé représenter. Cet index de validité de contenu a évolué depuis son apparition en 1975.

YUSOFF (2019) présente une alternative pour estimer la validité de contenu qui consiste à calculer les moyennes de l'indice de validité de contenu (Item-level-Content Validity Index (ICVI) et Scale-level-Content Validity Index (SCVI)). Contrairement à LAWSHE (1975) le formulaire d'évaluation de YUSOFF (2019) contient 4 réponses possibles par item. Pour répondre aux questions, les experts choisissent l'une des 4 valeurs possibles : 1) « Non pertinent », 2) « Plutôt pertinent », 3) « Assez pertinent », 4) « Très pertinent ». Pour calculer l'ICVI, l'équation 6.3 est utilisée où x est le nombre de réponses "Plutôt pertinent" et "Très pertinent" et N est le nombre d'experts.

$$ICVI = \frac{x}{N} \quad (6.3)$$

Le ICVI est en fait une mesure de la proportion d'accord entre les experts sur la pertinence de chaque item. Le SCVI est calculé comme la moyenne des ICVI de tous les items, que l'on note $SCVI_{Ave}$ (Ave pour « Average »). C'est ce que présente l'équation 6.4, où :

- $SCVI_{Ave}$ représente l'indice de validité de contenu à l'échelle moyenne.
- n est le nombre total d'items dans l'instrument de mesure.
- $ICVI_i$ désigne l'indice de validité de contenu au niveau de l'item i .

$$SCVI_{Ave} = \frac{1}{n} \sum_{i=1}^n ICVI_i \quad (6.4)$$

$$SCVI_{UA} = \frac{|ICVI_i \geq \theta|}{n} \quad (6.5)$$

On calcule un autre type de **SCVI**, en évaluant la proportion d'items dans l'instrument qui ont atteint un certain seuil de **ICVI** que l'on note $SCVI_{UA}$ (UA pour Universal Agreement) comme le montre l'équation 6.5, où :

- $SCVI_{UA}$ représente l'indice de validité de contenu global basé sur l'Accord Universel.
- $|ICVI_i \geq \theta|$ est le nombre d'items dans l'instrument qui ont obtenu un indice de validité de contenu au niveau de l'item ($ICVI_i$) égal ou supérieur à un seuil prédéterminé θ .
- n est le nombre total d'items dans l'instrument de mesure ou le questionnaire.
- Le seuil θ pour l'**ICVI**.

Pour les $SCVI_{UA}$, le seuil θ est souvent fixé, de manière empirique, à 0.78 ou 0.80 comme cela a été le cas dans LAWSHE (1975). Il représente le niveau d'accord minimum entre les experts pour qu'un item soit considéré comme ayant une validité de contenu satisfaisante. On peut remarquer que ce seuil, pour LAWSHE (1975) était fixé pour le **CVI** globalement. Ici, il est appliqué pour chacun des items.

Ainsi, dans la proposition de YUSOFF (2019), un item est considéré comme valide en termes de contenu s'il atteint un seuil de consensus, prédéfini, parmi un panel d'experts.

Toutefois, IBRAHIM et al. (2022) font remarquer que cette méthode sur laquelle se base le calcul de l'**ICVI** ne tient pas compte de l'accord aléatoire et repose uniquement sur l'accord observé. Cette approche peut mener à une surestimation de la validité des items du questionnaire. L'**ICVI** ne fait aucune distinction entre un consensus véritable des experts et un accord dû au hasard. De ce fait, des items non pertinents pourraient être incorrectement validés si l'accord s'établit de manière fortuite. L'omission de cette correction pour l'accord fortuit compromet donc directement la précision et la robustesse de la validation de contenu.

Une nouvelle approche, appelée le Content Validity Template (**CVT**), proposée par IBRAHIM et al. (2022), s'appuie sur les travaux de YUSOFF (2019) pour résoudre le problème de la non-correction des accords dus au hasard dans les évaluations de validité de contenu. Au lieu de se reposer uniquement sur le **CVI**, qui ne corrige pas nécessairement les accords fortuits, le **CVT** utilise le coefficient AC_2 de Gwet. Il s'agit d'une méthode statistique qui évalue le degré d'accord en tenant compte des erreurs aléatoires. Ce coefficient, défini par GWET (2014), inclut l'erreur standard due à la variation des évaluateurs. Le coefficient AC_2 varie de valeurs négatives, indiquant un accord « pauvre », à 1, indiquant un accord « presque parfait », comme illustré dans la table 6.5.

Toutefois, pour assurer la validité des résultats, il est nécessaire de calculer l'interprétation inférentielle du coefficient AC_2 . Pour que la validation soit reconnue, le coefficient doit atteindre les niveaux « substantiel » ou « presque parfait ». Ces valeurs sont illustrées dans la table 6.5 et se situent dans une plage de valeur allant de 0.61 à 1 pour l' AC_2 . Cette approche fournit une mesure plus robuste de l'accord entre les évaluateurs, comme nous l'avons évoqué plus haut. De plus, cette méthode

Coefficient AC_2	Interprétation
< 0.00	Pauvre
0.00 – 0.20	Faible
0.21 – 0.40	Modéré
0.41 – 0.60	Modéré à bon
0.61 – 0.80	Substantiel
0.81 – 1.00	Presque parfait

TABLE 6.5 – Interprétation des valeurs du Coefficient AC_2 de Gwet

est appropriée pour des instruments comprenant de 5 à 300 items et pour un panel de 5 à 30 experts. La [CVT](#) propose aux non-statisticiens un outil statistique robuste et facile à utiliser, intégré dans un modèle de feuille de calcul.

Dans la section suivante, nous présentons la mise en œuvre des deux techniques de validation d'apparence et de contenu ainsi que les méthodes de calcul que nous avons choisi d'appliquer. Nous détaillons également les valeurs obtenues par l'utilisation du [CVT](#).

2 APPLICATION DES TECHNIQUES DE VALIDATION AU SMAQ

Dans la suite de ce chapitre, nous nous attacherons tout d'abord à réaliser la validation apparente du [SMAQ](#). Cette phase de validation vise à s'assurer que les définitions utilisées sont pertinentes, que les questions sont valides, utilisables et compréhensibles pour les répondants. Nous explorerons ensuite la validité de contenu du questionnaire pour identifier les questions avec lesquelles les experts du domaine sont le plus en accord. L'objectif est de recueillir des avis critiques qui nous permettront de développer une version améliorée du [SMAQ](#). Cette double validation est importante pour garantir que le questionnaire mesure ce qu'il est censé mesurer, mais aussi qu'il le fait de manière efficace, fiable et compréhensible.

2.1 Formulaires de validation

En effet, comme nous l'avons vu plus haut, l'évaluation de la validité apparente permet de déterminer à quel point le [SMAQ](#) « ressemble » subjectivement à une évaluation de la maturité d'un [SdMT](#). La validité de contenu évalue dans quelle mesure le [SMAQ](#) est pertinent et représentatif de cette évaluation. Deux nouveaux formulaires ont donc été élaborés pour réaliser la validation apparente et la validation de contenu. Au cours de la réalisation de l'enquête, les deux formulaires ont été fusionnés. Les items concernant la validité de contenu ont été proposés en premier et ceux concernant la validité apparente ont été proposés à la suite dans un même document en ligne.

Ainsi, la structure globale du document proposé pour l'évaluation était la suivante.

UNE PAGE DE PRÉSENTATION Une première page d'information sur le règlement de la Commission Nationale de l'Informatique et des Libertés (CNIL) et les instructions pour répondre aux questions était proposée.

UNE PAGE DE PROFIL DE L'EXPERT À la suite de cette première page se trouvait des questions concernant le profil de l'expert. Il pouvait indiquer son nom et son niveau d'expertise dans le domaine du génie logiciel, ainsi que son expérience de la norme ISO/IEC 62304. Les questions de ce premier contact avec l'expert sont présentées dans la table 6.6.

Description de votre profil et de votre expérience.
Merci d'indiquer votre nom <i>John Doe</i>
Avez-vous une formation en relation avec le développement logiciel? Non Oui
Quelle est votre expérience dans le domaine du développement logiciel? Moins de 2 ans De 2 à 5 ans Plus de 5 ans
Avez-vous déjà eu une activité en lien avec la norme 62304 ? Non Oui
Quelle est votre expérience avec la norme 62304 ? xxxxxxxxxxxxxxxxxx

TABLE 6.6 – Seconde page du formulaire d'évaluation de la validité du SMAQ

Les réponses à ces questions de présentation nous ont permis de vérifier le statut d'expert des répondants et de les classer en trois groupes. Ces trois groupes sont, 1) experts juniors, 2) experts intermédiaires, 3) experts séniors.

15 PAGES POUR LA VALIDATION DE CONTENU Pour procéder à la validation du contenu du SMAQ, un protocole a été établi conformément aux recommandations de YUSOFF (2019) déjà décrites plus haut. Pour chaque item du SMAQ, une question d'évaluation de la validité de contenu était proposée sur le format de l'exemple présenté dans la table 6.7. Il y avait donc, 156 questions correspondant aux 156 items du SMAQ qui représentaient le formulaire d'évaluation de la validité de contenu. Les questions dans ce formulaire n'ont pas été présentées sous la forme d'un regroupement par thème, comme dans le SMAQ, pour ne pas influencer l'expert. Elles ont été présentées par ordre alphabétique.

Artefact "Architecture logicielle" (Software Architecture)
Définition : L'architecture logicielle est un ensemble de concepts fondamentaux ou de propriétés d'un logiciel dans son environnement, matérialisés par ses éléments, ses relations, et par les principes de sa conception et de son évolution. Elle fournit un ensemble de règles pour définir la structure d'un système et les interrelations entre ses parties.
Trouvez-vous la question "Avez-vous défini une architecture logicielle?" ...
<ul style="list-style-type: none"> • Pas du tout pertinente • Peu pertinente • Pertinente • Très pertinente • Je ne sais pas

TABLE 6.7 – Exemple de questions présentes dans le formulaire d'évaluation de la validité de contenu.

UNE PAGE POUR LA VALIDITÉ APPARENTE À la fin du formulaire d'évaluation de la validité de contenu se trouvait le formulaire d'évaluation de la validité apparente. Les questions de ce dernier formulaire sont données dans la table 6.8.

Questions de validité apparente	Oui	Non
Les définitions fournies pour chaque question vous semblent-elles claires ?		
Les définitions fournies pour chaque question vous semblent-elles utiles ?		
Les questions vous semblent-elles correspondre au domaine du génie logiciel ?		
Selon vous, la relation entre les questions posées et la maturité d'une technologie pilotée par un logiciel vous semble-t-elle établie ?		

TABLE 6.8 – Évaluation de la validité apparente du SMAQ

En plus de ces questions pour l'évaluation de la validité apparente, dans le même formulaire, nous avons souhaité poser deux questions générales aux experts sur l'enquête elle-même. Ces dernières questions étaient les suivantes :

- Avez-vous des commentaires à formuler sur ce questionnaire d'évaluation de la maturité d'une [SdMT](#) ?
- Combien de temps avez-vous passé pour remplir le formulaire ?

Conformément aux préconisations de YUSOFF et al. (2021), elles nous ont permis de vérifier le temps passé par les répondants sur les questions des formulaires. De plus, nous avons pu recueillir les impressions générales sur le [SMAQ](#).

Le formulaire d'évaluation de la validation apparente et de contenu était donc constitué des sections suivantes :

- la page d'instruction et d'information sur les droits,
- la page de profil,
- les pages d'évaluation de la validité de contenu,
- la page d'évaluation de la validité de face.

2.2 L'enquête d'évaluation de la validité

Après la constitution des formulaires pour l'évaluation de la validité du [SMAQ](#), l'étape suivante de notre protocole a été de recruter les experts pour former le panel. Les critères suivants ont été établis. Premièrement, il était important que tous les experts soient eux-mêmes impliqués dans des projets de recherche mettant en œuvre des [SdMT](#) dans le domaine des [GMCAO](#). Il n'était pas nécessaire qu'ils soient spécialistes de la norme ISO/IEC 62304, mais ils devaient avoir une expérience pertinente dans le développement de telles technologies. De même, il n'était pas nécessaire que cette expérience inclue un aspect qualité. Une expérience significative pouvait compenser l'absence de formation formelle dans le domaine du génie logiciel.

Nous avons alors pu solliciter les experts retenus par courrier électronique, en les informant du but et des implications de cette enquête. Ils ont été recrutés dans les différents laboratoires du [LabEx CAMI](#) qui a été présenté dans le chapitre 1 (voir page 14). Après avoir obtenu le consentement écrit des experts par courrier électronique à la suite de la sollicitation, nous avons envoyé le lien vers l'enquête en ligne.

3 VALIDATION DE NOS RÉSULTATS

3.1 SMAQ

Pour mémoire, à la fin de la première étape décrite dans le chapitre 4, un total de 165 termes différents a été annoté dans le texte de la norme ISO/IEC 62304. L'étape de mappage a abouti à 156 SEA.

3.2 Profils des experts

Le panel consistait en 12 experts, tous provenant du domaine de la recherche dans les [SdMT](#). 11 d'entre eux sont membres de différents laboratoires de recherche spécialisés en GMCAO ou en biomécanique et l'un d'entre eux est un ingénieur logiciel d'une société spécialisée dans le domaine de l'assurance qualité des dispositifs médicaux. Parmi l'ensemble des experts, l'un d'entre eux était considéré comme un expert « junior », deux étaient considérés comme experts « intermédiaires » et 9 d'entre eux étaient considérés comme des experts « seniors ». La moitié du panel avait une expérience de la norme ISO/IEC 62304 (5 experts « seniors » et un experts « intermédiaire »). La figure 6.2 montre cette répartition.

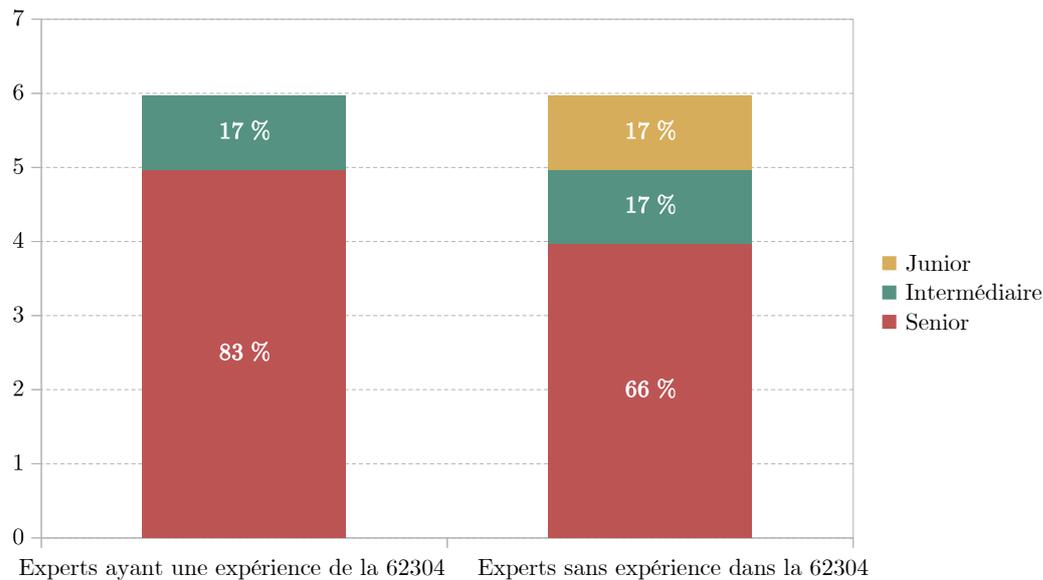


FIGURE 6.2 – Répartition des experts en fonction de leur expérience dans la norme ISO/IEC 62304

Onze des experts ont eu dans leur parcours professionnel, une formation dans le domaine de l'ingénierie logicielle, et un expert avait plus de cinq ans d'expérience dans le développement de logiciels dans un contexte de recherche et de développement.

3.3 Validé apparente

Ci-dessous les réponses données à l'évaluation par les experts de la validité apparente du SMAQ sont présentées :

- 7 experts sur 12 (59%) ont considéré que les définitions fournies pour chaque définition étaient claires.
- 11 experts sur 12 (92%) ont considéré que les définitions fournies sont utiles.
- 12 experts sur 12 (100%) ont considéré que les questions étaient pertinentes pour le domaine du génie logiciel.
- 10 experts sur 12 (83%) ont considéré que les questions étaient bien en relation avec l'évaluation de la maturité.

Ces réponses révèlent plusieurs aspects concernant la pertinence du SMAQ et son utilité dans la mesure de la maturité des SdMT.

CLARTÉ DES DÉFINITIONS (59%)

La clarté des définitions est très importante et fait, naturellement, partie de la vérification de la validité apparente. Elle est nécessaire pour éviter toute ambiguïté dans l'interprétation des questions et des réponses et c'est ce que nous voulions vérifier par les questions de la table 6.8. La révision des définitions les moins claires en consultation avec les experts pourrait améliorer cet aspect. Le fait que 7 experts sur 12 trouvent les définitions claires suggère qu'il y a encore une place significative pour l'amélioration.

UTILITÉ DES DÉFINITIONS (92%)

Le fait que 11 experts sur 12 considèrent les définitions comme utiles indique une forte reconnaissance de leur valeur ajoutée dans le contexte de l'évaluation de la maturité des SdMT.

PERTINENCE DES QUESTIONS (100%)

Il y a unanimité des experts quant à la pertinence des questions pour le domaine du génie logiciel. Ceci est un indicateur fort que le SMAQ est bien ciblé et couvre les items essentiels nécessaires à l'évaluation de la maturité des SdMT. Cela va dans le sens d'une très bonne adéquation du questionnaire avec le domaine de l'évaluation de la maturité des SdMT.

RELATION AVEC L'ÉVALUATION DE LA MATURITÉ (83%)

10 experts sur 12 considèrent que les questions sont bien en relation avec l'évaluation de la maturité. Cela signifie que la majorité des experts trouvent que le questionnaire aborde efficacement les aspects primordiaux de la maturité technologique. Cependant, comme pour la clarté des définitions, ce taux indique probablement aussi la possibilité d'améliorer certaines questions.

Comme nous l'avons rapporté plus haut dans ce chapitre, la validité donne une première idée sur des aspects globaux concernant la perception du SMAQ par les experts.

3.4 Validité de contenu

Les réponses des experts ont été enregistrées en utilisant le CVT fourni par IBRAHIM et al. (2022). Dans notre méthode, l'indice de validité de contenu pour les items (ICVI) est calculé comme décrit dans l'équation 6.3. Nous avons retenu un seuil de 0.80 pour accepter ou rejeter un item. L'analyse a produit les résultats suivants :

- Le Coefficient d'Accord corrigé pour le hasard, Gwet AC_2 , était de 0,690 et a été fourni comme résultat d'un calcul intégré au CVT.
- Ce coefficient est utilisé pour évaluer la concordance en termes de probabilité inférentielle. Avec une valeur de 0,690, ce coefficient est classé comme **substantiel** (voir la table 6.5), indiquant un niveau d'accord robuste entre les évaluateurs. Selon les critères que le CVT fourni à partir des réponses saisies, cette valeur montre que l'accord dépasse la probabilité minimale de 95 % pour être considéré comme substantiel.

Suite à la saisie des résultats collectés auprès des experts dans le CVT nous avons obtenu la liste des ICVI. Ils ont mis en évidence les questions acceptées, c'est-à-dire celles qui ont une valeur d'ICVI supérieure ou égale au seuil $\theta = 0.8$. La figure 6.3 montre les distributions des réponses. Au total, 91 questions du SMAQ sur 156 ont reçu un ICVI au-dessus du seuil de valeur 0,8 (27 avec un ICVI de 1,0, 32 avec un ICVI de 0,92 et 32 avec un ICVI de 0,83). Cette distribution montre un bon niveau d'accord entre les experts, malgré leur diversité. On voit également que bien que les experts se répartissent entre « experts juniors », « experts intermédiaires » et

« experts seniors », cela n’a pas eu d’effet. 91 questions du SMAQ ont ainsi pu être validées comme pertinentes.

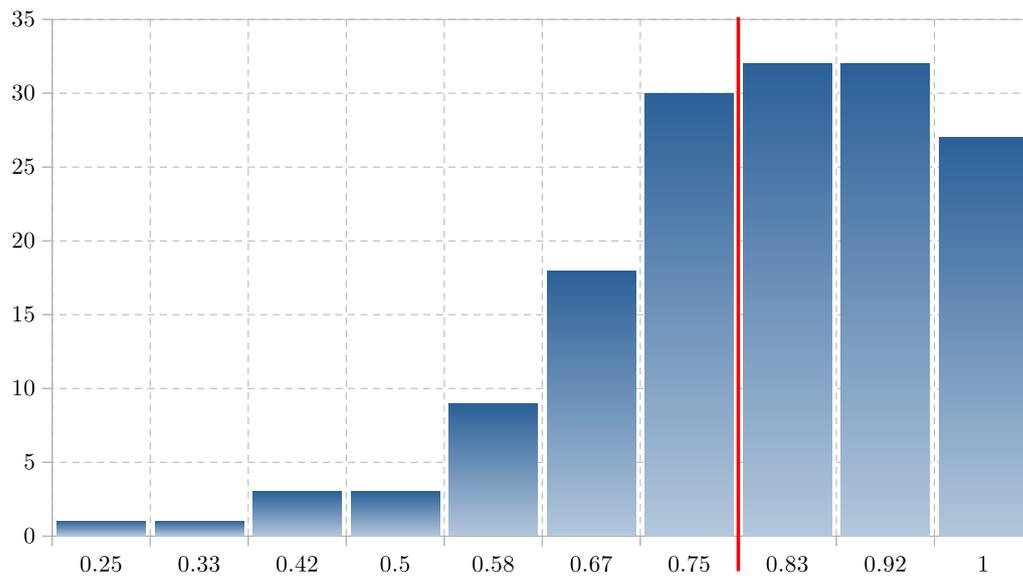


FIGURE 6.3 – Distribution des CVI calculés à partir des réponses des experts. La ligne rouge montre la valeur de coupure entre les questions acceptées et rejetées à la valeur seuil $\theta = 0.8$.

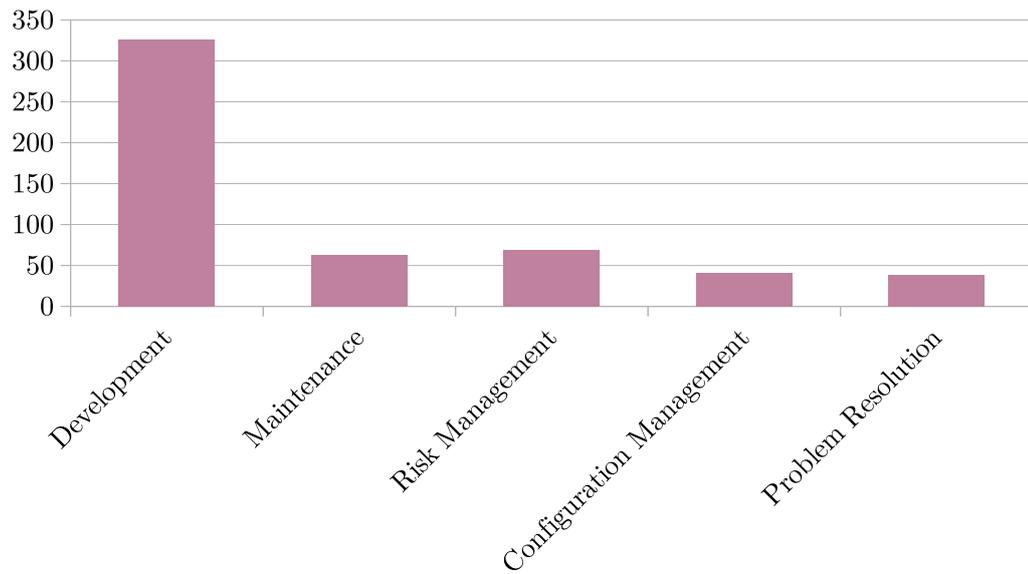
3.5 Mise à jour du SMAQ

Le SMAQ a été mis à jour en utilisant les résultats de validité de contenu en ne conservant que les 91 questions validées par les experts.

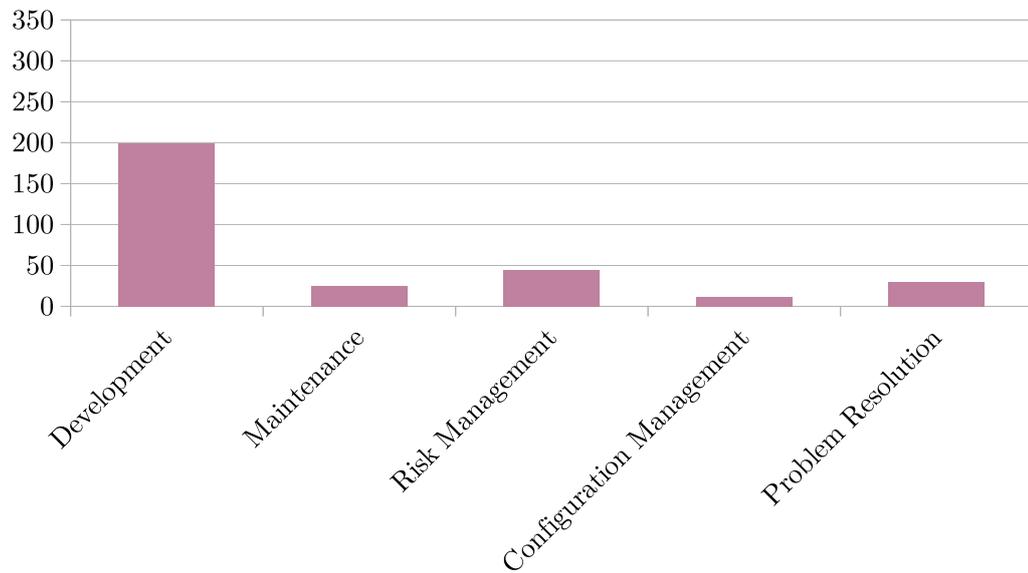
IMPACT SUR LE SCORE

En considérant les 91 SEA correspondants aux questions validées, le poids de chacun de ces SEA a été recalculé en suivant la même méthodologie que celle expliquée au chapitre 4 au moyen de l'équation 4.2 (page 88). Le nombre total de citations est passé de $N = 537$ citations pour 156 SEA extraits à $N = 316$ pour les 91 SEA validés.

La figure 6.4 montre la différence de répartition du nombre de citations des 91 SEA validés parmi les cinq clauses de la norme. La clause 5, « Développement », reste la clause principale avec 199 citations, ce qui augmente légèrement son importance relative de 61% à 64%.



(a) Valeurs avant la validation de contenu



(b) Valeurs après la validation de contenu

FIGURE 6.4 – Nombre de citations pour chaque activité principale de la norme

Le score de maturité a, lui aussi, été mis à jour. Le nombre de SEA requis pour obtenir un score minimum de 50% est passé de 20 dans la version initiale du SMAQ (voir Fig. 6.5a) à 12 dans la version validée (voir Fig. 6.5b).

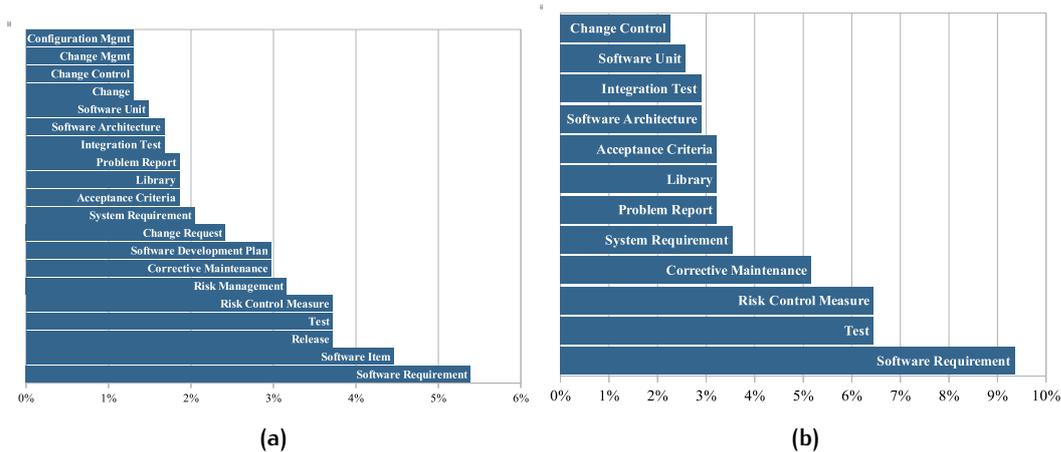


FIGURE 6.5 – Les SEA les plus cités sur le modèle extrait (à gauche) et le modèle validé (à droite) représentant un poids total minimal de 50% du score. L’axe des x est le poids de chaque SEA. a) Les 20 SEA les plus cités après extraction représentent un total de 50,2% du score. b) Les 12 SEA validés les plus cités représentent un total de 51,3% du score.

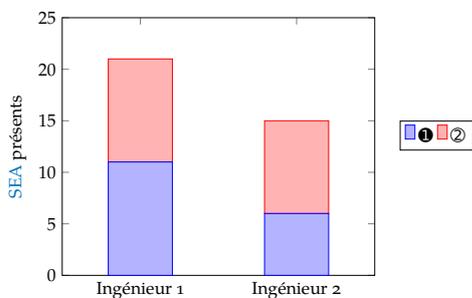
IMPACT SUR L’ÉVALUATION DE MATURITÉ DE FOLLOWKNEE

Ingénieur 1	31%
Ingénieur 2	28%

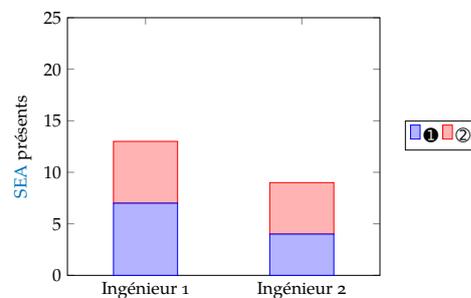
(a) Tableau des scores par ingénieurs avant validation

Ingénieur 1	38%
Ingénieur 2	31%

(c) Tableau des scores par ingénieurs après validation



(b) Distribution des SEA de type 1 & 2 parmi les réponses positives avant validation



(d) Distribution des SEA de type 1 & 2 parmi les réponses positives après validation

FIGURE 6.6 – Impact des mises à jour du SMAQ sur l’évaluation de la maturité FollowKnee sur la base des mêmes réponses.

La figure 6.6 illustre l’effet significatif des révisions apportées au SMAQ sur les scores de maturité des ingénieurs impliqués dans le projet FollowKnee en reportant les réponses au SMAQ initial.

Le recalcul des scores de maturité, comme illustré dans les figures 6.6a et 6.6c de la figure 6.6, montre une augmentation de 7% pour l’ingénieur 1, passant de 31% à 38%, et de 3% pour l’ingénieur 2, passant de 28% à 31%. Cette augmentation

résulte directement de la suppression des **SEA** associés aux questions éliminées du **SMAQ**. En concentrant l'évaluation sur les **SEA** jugés essentiels pour la maturité technologique, le score ajusté offre une représentation plus favorable du niveau de développement atteint par le projet.

Les graphiques 6.6b et 6.6d montrent une distribution modifiée des **SEA** de type ① et de type ② après la mise à jour du **SMAQ**. Malgré la réduction globale du nombre de **SEA**, la proportion relative entre les types de **SEA** reste similaire. C'est aussi ce que l'on observe dans la distribution des **SEA** selon les clauses de la norme (voir figure 6.4) qui garde la même cohérence. Cela indique que la suppression s'est uniformément appliquée à travers les différentes catégories. Cette cohérence dans la répartition des **SEA** montre que les ajustements du **SMAQ** ont été appliqués d'une manière qui n'altère pas l'intégrité analytique de l'outil d'évaluation.

4 SYNTHÈSE DE LA VALIDATION DU SMAQ

La validation visait à démontrer la pertinence et la fiabilité du **SMAQ** pour l'évaluation de la maturité technologique des **SdMT**. Dans cet objectif, un panel d'experts spécialisés dans le domaine des **SdMT** a été constitué. Le nombre d'experts choisis suivait sur les recommandations de YUSOFF (2019) pour l'application de la méthode **CVI** grâce au **CVT**.

Selon YUSOFF (2019), un minimum de six experts est nécessaire pour garantir des valeurs de **CVI** fiables, tandis que le nombre maximum recommandé est de dix. Ce seuil supérieur est préconisé afin de garantir une diversité adéquate des perspectives tout en facilitant une gestion efficace du processus de validation. Toutefois, le cadre étendu fourni par IBRAHIM et al. (2022) permet d'augmenter ce nombre jusqu'à trente experts.

En mobilisant douze experts, notre étude se situe donc bien dans l'intervalle optimal recommandé pour une validation rigoureuse et représentative.

4.1 Experts

Les répondants étaient principalement des experts expérimentés avec une expérience significative dans le développement de technologies médicales pilotées par logiciel (11 experts sur 12). Ces profils semblent appropriés pour la validation du **SMAQ**.

De plus, il était important que tous les experts n'aient pas connaissance de la norme, car le **SMAQ** a pour objectif d'être un outil d'évaluation simple et utilisable en environnement de recherche. Comme le montre la figure 6.2, 50% des experts interrogés avaient une expérience préalable avec la norme ISO/IEC 62304, tandis que les autres 50% n'en avaient pas.

Les **ICVI** des items ont validé 91 questions du **SMAQ**, et par conséquent 91 **SEA**. La distribution de ces 91 **SEA** en termes de leur nombre de citations, et donc de leur poids relatif, parmi l'activité standard (voir Fig. 4.11 page 87) n'a pas fondamentalement changé même si les questions étaient présentées dans un ordre alphabétique sans regroupement d'activités. De même que cette mise à jour n'a pas modifié la cohérence des réponses des ingénieurs du projet « FollowKnee ». Cela indique que

l'évaluation proposée en utilisant une représentation de la norme ISO/IEC 62304 uniquement à travers les [SEA](#), indépendamment des exigences, reste robuste au filtre des experts. En d'autres termes, l'importance relative de l'activité calculée selon le nombre des citations des [SEA](#) reste très similaire après validation.

4.2 Validité apparente et de contenu

Il convient de noter le bon accord entre les experts tel que mesuré par le [CVT](#). Cela indique le bon niveau de fiabilité (voir section [3.4](#)) de l'accord sur l'ensemble de données en fournissant une valeur de AC_2 définie comme *substantielle*. En outre, le fait que les profils d'experts soient diversifiés tandis que la fiabilité de l'accord interévaluateurs est bonne, semble montrer que le questionnaire est pertinent pour différents types de profils d'utilisateurs.

Bien que le coefficient de Kappa soit souvent utilisé pour quantifier l'accord interjuges au-delà de ce qui pourrait être dû au hasard, il n'a pas été appliqué ici. En effet, le niveau élevé d'accord général et le format dichotomique simple des réponses (oui/non) permettent aux calculs de pourcentage d'être directement significatifs et suffisamment robustes pour notre évaluation initiale. L'utilisation du coefficient de Kappa pourrait ajouter une complexité inutile et n'apporterait pas d'informations supplémentaires significatives dans notre cas.

Sur la base des pourcentages calculés, on peut dire que les experts s'accordent unanimement à dire que les questions utilisées dans le [SMAQ](#) sont pertinentes et étroitement liées au domaine du génie logiciel. De plus, 83% des experts considèrent que le [SMAQ](#) est approprié pour évaluer la maturité des [SdMT](#) ce qui valide sa destination. Il convient de noter que tous les experts ne partagent pas la même opinion sur la clarté des définitions fournies. Cela a ainsi pu influencer la manière dont les experts ont répondu à la question de la pertinence.

Les réponses des experts ont validé le [SMAQ](#) en termes de validité apparente. Ce qui confirme que le [SMAQ](#) semble être pertinent pour évaluer la maturité des [SdMT](#) et que les questions sont raisonnables, sans ambiguïté et claires. Ainsi, la décision que nous avons prise de nous en tenir à une évaluation plus directe et moins formelle de l'accord, en nous basant principalement sur les pourcentages, est adéquate pour répondre aux besoins de la validation apparente.

Les 91 SEAs restants seront utilisés pour développer une version 2 du [SMAQ](#). Les définitions doivent être améliorées et clarifiées. Elles sont tirées du [SEVOCAB](#), et sont elles-mêmes extraites de 188 sources ISO et IEEE. En tant que telles, elles peuvent contenir des ambiguïtés ou des éléments trop étroitement liés au contexte d'assurance qualité d'où elles proviennent. Nous avons indiqué ce problème de « silos » dans la terminologie du génie logiciel en étudiant les travaux de HENDERSON-SELLERS et al. (2013) au chapitre 2 à la section 3 (page 49). Il est donc nécessaire de simplifier les définitions et de les adapter au public ciblé par le [SMAQ](#). Ajouter des illustrations aux définitions (graphiques, exemples, etc.) peut également aider à améliorer la compréhension des SEAs.

Enfin, comme YUSOFF et al. (2021) l'indiquent, il sera également nécessaire de normaliser les scores et leur interprétation. Cela pourra être fait au moyen d'une étude plus large, autour d'un plus grand nombre de projets et sur le même format que pour le prétest réalisé avec le projet « FollowKnee ». Cette étape représente une

étude en soi et demande un travail de grande ampleur dans notre cas, car il doit être mené auprès d'un nombre significatif de projets de recherche impliquant potentiellement un grand nombre d'intervenants sur une période de temps permettant de couvrir la vie de plusieurs projets. Une telle étude à grande échelle permettrait de couvrir davantage le cadre de validation préconisé par Cook et al. (2006) avec l'ensemble des preuves qu'ils préconisent.

5 CONCLUSION

Ce chapitre a consolidé la validité du SMAQ, instrument que nous proposons pour évaluer la maturité des SdMT. Grâce à une démarche méthodique articulée autour de la validité apparente et de la validité de contenu, nous avons pu affiner l'outil pour qu'il reflète avec précision les exigences et les standards de l'ingénierie logicielle, notamment en faisant intervenir des experts du domaine. Les résultats obtenus attestent de la pertinence du SMAQ, avec une cohérence marquée entre les jugements des experts et les objectifs du questionnaire.

Cette validation a été menée pour les deux premiers types indispensables de validité (la validité apparente et la validité de contenu). Il est important que ce travail soit poursuivi par une validité de construit

En effet, la validité de construit est essentielle pour s'assurer que le SMAQ mesure réellement la maturité des technologies médicales pilotées par logiciel de manière précise et cohérente avec les théories sous-jacentes. Elle permet de garantir que les différentes dimensions du concept sont correctement évaluées et de s'assurer de la fiabilité et de la validité des résultats obtenus.

Bien qu'étant limitée, cette première validation était une étape indispensable pour vérifier notre modèle théorique de la maturité des SdMT par rapport à la réalité terrain. Elle permet d'ouvrir la voie à la validation de construit, qui n'aurait pu être envisagée aujourd'hui sans cette première validation.

L'approbation du questionnaire par un panel d'experts et l'application des principes de validation énoncés dans les recommandations de YUSOFF (2019) et TAHERDOOST (2016) confirment que le SMAQ est un outil fiable et valide pour mesurer la maturité des SdMT. Après la mise en œuvre de tests à l'échelle, il sera possible d'envisager son intégration dans des processus d'évaluation réguliers, pour des technologies dans l'environnement de la recherche et du transfert en GMCAO.

7

DISCUSSION ET CONCLUSION

1	Synthèse des contributions	133
2	Limitations et piste d'améliorations	138
3	Perspectives	140
4	Conclusion	141

Les travaux réalisés dans le cadre de cette thèse sont un élément nouveau pour le domaine des [GMCAO](#) et les deux laboratoires impliqués dans ces réalisations. Ils ont donné lieu à une publication ([HABERBUSCH et al. 2023](#)) dans le cadre de la 45e conférence internationale annuelle de la Société d'ingénierie en médecine et biologie (Engineering in Medicine & Biology Society ([EMBC](#))) en 2023.

Ce dernier chapitre fait la synthèse des contributions et conclue sur ces travaux qui pourront servir de base solide pour des projets de recherche à venir.

1 SYNTHÈSE DES CONTRIBUTIONS

L'objectif général de ce travail est l'évaluation de la maturité des [SdMT](#). Il s'est concrétisé par le développement et la validation d'un modèle pour évaluer cette maturité pendant la phase de recherche (Chapitres [2](#) et [3](#)). La création des outils d'évaluation, détaillée dans le Chapitre [4](#), a impliqué l'extraction des éléments d'artefacts logiciels de la norme ISO/IEC 62304 et le développement du questionnaire SMAQ pour quantifier la maturité technologique des SdMT. Le SMAQ a été testé sur un projet réel, « FollowKnee », afin d'évaluer son application pratique (Chapitre [5](#)). Enfin, la validation a été atteinte par des méthodes statistiques et une enquête auprès d'experts pour vérifier la fiabilité et la validité du SMAQ (Chapitre [6](#)).

1.1 Une démarche en trois étapes

Nous avons mis en œuvre une méthodologie en trois étapes : la création des outils d'évaluation, le prétest des outils sur un cas réel et la validation des outils au moyen de méthodes statistiques, comme cela est rappelé ci-dessous.

Au chapitre [4](#), nous décrivons la première étape. Elle a permis le développement des outils d'évaluation. Après avoir extrait les [SEA](#) de la norme ISO/IEC 62304, nous avons mis au point une méthode pour calculer le score de maturité technologique d'une [SdMT](#). La norme ISO/IEC 62304 a été utilisée pour identifier les [SEA](#) et le référentiel [SEVOCAB](#) a été utilisé pour vérifier la pertinence des [SEA](#) dans le domaine du génie logiciel. Dans le même temps, nous avons créé le [SMAQ](#). Il s'agit de l'instrument qui permet de recueillir les informations nécessaires au calcul du score auprès des équipes en charge du développement de la [SdMT](#). Ces deux ou-

tils, que sont le score et le SMAQ, nous permettent de quantifier la maturité d'une SdMT basée sur les SEA qui la constituent.

Dans le chapitre 5 nous décrivons la deuxième étape de nos travaux où nous expérimentons nos outils d'évaluation dans le cadre d'un projet de développement d'une SdMT en environnement réel. Cette étude de cas sur le projet « FollowKnee » a permis l'application pratique du modèle d'évaluation de maturité pour tester son efficacité et recueillir des données empiriques. Lors de cette expérimentation, l'analyse des résultats a permis l'évaluation de la concordance entre les niveaux de maturité observés et les attentes basées sur le modèle théorique. L'interview menée auprès des experts nous a permis de mieux interpréter ces résultats et de tirer des conclusions pour améliorer le SMAQ.

Enfin, le chapitre 6 décrit la troisième étape de nos travaux. Il s'agissait de mener à bien la validation statistique de nos outils d'évaluation de la maturité et notamment la validation du SMAQ. Pour cela nous avons utilisé des techniques statistiques reconnues et largement utilisées pour valider la fiabilité et la validité de tels instruments. Nous avons donc mené une enquête auprès d'experts du domaine. Elle nous a permis de recueillir leurs avis pour affiner le questionnaire et le modèle d'évaluation de la maturité.

Les chapitres 2 à 3 contribuent à une compréhension complète du processus d'évaluation de la maturité. Ils nous permettent de décrire une approche intégrée en partant de la théorie vers l'application pratique et enfin vers la validation empirique et statistique du modèle proposé.

Cette section revient sur les contributions et les méthodologies employées pour les atteindre. Il met en évidence l'approche systématique et rigoureuse que nous avons adoptée pour aborder la question complexe de la maturité technologique des SdMT.

1.2 Développement du modèle d'évaluation de maturité

Nous avons réalisé l'extraction des SEAs clés de l'ISO/IEC 62304. Ils ont été utilisés pour construire un questionnaire d'évaluation de la maturité technologique des SdMT.

EXTRACTION ET QUANTIFICATION DES SEA POUR L'ÉVALUATION DE LA MATURITÉ

Durant la première étape, lors de l'extraction, nous avons extrait 165 termes « candidats » pour représenter les SEA. Sur les 165 termes annotés dans la norme ISO/IEC 62304, un mappage réussi avec les termes définis dans SEVOCAB a permis d'identifier 156 SEA pertinents pour évaluer la maturité des SdMT. Ces SEA représentent divers aspects du cycle de vie du développement logiciel spécifié par la norme. Lorsqu'on relocalise les termes correspondants dans le texte de la norme, on peut déterminer quels SEA sont produits par ses activités. On a choisi de mesurer l'importance relative des SEA dans le score, en utilisant leur fréquence de citation dans le texte. La construction du score de maturité s'appuie sur la présence et la fréquence des citations de ces SEA dans la norme. Par extension, en réalisant la somme des citations pour une clause donnée on peut en déduire son importance relative pour le score. Par exemple, la clause 5, « Développement », est celle qui contient le plus

grand nombre de citations (326 citations avant validation). C'est la clause la plus importante dans le score final de maturité.

Le score étant calculé par rapport au poids relatif de chaque **SEA**, il est possible de caractériser l'importance relative de chaque **SEA** dans le score.

Notre analyse a montré qu'avant l'étape de validation, les 20 **SEA** les plus cités contribuent à 50,2% du score de maturité total. Cette observation met en évidence que certains **SEA** sont particulièrement significatifs pour atteindre un niveau plus élevé de maturité technologique.

CORRÉLATION ENTRE EFFORT DE DÉVELOPPEMENT ET TYPES DE SEA

Selon notre modèle, atteindre un score maximal de maturité nécessite d'augmenter le niveau de **SEA** avec des **SEA** de type ② tout en produisant des **SEA** de type ①. La progression peut être séquentielle, en se concentrant sur des **SEA** clés pour rapidement atteindre 50 % de maturité.

Cependant, les 50 % restants exigent une production plus importante de **SEA** ayant des poids très bas, comme le montre la figure 4.12 (page 90). Cela peut aider à définir une stratégie dans un projet de recherche en orientant les travaux vers un transfert potentiel ou alors en cherchant à développer davantage la technologie pour l'approfondir dans un domaine spécifique de la recherche, par exemple.

Pour obtenir concrètement la corrélation avec l'effort, nos résultats permettent d'entrevoir des pistes. La corrélation entre la proportion de **SEA** de type ② et de type ① et l'effort nécessaire pour atteindre un certain niveau de maturité suggère deux niveaux d'effort distinct pour l'élévation du niveau de maturité.

Les 91 **SEA** obtenus après validation montrent que le score de maturité au-delà des 51% reste accessible au prix d'un effort conséquent, pour les mêmes raisons qu'avant validation : le nombre de **SEA** de faible poids étant encore majoritaire.

STRUCTURE ET ÉLÉMENTS CONSTITUTIFS DU MODÈLE DE MATURITÉ

Suivant la terminologie de description des modèles de maturité dans les **SI** définie dans LASRADO et al. 2015, une structure complète de modèle de maturité pour les systèmes d'information peut être décrite par cinq éléments principaux que nous avons vus au chapitre 2 :

- Les *questions d'évaluation* sont la manière dont la maturité est évaluée.
- Les *niveaux de maturité* sont les différents niveaux qui structurent le modèle de maturité et à travers lesquels une technologie progresse pour améliorer sa maturité.
- Le *chemin de maturité* est le chemin à suivre pour améliorer le niveau de maturité.
- Les *dimensions* représentent les différents aspects ou domaines spécifiques qui sont évalués dans le cadre du modèle de maturité et qui évoluent le long du chemin de maturité.
- Les *sous-catégories* sont des sous-éléments qui constituent la maturité. Cet élément n'est pas obligatoire pour construire un modèle de maturité complet.

Dans cette étude, nous nous sommes positionnés sur quatre de ces éléments, pour faire des propositions de structure pour un modèle de maturité complet :

- Les *questions d'évaluation* sont listées dans le SMAQ lui-même. De plus, le regroupement des questions fournit une manière pratique de présenter les questions.
- Les *niveaux de maturité* sont directement donnés par le score calculé à partir des réponses au SMAQ, de 0% à 100%. Comme nous l'avons esquissé dans le chapitre 2 à la section 2.5 page 45, ils sont au nombre de trois.
 - *Développement initial*
 - *Préparation aux essais cliniques*
 - *Pré-transfert*
- Le *chemin de maturité* est fourni par l'ajout progressif de tous les SEAs cités, bien que différents chemins puissent être suivis (implémenter les SEA de type ① en premier, implémenter les SEA ayant le plus de poids en premier ...). Cette multiplicité des chemins était montrée par LASRADO et al. (2015) comme une conséquence de la mise en œuvre de modèle de maturité non linéaire.
- Les *dimensions* sont caractérisées par les SEA qui représentent les niveaux.

UNE ÉVOLUTION DE LA MATURITÉ PROGRESSIVE ET NON LINÉAIRE

Ce modèle de maturité obtenu est représenté par la figure 7.1. Cette figure présente une progression possible parmi beaucoup d'autres. Par exemple, on peut très bien avoir un prétransfert avant un préparation aux essais cliniques. De sorte que cette étape ne soit pas gérée au laboratoire.

Nous y retrouvons l'échelle représentant les phases de la figure 2.2 page 30, agrandie au niveau de notre objet d'étude. Il s'agit de la phase de recherche et la période couverte par les ajustements techniques en démarrage de transfert de technologies. Elle se découpe en trois parties selon les niveaux de maturité. Chaque niveau peut-être caractérisé par un ensemble de SEA (D1, D2, D3) représentant les dimensions de maturité. Les scores correspondent à ces niveaux et dans l'objectif d'atteindre le transfert peuvent être considérés dans un sens croissant.

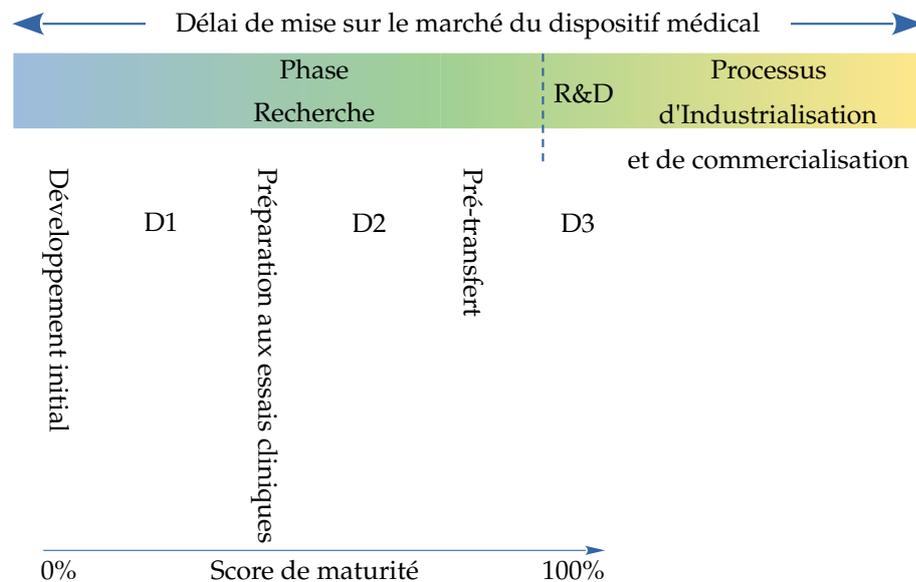


FIGURE 7.1 – Avec ce modèle, l’extrémité de l’échelle à 100% ne constitue pas nécessairement l’objectif à atteindre. Mais, il a permis de donner une référence de maturité pour une issue possible au projet.

Cette présentation du modèle de maturité n’impose pas de suivre une progression linéaire. En ce sens, nous nous rapprochons du cadre défini dans les travaux de BLEVINS et al. (2018) et MOREAU-GAUDRY et al. (2010).

Les aller-retour dans les projets de recherche peuvent faire évoluer les scores dans les deux sens. Cela est compensé par le fait que l’évaluation de score de maturité peut être effectuée à tout moment de la vie de ces projets.

LES SEA DANS LE MODÈLE DE MATURITÉ

Dans ce contexte, on peut se demander comment évoluent les SEA au sein de cette structure. Et notamment, si les SEA sont différents d’une dimension à une autre ou peuvent être présents dans différentes dimensions en même temps.

Dans ce cas, on peut revenir à la définition des SEA de MÉNDEZ FERNÁNDEZ et al. (2019). Les auteurs nous apportent une réponse au travers de la notion d’équivalence et de raffinement des SEA. En utilisant ces principes, nous pouvons organiser les SEA dans les dimensions en commençant par les SEA « raffinables » et en progressant vers leur version raffinée (voir l’exemple du chapitre 2, section 1.5 page 58)

La nature de ce raffinement reste à déterminer spécifiquement pour chaque SEA et notamment en fonction de s’il s’agit d’un SEA de type ❶ ou un SEA de type ❷.

En effet, un SEA de type ❶ a une marge de raffinement limitée puisqu’il est dans un état fortement formalisé selon notre définition. Alors que ce niveau de formalisme est moins important pour les SEA de type ❷. Cela laisse une plus grande possibilité d’évolution pour les SEA de type ❷ en augmentant le niveau de formalisme.

2 LIMITATIONS ET PISTE D'AMÉLIORATIONS

2.1 La classification des SEA

Durant nos recherches, nous avons évalué et classé les SEA selon leur niveau de formalisation, en prenant en compte l'approche utilisée pour structurer leur contenu syntaxique. Cela varie du formalisme rigoureux du code logiciel à la flexibilité du langage naturel. En conséquence, des SEA tels que « Software Item », ont été classés en classe ① car leur formalisme repose sur le langage de programmation, considéré comme très formel. D'autre part, des SEA comme « Problem Report » ont été classés en classe ② car ils sont plus proches du langage naturel, pouvant laisser plus de place à des interprétations.

La difficulté réside dans la compréhension des évolutions entre les différents niveaux, car nous ne traitons pas des mêmes SEA à chaque fois. Pour clarifier, prenons l'exemple des deux SEA mentionnés précédemment. Le SEA « Software Item », en tant que code au stade du *Développement initial*, maintiendra probablement une structure similaire lorsqu'il atteindra le stade *Prétransfert*. En revanche, le SEA « Problem Report » subira des modifications structurelles significatives pour se conformer aux exigences du formalisme réglementaire.

Dans les deux cas, nous observerons des évolutions significatives, avec des différences marquées dans les artefacts de chaque niveau. Pour le SEA « Software Item », même si les transformations peuvent sembler mineures, les exigences du cycle de vie en environnement réglementé influenceront sans doute sa structure syntaxique finale. Cela entraînera des adaptations telles que le redécoupage fonctionnel ou l'adoption de bonnes pratiques spécifiques. De la même manière, le SEA « Problem Report » devra s'adapter à un cadre réglementaire plus strict. Il devra respecter des normes de bonnes pratiques plus rigoureuses et un formalisme qualité plus élaboré, comme des éléments de traçabilité uniques, inexistants auparavant. Ces changements illustrent la transformation au moyen du raffinement qui par itération peut aboutir à une équivalence entre les versions des SEA à différents niveaux.

Il ressort clairement de notre analyse que, bien que le classement actuel des SEA puisse sembler peu intuitif, il a des implications significatives sur la perception de la progression dans le modèle de maturité que nous proposons. Il est donc important de travailler à rendre ce classement plus intuitif afin de faciliter l'adoption et la compréhension du concept de maturité guidé par les SEA.

Nous avons identifié une autre limitation dans nos résultats concernant l'adéquation des SEA avec les réalités industrielles. Il existe une divergence entre les SEA issus de la recherche académique qui sont certainement très différents de ceux utilisés dans les milieux réglementés industriels. Les SEA académiques nécessitent un raffinement significatif pour répondre aux exigences réglementaires, ce qui peut limiter leur applicabilité directe en industrie et leur adoption en contexte de recherche. Il est nécessaire de bien étudier les différents raffinements possibles de SEA afin de mieux les qualifier (et pourquoi ne pas inclure les niveaux de raffinement dans le score).

2.2 Questionnaire d'évaluation

Des limitations ont également été observées sur certains aspects.

Notre étude a révélé une variabilité dans les réponses des ingénieurs concernant la présence de certains [SEA](#), reflétant des perceptions divergentes qui pourraient influencer l'évaluation de la maturité technologique.

L'utilisation du [SMAQ](#) a également exposé des problèmes de clarté dans les définitions fournies, ce qui a été souligné par certains experts. Cette ambiguïté pourrait compromettre la précision de l'évaluation de la maturité. Cette variabilité peut être expliquée en reprenant le concept du pragmatisme dans la définition de [LASRADO et al. \(2015\)](#). Il faudrait travailler les définitions et les illustrations pour rendre plus claires et compréhensibles les questions du [SMAQ](#). Cette limitation pourrait être également contournée en proposant aux ingénieurs de partager le même espace sémantique au travers d'une meilleure communication et de formation autour des objectifs du projet.

Un regroupement des questions dans le [SMAQ](#) doit être proposé pour refléter les divers objectifs des projets de [SdMT](#), tels que la preuve de concept, le transfert de technologie, la collaboration de recherche et la faisabilité clinique. Chaque objectif devrait être traduit en une liste spécifique de [SEA](#) à produire, adapté à chaque étape du projet. Par exemple, l'analyse des risques peut être jugée non pertinente au début d'un projet de recherche, mais devient déterminante lors de l'approche des tests cliniques. Organiser les [SEA](#) en listes pondérées en fonction des différents objectifs facilitera une évaluation plus précise de la maturité technologique et fournira des lignes directrices claires basées sur les [SEA](#).

2.3 Modèle de maturité

Ces idées de regroupement pourraient également offrir des pistes pour la construction de sous-catégories pour les domaines des différents niveaux, voir créer de nouveaux domaines. Notre étude ne comporte pas de sous-catégories. On pourrait les caractériser en établissant une correspondance avec les clauses de la norme ISO/IEC 62304. Ainsi, les [SEA](#) présents dans une dimension pourraient être vus comme la couverture qu'ils représentent par rapport aux clauses de la norme ISO/IEC 62304. Cela aurait une signification pour les deux derniers niveaux qui nécessitent un rapprochement vers la réglementation.

Il est également essentiel de poursuivre le raffinement des [SEA](#) dans les travaux futurs, notamment en intégrant mieux ces concepts avec la nature des [SEA](#) comme définis dans [LASRADO et al. \(2015\)](#). Il faudra également envisager l'évolution des listes d'artefacts vers un modèle d'artefacts robuste.

L'élaboration d'« indicateurs » de maturité, pour fournir des lignes directrices pratiques comme proposées dans [HAUSCHILD et al. \(2022\)](#), est une autre direction importante pour les travaux futurs. Ces indicateurs, basés sur des listes spécifiques de [SEA](#) adaptées à des objectifs précis tels que la réutilisation, la modularité et les tests cliniques, amélioreront l'interprétation des résultats d'évaluation de la maturité et permettraient de construire les sous-catégories comme nous venons de l'aborder.

Selon LASRADO et al. (2015), il existe un manque de validation empirique robuste dans les modèles de maturité analysés. Bien que cette critique s'adresse aux modèles eux-mêmes et non à leur étude, cela implique une limitation importante de la littérature qui peut affecter la fiabilité de notre propre modèle. Cette limitation pourrait être adressée par la conduite de test à plus grande échelle.

De plus, un test à plus grande échelle de la mesure de maturité est nécessaire pour valider les scores de maturité et développer ces indicateurs. Cela comprendra des tests pour la validité de construit et la validité de critère, pour s'assurer que le questionnaire mesure réellement le concept théorique qu'il est censé évaluer et se compare de manière appropriée à d'autres mesures externes, comme expliqué dans (TAHERDOOST 2016). Ce test à grande échelle sur un nombre significatif de projets permettra de consolider le modèle de maturité proposé. En outre, il serait intéressant d'évaluer l'impact de l'utilisation d'une telle approche sur le gain temps/argent lors de ce test.

3 PERSPECTIVES

Les travaux présentés dans cette thèse ouvrent plusieurs voies prometteuses pour la recherche future et l'application pratique dans le domaine des SdMT. La validation du SMAQ et la confirmation de sa fiabilité et de sa pertinence offrent un cadre robuste pour l'évaluation continue de la maturité technologique.

On peut déjà entrevoir des implications pratiques de cette méthode qui devront être confirmées par des tests et des interviews. Elle devrait permettre de :

- Prioriser les ajustements nécessaires sur les SEA jugés importants pour l'évolution du projet, avec une attention particulière aux artefacts de type ② qui nécessitent plus de raffinement.
- Fournir une mesure objective du progrès vers la conformité à mesure que les SEA de laboratoire sont raffinés et adaptés.
- Utiliser le score de maturité comme un outil de communication efficace avec les parties prenantes pour démontrer la préparation de la SdMT à passer à une étape suivante.

Cette approche semble pouvoir permettre de fluidifier les développements des SdMT du laboratoire vers l'industrie, sans pour autant lancer des activités qui n'ont pas lieu d'être en phase de recherche académique. Elle permettra également de gérer efficacement les ressources, en se concentrant sur les aspects les plus importants si la décision est prise de poursuivre vers le transfert, mais aussi dans le cas de développements spécifiques, pour permettre d'envisager des essais cliniques.

Les perspectives suivantes sont également envisageables, à plus longs termes :

- **Amélioration continue du SMAQ** : notre méthodologie, enrichie par des normes supplémentaires telles que l'utilisabilité ISO/IEC 62366 (ISO/IEC 2015) et la gestion des risques ISO/IEC 14971 (ISO/IEC 2007), offrirait une évaluation plus exhaustive de la maturité.
- **Intégration dans les processus réglementaires** : la précision et la fiabilité du SMAQ pourraient être mises à profit pour informer les processus réglementaires en utilisant le score dans le contexte post-transfert. Ceci pourrait aider

les organismes de régulation à mieux appréhender les technologies avant leur approbation. Mais aussi, cela pourrait apporter au fabricant un point de comparaison sur les aspects réglementaires. Cela faciliterait d'un côté comme de l'autre, la mise sur le marché de produits à la fois innovants et sûrs.

- **Formation et sensibilisation** : le développement de formations basées sur le modèle de maturité proposé peut contribuer à mieux préparer les ingénieurs et les professionnels de la santé à l'utilisation et à l'évaluation des technologies médicales. En donnant, une idée précise de la maturité, cela pourrait accroître la sensibilisation aux défis spécifiques de l'intégration technologique dans le domaine médical. Dans les équipes de recherche, ce score permet de, mieux prendre conscience du chemin à parcourir pour le transfert. Il pourrait permettre de former les personnels de recherche à l'importance de développer telles ou telles parties de la [SdMT](#).
- **Collaborations interdisciplinaires** : en fournissant un score et un modèle de maturité sur des technologies communes, les collaborations entre ingénieurs, chercheurs, cliniciens et régulateurs seraient facilitées. Ces interactions multidisciplinaires sont très importantes comme nous l'avons vu dans les travaux de BLEVINS et al. (2018) et MOREAU-GAUDRY et al. (2010). Elles permettent d'aligner les avancées technologiques sur les besoins cliniques et réglementaires.

4 CONCLUSION

Les modèles de maturité actuels, bien que linéaires dans leur approche, ne sont souvent pas adaptés aux contextes de recherche où la flexibilité est indispensable.

Dans ce travail, nous avons proposé l'ébauche d'une méthode pratique permettant une évaluation objective de la maturité technologique.

Le score de maturité, évaluable à tout moment du cycle de vie d'une [SdMT](#), permet une comparaison objective entre les projets, identifie les forces et faiblesses, et facilite l'élaboration d'une feuille de route vers le niveau de maturité ciblé. Cette capacité de suivi en temps réel offre une visibilité continue sur la maturation de la technologie, facilitant la prise de décision et la gestion du développement sans contraindre à la mise en œuvre d'un processus nouveau.

A

ANNEXES

A.1	Listings	143
A.2	Termes annotés	146
A.3	SEA	151
A.4	Calcul du score sur un projet réel	158

A.1 LISTINGS

```
1 %
2 %
3 % Environment: article
4 %
5 % Define an article of the norm
6 % Using \obeylines to get keep the text formatting as in the original text.
7 %
8 %
9 \newenvironment{article}[1]{%
10   \subsection{\texorpdfstring{#1}{\getCurrentSectionNumber}}%
11   \begingroup \obeylines %
12 }
13 {%
14 \medskip
15 \endgroup
16 \medskip
17 }
18 % same as article but for article that are at section level, not subsection
19 \newenvironment{articlesection}[1]{%
20   \section{\texorpdfstring{#1}{\getCurrentSectionNumber}}%
21   \begingroup \obeylines %
22 }
23 {%
24 \medskip
25 \endgroup
26 \medskip
27 }
28 % same as above but for article that are at subsection level, not
   subsection
```

```

29 \newenvironment{articlesubsubsection}[1]{%
30     \subsubsection{\texorpdfstring{#1}{\getCurrentSectionNumber}}%
31     \begingroup \obeylines %
32 }
33 {%
34 \medskip
35 \endgroup
36 \medskip
37 }

```

Listing A.1 – Définition d'un environnement article en L^AT_EX

```

1 %
2 -----
3 % Command: term
4 %
5 % Usage:
6 % \term{txt}[term]
7 % \term{term}
8 %
9 % add an entry term in index SETERMS and index SETERMSECTIONS.
10 % The first form (2 parameters) prints "txt" on the pdf output
11 % The second form (1 parameter) prints "term" on the pdf output
12 %
13 % term is case insensitive
14 %
15 -----
16 \usepackage{xparse}
17 \usepackage{mfirstuc}
18 \NewDocumentCommand{\term}{ m o }{%
19     \IfNoValueTF {#2} {%
20         \textbf{#1}\uppercase{\sindex[SETERMS]{#1}}\uppercase{\sindex
21             [SETERMSECTIONS]{#1}}
22     }{%
23         \textbf{#1}\uppercase{\sindex[SETERMS]{#2}}\uppercase{\sindex
24             [SETERMSECTIONS]{#2}}
25     }
26 }

```

Listing A.2 – Définition d'une commande term en L^AT_EX

Dans le listing [A.2](#), la commande `\term` est définie avec deux arguments :

Le premier argument (`#1`) est obligatoire et est utilisé pour afficher du texte en gras avec `\textbf{#1}`.

Le deuxième argument (`\(#2`) est optionnel (`o` dans la déclaration `\NewDocumentCommand{\term}{ m o }`). Si le deuxième argument est fourni, le terme est indexé avec ce deuxième argument, sinon le terme est indexé avec le premier argument.

L'utilisation de `\uppercase` indique que les termes indexés doivent être en majuscules dans les index.

Ainsi, dans la commande `\term{terme}{termeIndex}` le premier argument est le terme qui sera affiché en gras, et le deuxième argument (optionnel) est utilisé pour indexer ce terme, avec le premier argument utilisé comme index par défaut s'il n'est pas fourni.

A.2 TERMES ANNOTÉS

TABLE A.1 – 165 termes annotés et 537 citations au total

Termes	Citations	Termes	Citations	Termes	Citations
Acceptance Criteria	10	Problem Investigation	1	Software Maintenance	4
Acceptance Requirement	1	Problem Investigation Report	1	Software Maintenance Plan	2
Anomaly Documentation	2	Problem Report	10	Software Maintenance Requirement	2
Anomaly Risk Evaluation	2	Problem Report Analysis	3	Software Method	1
Capability Requirement	1	Problem Resolution	14	Software Requirement	29
Change	7	Problem Safety Evaluation	4	Software Risk Analysis	1
Change Control	7	Regression Test	5	Software Safety Class	1
Change Management	1	Regulatory Requirement	1	Software Safety Classification	3
Change Request	13	Release	20	Software Standard	1
Change Risk	1	Release Documentation	3	Software Tool	3
Change Risk Management	3	Release Procedure	1	Software Tool Control List	1
Change Safety Management	1	Release Repeatability	1	Software Tool Documentation	1
Change Traceability	1	Requirement Traceability	3	Software Unit	8
Change Verification	3	Risk	4	Software Unit Detailed Design	2

Suite sur la page suivante

Tableau des Termes annotés et nombre de citations totales (suite)

Termes	Citations	Termes	Citations	Termes	Citations
Configuration Control	1	Risk Analysis	2	Software Unit Implementa- tion	3
Configuration Documenta- tion	3	Risk Control	4	Software Unit Integration	3
Configuration Identification	1	Risk Control Measure	20	Software Unit Interface Detai- led Design	1
Configuration Item	3	Risk Control Measure Verifi- cation	5	Software Unit Test	1
Configuration Item Docu- mentation	3	Risk Control Requirement	1	Software Unit Test Evaluation	1
Configuration Item Identifica- tion	2	Risk Management	14	Software Unit Verification	6
Configuration Item Verifica- tion	2	Risk Management Plan	1	Software Verification	4
Configuration Item Version	2	Risk Traceability	2	Software Verification Plan	1
Configuration Management	6	Safety	1	Soup	10
Configuration Management Information	2	Security	1	Soup Configuration Item	4
Configuration Management Plan	1	Security Requirement	1	Soup Designator	2

Suite sur la page suivante

Tableau des Termes annotés et nombre de citations totales (suite)

Termes	Citations	Termes	Citations	Termes	Citations
Configuration Tool Documentation	2	Segregation	5	Soup Failure	1
Data Definition Requirement	1	Sequences Of Events	2	Soup Functional Requirement	2
Database Requirement	1	Sequences Of Events Potential Risk	1	Soup Hardware Requirement	2
Deliverable	4	Software Architecture	9	Soup Identification	1
Deliverable Verification	3	Software Architecture Design	1	Soup Integration	1
Documentation Configuration Item	1	Software Architecture Documentation	3	Soup Known Anomaly	1
Documentation Information	2	Software Architecture Verification	2	Soup Management	1
Documentation Plan	1	Software Defect	1	Soup Performance Requirement	2
Documentation Requirement	1	Software Design	2	Soup Risk Analysis	2
External Component	2	Software Design Validation	1	Soup Software Requirement	2
Feedback	1	Software Detailed Design	3	Soup Version	1
Feedback Criteria	1	Software Development Life Cycle	3	Standard Library	1
Feedback Documentation	4	Software Development Life Cycle Model	3	System Configuration	1

Suite sur la page suivante

Tableau des Termes annotés et nombre de citations totales (suite)

Termes	Citations	Termes	Citations	Termes	Citations
Feedback Evaluation	3	Software Development Plan	16	System Design	1
Feedback Report	2	Software Development Plan Update	1	System Development Plan	1
Functional Requirement	1	Software Development Validation	1	System Input Requirement	1
Hardware Failure	1	Software Failure	1	System Interface	1
Hardware Item Integration	2	Software Integration	5	System Interface Detailed Design	2
Installation Requirement	1	Software Item	24	System Interface Requirement	1
Integration Plan	5	Software Item Functionality	1	System Output Requirement	1
Integration Test	9	Software Item Integration	4	System Requirement	11
Integration Test Plan	1	Software Item Interface	3	Test	20
Integration Test Report	3	Software Item Potential Risk	2	Test Coverage	1
Integration Test Verification	1	Software Item Potential Risk Identification	1	Test Criteria	2
Modification	6	Software Item Regression Test	1	Test Documentation	3
Modification Analysis	1	Software Item Requirement	1	Test Performance	1
Modification Kit	1	Software Item Risk	3	Test Report	3

Suite sur la page suivante

Tableau des Termes annotés et nombre de citations totales (suite)

Termes	Citations	Termes	Citations	Termes	Citations
Potential Risk Identification	1	Software Item Risk Analysis	2	Usability Engineering Requirement	2
Problem Analysis	1	Software Item Safety Class	1	User Interface Integration	1
Problem Correction	1	Software Item Test	2	User Interface Requirement	1

A.3 SEA

TABLE A.2 – Liste des 156 SEA, organisés selon leur classe, dans l'ordre de leur poids relatif dans la norme avec la répartition de leurs citations dans les clauses de la norme.

SEA	Classes	Clauses	Citations
Software Item	① ₁	5.1.1, 5.1.4, 5.3.1, 5.3.2, 5.3.3, 5.3.4, 5.3.5, 5.3.6, 5.4.2, 5.5.3, 5.6.2, 5.6.4, 5.6.6, 6.2.2, 6.3.2, 7.2.2, 7.3.2, 8.2.2	24
Release	① ₂	5.8, 5.8.1, 5.8.5, 5.8.8, 6.1, 6.2.1.1, 6.2.1.2, 6.2.1.3, 6.2.3, 6.2.4, 6.2.5, 6.3.2, 9.1, 9.3	20
Test	① ₃	5.1.1, 5.1.5, 5.5.2, 5.6.3, 5.6.4, 5.7, 5.7.1, 5.7.2, 5.7.3, 5.7.4, 9.8	20
Acceptance Criteria	① ₄	5.1.6, 5.5.3, 5.5.4, 5.6.4, 5.7.1, 5.7.4	10
Library	① ₅	5.1.7, 5.3.3, 5.3.4, 5.3.6, 7.4.1, 7.4.2	10
Integration Test	① ₆	5.1.5, 5.5.2, 5.6.4, 5.6.5, 5.6.8, 5.7.1	9
Software Architecture	① ₇	5.3.1, 5.3.2, 5.3.6, 5.4.1, 5.4.4	9
Software Unit	① ₈	5.4.1, 5.4.2, 5.4.3, 5.5.3, 5.5.4	8
Change	① ₉	6.2.5, 6.3.1, 6.3.2, 7.4.1, 8.2.2, 9.8	7
Regression Test	① ₁₀	5.6.6, 5.7.3, 9.8	5
Software Item Risk	① ₁₁	7.1.3, 7.1.4, 7.2.1, 7.3.3	5
Deliverable	① ₁₂	5.1.1, 5.1.6	4
Library Configuration Item	① ₁₃	5.1.1, 8.1.1, 8.1.2	4
Risk	① ₁₄	5.8.3, 7.1.1, 7.1.3, 7.2.2	4
Configuration Item	① ₁₅	5.1.11, 8.2.1, 8.3	3
Software Development Life Cycle	① ₁₆	5.1.6, 5.1.8, 8.2.1	3
Software Item Interface	① ₁₇	5.3.2, 5.3.6	3
Software Tool	① ₁₈	5.1.1, 5.1.4, 5.1.10	3
Configuration Item Version	① ₁₉	8.1.1, 8.1.3	2

SEA	Classes	Clauses	Citations
External Component	① ₂₀	5.3.2 , 5.4.3	2
Software Item Test	① ₂₁	5.6.3 , 9.8	2
Test Criteria	① ₂₂	5.2.6	2
Change Risk	① ₂₃	7.4.1	1
Documentation Configuration Item	① ₂₄	8.1.1	1
Issue	① ₂₅	6.1	1
Issue Criteria	① ₂₆	6.1	1
Library Version	① ₂₇	7.1.3	1
Problem Repair	① ₂₈	9.2	1
Safety	① ₂₉	9.1	1
Security	① ₃₀	9.1	1
Software Development Plan Update	① ₃₁	5.1.2	1
Software Item Regression Test	① ₃₂	9.8	1
Software Tool Control List	① ₃₃	5.1.10	1
Software Unit Test	① ₃₄	5.5.2	1
Standard Library	① ₃₅	8.1.2	1
System Interface	① ₃₆	6.2.3	1
Test Coverage	① ₃₇	5.7.1	1
Software Requirement	② ₁	5.1.1 , 5.1.3 , 5.2 , 5.2.1 , 5.2.2 , 5.2.3 , 5.2.4 , 5.2.5 , 5.2.6 , 5.3.1 , 5.3.4 , 5.3.6 , 5.7.1 , 5.7.4 , 6.3.1 , 7.2.2 , 9.4	29
Risk Control Measure	② ₂	5.1.1 , 5.2.3 , 7.1.1 , 7.2 , 7.2.1 , 7.2.2 , 7.3.1 , 7.3.2 , 7.3.3 , 7.4.1 , 7.4.2	20
Risk Management	② ₃	5.1.7 , 5.7.3 , 6.1 , 6.3.1 , 7 , 7.1.4 , 7.1.5 , 7.2.1 , 7.3.2 , 7.4.3 , 9.2 , 9.5	17
Corrective Maintenance	② ₄	5.1.1 , 5.1.9 , 5.6.8 , 5.7.2 , 6.1 , 6.2.2 , 9 , 9.2 , 9.5 , 9.7	16
Software Development Plan	② ₅	5.1 , 5.1.1 , 5.1.2 , 5.1.3 , 5.1.4 , 5.1.5 , 5.1.6 , 5.1.7 , 5.1.8 , 5.1.9	16
Change Request	② ₆	6.2.3 , 6.2.4 , 6.2.5 , 8.2.1 , 8.2.2 , 9.2 , 9.4 , 9.7	13

SEA	Classes	Clauses	Citations
System Requirement	② ₇	5.1.1 , 5.1.3 , 5.2.1 , 5.2.5 , 5.2.6 , 5.3.6	11
Problem Report	② ₈	6.2.1.2 , 6.2.2 , 9.1 , 9.5 , 9.6 , 9.7	10
Change Control	② ₉	8.2 , 8.2.1 , 8.2.2 , 8.2.3 , 9.4	7
Change Management	② ₁₀	5.1.1 , 6.3 , 6.3.1 , 6.3.2 , 7.4.2	7
Configuration Management	② ₁₁	5.1.1 , 5.1.9 , 6.1 , 6.3.2 , 8 , 8.1.3	7
Software Unit Verification	② ₁₂	5.5 , 5.5.2 , 5.5.5	6
Fault Isolation	② ₁₃	5.3.5	5
Integration Plan	② ₁₄	5.1.5 , 5.5.2 , 5.6.1 , 5.6.2 , 5.6.3 , 5.6.4 , 5.6.5 , 5.6.8 , 5.7.1	5
Risk Control Measure Verification	② ₁₅	7.3 , 7.3.1 , 7.3.2 , 7.3.3	5
Software Integration	② ₁₆	5.6 , 5.6.2 , 5.6.6 , 5.6.8	5
Issue Documentation	② ₁₇	6.1 , 6.2.1 , 6.2.1.2	4
Problem Safety Evaluation	② ₁₈	6.2.1.3 , 9.2	4
Risk Control	② ₁₉	5.2.6 , 5.3.5 , 5.3.6	4
Software Item Integration	② ₂₀	5.1.5 , 5.6.2 , 5.6.6	4
Software Maintenance	② ₂₁	5.1.9 , 6 , 6.1 , 6.3.1	4
Software Verification	② ₂₂	5.6.2 , 5.8.1	4
Change Risk Management	② ₂₃	7.4 , 7.4.2	3
Change Verification	② ₂₄	8.2.3	3
Configuration Documentation	② ₂₅	8.1.3 , 8.3	3
Configuration Identification	② ₂₆	8.1 , 8.1.1	3
Configuration Item Documentation	② ₂₇	5.1.11 , 8.1.3	3
Deliverable Verification	② ₂₈	5.1.6	3
Integration Test Report	② ₂₉	5.6.7	3
Issue Evaluation	② ₃₀	6.2.1 , 6.2.1.2	3
Library Identifier	② ₃₁	8.1.2	3

SEA	Classes	Clauses	Citations
Problem Report Analysis	② ₃₂	9.6 , 9.7	3
Release Documentation	② ₃₃	5.8.4 , 5.8.5	3
Requirement Traceability	② ₃₄	5.1.1 , 5.2.6 , 5.7.4	3
Software Architecture Documenta- tion	② ₃₅	5.3.1 , 5.3.2 , 5.3.6	3
Software Detailed Design	② ₃₆	5.4 , 5.4.4	3
Software Development Life Cycle Model	② ₃₇	5.1.1	3
Software Safety Classification	② ₃₈	5.1.1 , 8.2.2	3
Software Unit Implementation	② ₃₉	5.5 , 5.5.1	3
Software Unit Integration	② ₄₀	5.6.1 , 5.6.2	3
Test Documentation	② ₄₁	5.6.3 , 9.8	3
Test Report	② ₄₂	5.7.5	3
Anomaly Documentation	② ₄₃	5.8.2 ,	2
Anomaly Risk Evaluation	② ₄₄	5.8.3 ,	2
Configuration Item Verification	② ₄₅	5.1.11	2
Configuration Management Infor- mation	② ₄₆	5.1.9	2
Configuration Tool Documentation	② ₄₇	5.8.5	2
Documentation Information	② ₄₈	5.1.8	2
Hardware Item Integration	② ₄₉	5.6.2	2
Issue Report	② ₅₀	6.2.1.1	2
Library Functional Requirement	② ₅₁	5.3.3	2
Library Hardware Requirement	② ₅₂	5.3.4	2
Library Performance Requirement	② ₅₃	5.3.3	2
Library Risk Analysis	② ₅₄	7.1.3	2

SEA	Classes	Clauses	Citations
Library Software Requirement	② ₅₅	5.3.4	2
Risk Analysis	② ₅₆	5.2.4	2
Risk Traceability	② ₅₇	7.3.3	2
Software Architecture Verification	② ₅₈	5.3.6	2
Software Design	② ₅₉	5.2.3 , 5.5.4	2
Software Item Risk Analysis	② ₆₀	7.1.1	2
Software Maintenance Plan	② ₆₁	6.1	2
Software Maintenance Requirement	② ₆₂	5.2.2	2
Software Unit Detailed Design	② ₆₃	5.4.2	2
System Interface Detailed Design	② ₆₄	5.4.3	2
Usability Engineering Requirement	② ₆₅	5.2.2	2
Acceptance Requirement	② ₆₆	5.2.2 ,	1
Capability Requirement	② ₆₇	5.2.2 ,	1
Change Analysis	② ₆₈	6.2	1
Change Safety Management	② ₆₉	7.4.1	1
Change Traceability	② ₇₀	8.2.4	1
Configuration Control	② ₇₁	5.1.9	1
Configuration Management Plan	② ₇₂	5.1.9	1
Data Definition Requirement	② ₇₃	5.2.2	1
Database Requirement	② ₇₄	5.2.2	1
Documentation Plan	② ₇₅	5.1.8	1
Documentation Requirement	② ₇₆	5.2.2	1
Functional Requirement	② ₇₇	5.2.2	1
Hardware Failure	② ₇₈	5.2.3	1

SEA	Classes	Clauses	Citations
Installation Requirement	② ₇₉	5.2.2	1
Integration Test Plan	② ₈₀	5.1.5	1
Integration Test Verification	② ₈₁	5.6.5	1
Library Anomaly	② ₈₂	7.1.3	1
Library Failure	② ₈₃	7.1.3	1
Library Integration	② ₈₄	5.1.5	1
Library Management	② ₈₅	6.1	1
Problem Analysis	② ₈₆	6.2	1
Regulatory Standard Requirement	② ₈₇	5.2.2	1
Release Procedure	② ₈₈	5.8.8	1
Release Repeatability	② ₈₉	5.8.8	1
Risk Control Requirement	② ₉₀	7.2.2	1
Risk Identification	② ₉₁	7.1.2	1
Risk Management Plan	② ₉₂	5.1.7	1
Security Requirement	② ₉₃	5.2.2	1
Software Architecture Design	② ₉₄	5.3	1
Software Defect	② ₉₅	5.2.3	1
Software Design Validation	② ₉₆	5.1.3	1
Software Development Validation	② ₉₇	5.1.3	1
Software Failure	② ₉₈	7.1.1	1
Software Item Functionality	② ₉₉	7.2.2	1
Software Item Requirement	② ₁₀₀	7.2.2	1
Software Item Risk Identification	② ₁₀₁	7.1.2	1
Software Item Safety Class	② ₁₀₂	7.2.2	1
Software Method	② ₁₀₃	5.1.4	1

SEA	Classes	Clauses	Citations
Software Risk Analysis	② ₁₀₄	7.1	1
Software Safety Class	② ₁₀₅	6.2.2	1
Software Standard	② ₁₀₆	5.1.4	1
Software Tool Documentation	② ₁₀₇	5.8.5	1
Software Unit Interface Detailed Design	② ₁₀₈	5.4.3	1
Software Unit Test Evaluation	② ₁₀₉	5.5.2	1
Software Verification Plan	② ₁₁₀	5.1.6	1
System Configuration	② ₁₁₁	8.3	1
System Design	② ₁₁₂	5.1.3	1
System Development Plan	② ₁₁₃	5.1.1	1
System Input Requirement	② ₁₁₄	5.2.2	1
System Interface Requirement	② ₁₁₅	5.2.2	1
System Output Requirement	② ₁₁₆	5.2.2	1
Test Performance	② ₁₁₇	5.2.6	1
User Interface Integration	② ₁₁₈	5.6.2	1
User Interface Requirement	② ₁₁₉	5.2.2	1

A.4 CALCUL DU SCORE SUR UN PROJET RÉEL

TABLE A.3 – Calcul du score de maturité pour le projet FollowKnee

Types	Id	SEA	Nb de citations	Poids	Dans le SdMT?	Score
①	1	Software Item	24	0,045	1	4,47 %
①	2	Release	20	0,037	0	0,00 %
①	3	Test	20	0,037	1	3,72 %
①	4	Acceptance Criteria	10	0,019	0	0,00 %
①	5	Library	10	0,019	1	1,86 %
①	6	Integration Test	9	0,017	1	1,68 %
①	7	Software Architecture	9	0,017	1	1,68 %
①	8	Software Unit	8	0,015	1	1,49 %
①	9	Change	7	0,013	1	1,30 %
①	10	Regression Test	5	0,009	0	0,00 %
①	11	Deliverable	4	0,007	1	0,74 %
①	12	Library Configuration Item	4	0,007	0	0,00 %
①	13	Risk	4	0,007	0	0,00 %
①	14	Configuration Item	3	0,006	0	0,00 %
①	15	Software Development Life Cycle	3	0,006	1	0,56 %
①	16	Software Item Interface	3	0,006	1	0,56 %
①	17	Software Tool	3	0,006	1	0,56 %

Suite sur la page suivante

FollowKnee : Score de maturité du projet (suite)

Types	Id	SEA	Nb de citations	Poids	Dans le SdMT?	Score
①	18	External Component	2	0,004	1	0,37 %
Score obtenu avec les SEA ① présents : 18,99 %						
②	1	Software Requirement	29	0,054	1	5,40 %
②	2	Corrective Maintenance	16	0,030	0	0,00 %
②	3	Software Development Plan	16	0,030	0	0,00 %
②	4	Change Request	13	0,024	1	2,42 %
②	5	System Requirement	11	0,020	1	2,05 %
②	6	Problem Report	10	0,019	0	0,00 %
②	7	Change Control	7	0,013	0	0,00 %
②	8	Change Management	7	0,013	0	0,00 %
②	9	Configuration Management	7	0,013	0	0,00 %
②	10	Software Unit Verification	6	0,011	1	1,12 %
②	11	Fault Isolation	5	0,009	0	0,00 %
②	12	Integration Plan	5	0,009	0	0,00 %
②	13	Software Integration	5	0,009	1	0,93 %
②	14	Issue Documentation	4	0,007	1	0,74 %
②	15	Problem Safety Evaluation	4	0,007	0	0,00 %
②	16	Software Item Integration	4	0,007	1	0,74 %
②	17	Software Maintenance	4	0,007	1	0,74 %

Suite sur la page suivante

FollowKnee : Score de maturité du projet (suite)

Types	Id	SEA	Nb de citations	Poids	Dans le SdMT?	Score
②	18	Software Verification	4	0,007	0	0,00 %
②	19	Change Risk Management	3	0,006	0	0,00 %
②	20	Change Verification	3	0,006	0	0,00 %
②	21	Configuration Documentation	3	0,006	0	0,00 %
②	22	Configuration Identification	3	0,006	0	0,00 %
②	23	Deliverable Verification	3	0,006	0	0,00 %
②	24	Integration Test Report	3	0,006	0	0,00 %
②	25	Issue Evaluation	3	0,006	0	0,00 %
②	26	Library Identifier	3	0,006	0	0,00 %
②	27	Requirement Traceability	3	0,006	1	0,56 %
②	28	Software Architecture Documentation	3	0,006	1	0,56 %
②	29	Software Detailed Design	3	0,006	0	0,00 %
②	30	Software Development Life Cycle Model	3	0,006	1	0,56 %
②	31	Software Safety Classification	3	0,006	0	0,00 %
②	32	Software Unit Implementation	3	0,006	1	0,56 %
②	33	Software Unit Integration	3	0,006	1	0,56 %
②	34	Test Documentation	3	0,006	0	0,00 %
②	35	Test Report	3	0,006	0	0,00 %
②	36	Anomaly Documentation	2	0,004	0	0,00 %
②	37	Configuration Tool Documentation	2	0,004	0	0,00 %

Suite sur la page suivante

FollowKnee : Score de maturité du projet (suite)

Types	Id	SEA	Nb de citations	Poids	Dans le SdMT?	Score
②	38	Documentation Information	2	0,004	0	0,00 %
②	39	Hardware Item Integration	2	0,004	1	0,37 %
②	40	Issue Report	2	0,004	0	0,00 %
②	41	Library Functional Requirement	2	0,004	0	0,00 %
②	42	Library Hardware Requirement	2	0,004	0	0,00 %
②	43	Library Performance Requirement	2	0,004	0	0,00 %
②	44	Library Risk Analysis	2	0,004	0	0,00 %
②	45	Library Software Requirement	2	0,004	0	0,00 %
②	46	Software Architecture Verification	2	0,004	0	0,00 %
Score obtenu avec les SEA ② présents : 17,32 %						
Score de maturité de la SdMT "Planning" du projet FollowKnee : 36,31 %						

TABLE A.4 – Réponses à la présence ou l'absence des SEA de l'ingénieur 1

SEA	Réponse	SEA	Réponse	SEA	Réponse
Acceptance Criteria	No	Integration Plan	No	Risk	No
Anomaly Documentation	No	Integration Test	Yes	Software Architecture	No
Change	Yes	Integration Test Report	No	Software Detailed Design	No
Change Control	No	Issue Documentation	Yes	Software Development Life Cycle	No
Change Management	No	Issue Evaluation	No	Software Development Plan	No
Change Request	Yes	Issue Report	No	Software Integration	Yes
Change Risk Management	No	Library	Yes	Software Item	Yes
Change Verification	No	Library Configuration Item	No	Software Item Integration	Yes
Configuration Documenta- tion	No	Library Functional Require- ment	No	Software Item Interface	No
Configuration Identification	No	Library Hardware Require- ment	No	Software Maintenance	Yes
Configuration Item	No	Library Identifier	No	Software Requirement	Yes
Configuration Management	No	Library Performance Require- ment	No	Software Safety Classification	No
Configuration Tool Documen- tation	No	Library Risk Analysis	No	Software Tool	Yes
Corrective Maintenance	No	Library Software Require- ment	No	Software Unit	No
Deliverable	No	Problem Report	No	Software Verification	No

Suite sur la page suivante

Réponses à la présence ou l'absence des SEA de l'ingénieur 1 (suite)

SEA	Réponse	SEA	Réponse	SEA	Réponse
Documentation Information	No	Problem Safety Evaluation	No	System Requirement	Yes
External Component	No	Regression Test	No	Test	Yes
Fault Isolation	No	Release	No	Test Documentation	No
Hardware Item Integration	Yes	Requirement Traceability	Yes	Test Report	No

TABLE A.5 – Calcul du score de maturité pour le projet FollowKnee selon les réponses de l'ingénieur 1.

Types	Id	SEA	Nb de citations	Poids	Dans le SdMT?	Score
①	1	Software Item	24	0,045	1	4,47 %
①	2	Release	20	0,037	0	0,00 %
①	3	Test	20	0,037	1	3,72 %
①	4	Acceptance Criteria	10	0,019	0	0,00 %
①	5	Library	10	0,019	1	1,86 %
①	6	Integration Test	9	0,017	1	1,68 %
①	7	Software Architecture	9	0,017	1	1,68 %
①	8	Software Unit	8	0,015	1	1,49 %
①	9	Change	7	0,013	1	1,30 %
①	10	Regression Test	5	0,009	0	0,00 %
①	11	Deliverable	4	0,007	1	0,74 %
①	12	Library Configuration Item	4	0,007	0	0,00 %
①	13	Risk	4	0,007	0	0,00 %
①	14	Configuration Item	3	0,006	0	0,00 %
①	15	Software Development Life Cycle	3	0,006	1	0,56 %
①	16	Software Item Interface	3	0,006	1	0,56 %
①	17	Software Tool	3	0,006	1	0,56 %
①	18	External Component	2	0,004	1	0,37 %
Score obtenu avec les SEA ① présents : 18,99 %						

Suite sur la page suivante

FollowKnee : Score de maturité selon les réponses de l'ingénieur 1.(suite)

Types	Id	SEA	Nb de citations	Poids	Dans le SdMT?	Score
②	1	Software Requirement	29	0,054	1	5,40 %
②	2	Corrective Maintenance	16	0,030	0	0,00 %
②	3	Software Development Plan	16	0,030	0	0,00 %
②	4	Change Request	13	0,024	1	2,42 %
②	5	System Requirement	11	0,020	1	2,05 %
②	6	Problem Report	10	0,019	0	0,00 %
②	7	Change Control	7	0,013	0	0,00 %
②	8	Change Management	7	0,013	0	0,00 %
②	9	Configuration Management	7	0,013	0	0,00 %
②	10	Software Unit Verification	6	0,011	1	1,12 %
②	11	Fault Isolation	5	0,009	0	0,00 %
②	12	Integration Plan	5	0,009	0	0,00 %
②	13	Software Integration	5	0,009	1	0,93 %
②	14	Issue Documentation	4	0,007	1	0,74 %
②	15	Problem Safety Evaluation	4	0,007	0	0,00 %
②	16	Software Item Integration	4	0,007	1	0,74 %
②	17	Software Maintenance	4	0,007	1	0,74 %
②	19	Change Risk Management	3	0,006	0	0,00 %
②	20	Change Verification	3	0,006	0	0,00 %
②	21	Configuration Documentation	3	0,006	0	0,00 %

Suite sur la page suivante

FollowKnee : Score de maturité selon les réponses de l'ingénieur 1.(suite)

Types	Id	SEA	Nb de citations	Poids	Dans le SdMT?	Score
②	22	Configuration Identification	3	0,006	0	0,00 %
②	24	Integration Test Report	3	0,006	0	0,00 %
②	25	Issue Evaluation	3	0,006	0	0,00 %
②	26	Library Identifier	3	0,006	0	0,00 %
②	27	Requirement Traceability	3	0,006	1	0,56 %
②	29	Software Detailed Design	3	0,006	0	0,00 %
②	31	Software Safety Classification	3	0,006	0	0,00 %
②	34	Test Documentation	3	0,006	0	0,00 %
②	35	Test Report	3	0,006	0	0,00 %
②	36	Anomaly Documentation	2	0,004	0	0,00 %
②	37	Configuration Tool Documentation	2	0,004	0	0,00 %
②	38	Documentation Information	2	0,004	0	0,00 %
②	39	Hardware Item Integration	2	0,004	1	0,37 %
②	40	Issue Report	2	0,004	0	0,00 %
②	41	Library Functional Requirement	2	0,004	0	0,00 %
②	42	Library Hardware Requirement	2	0,004	0	0,00 %
②	43	Library Performance Requirement	2	0,004	0	0,00 %
②	44	Library Risk Analysis	2	0,004	0	0,00 %
②	45	Library Software Requirement	2	0,004	0	0,00 %

Suite sur la page suivante

FollowKnee : Score de maturité selon les réponses de l'ingénieur 1.(suite)

Types	Id	SEA	Nb de citations	Poids	Dans le SdMT?	Score
Score obtenu avec les SEA ② présents : 15,08 %						
Score de maturité de la SdMT "Planning" du projet FollowKnee : 34,07 %						

TABLE A.6 – Réponses à la présence ou l'absence des SEA de l'ingénieur 2

SEA	Réponse	SEA	Réponse	SEA	Réponse
Acceptance Criteria	No	Issue Report	No	Software Development Life Cycle Model	Yes
Anomaly Documentation	No	Library	Yes	Software Development Plan	No
Change	No	Library Configuration Item	No	Software Integration	Yes
Configuration Documentation	No	Library Functional Requirement	No	Software Item	Yes
Configuration Identification	No	Library Hardware Requirement	No	Software Item Integration	Yes
Configuration Item	No	Library Identifier	No	Software Item Interface	Yes
Configuration Management	No	Library Performance Requirement	No	Software Maintenance	No
Configuration Tool Documentation	No	Library Risk Analysis	No	Software Requirement	Yes
Corrective Maintenance	No	Library Software Requirement	No	Software Safety Classification	No
Deliverable	Yes	Problem Report	No	Software Tool	Yes
Deliverable Verification	No	Problem Safety Evaluation	No	Software Unit	Yes
Documentation Information	No	Regression Test	No	Software Unit Implementation	Yes
External Component	Yes	Release	No	Software Unit Integration	Yes
Fault Isolation	No	Requirement Traceability	No	Software Unit Verification	Yes

Suite sur la page suivante

Réponses à la présence ou l'absence des SEA de l'ingénieur 2(suite)

SEA	Réponse	SEA	Réponse	SEA	Réponse
Hardware Item Integration	No	Risk	No	Software Verification	No
Integration Plan	No	Software Architecture	Yes	System Requirement	Yes
Integration Test	Yes	Software Architecture Documentation	Yes	Test	Yes
Integration Test Report	No	Software Architecture Verification	No	Test Documentation	No
Issue Documentation	Yes	Software Detailed Design	No	Test Report	No
Issue Evaluation	No	Software Development Life Cycle	Yes		

TABLE A.7 – Calcul du score de maturité pour le projet FollowKnee selon les réponses de l'ingénieur 2.

Types	Id	SEA	Nb de citations	Poids	Dans le SdMT?	Score
①	1	Software Item	24	0,045	1	4,47 %
①	2	Release	20	0,037	0	0,00 %
①	3	Test	20	0,037	1	3,72 %
①	4	Acceptance Criteria	10	0,019	0	0,00 %
①	5	Library	10	0,019	1	1,86 %
①	6	Integration Test	9	0,017	1	1,68 %
①	7	Software Architecture	9	0,017	1	1,68 %
①	8	Software Unit	8	0,015	1	1,49 %
①	9	Change	7	0,013	0	0,00 %
①	10	Regression Test	5	0,009	0	0,00 %
①	11	Deliverable	4	0,007	1	0,74 %
①	12	Library Configuration Item	4	0,007	0	0,00 %
①	13	Risk	4	0,007	0	0,00 %
①	14	Configuration Item	3	0,006	0	0,00 %
①	15	Software Development Life Cycle	3	0,006	1	0,56 %
①	16	Software Item Interface	3	0,006	1	0,56 %
①	17	Software Tool	3	0,006	1	0,56 %
①	18	External Component	2	0,004	1	0,37 %
Score obtenu avec les SEA ① présents : 17,69 %						

Suite sur la page suivante

FollowKnee : Score de maturité selon les réponses de l'ingénieur 2 (suite)

Types	Id	SEA	Nb de citations	Poids	Dans le SdMT?	Score
②	1	Software Requirement	29	0,054	1	5,40 %
②	2	Corrective Maintenance	16	0,030	0	0,00 %
②	3	Software Development Plan	16	0,030	0	0,00 %
②	5	System Requirement	11	0,020	1	2,05 %
②	6	Problem Report	10	0,019	0	0,00 %
②	9	Configuration Management	7	0,013	0	0,00 %
②	10	Software Unit Verification	6	0,011	1	1,12 %
②	11	Fault Isolation	5	0,009	0	0,00 %
②	12	Integration Plan	5	0,009	0	0,00 %
②	13	Software Integration	5	0,009	1	0,93 %
②	14	Issue Documentation	4	0,007	1	0,74 %
②	15	Problem Safety Evaluation	4	0,007	0	0,00 %
②	16	Software Item Integration	4	0,007	1	0,74 %
②	17	Software Maintenance	4	0,007	0	0,00 %
②	18	Software Verification	4	0,007	0	0,00 %
②	21	Configuration Documentation	3	0,006	0	0,00 %
②	22	Configuration Identification	3	0,006	0	0,00 %
②	23	Deliverable Verification	3	0,006	0	0,00 %
②	24	Integration Test Report	3	0,006	0	0,00 %
②	25	Issue Evaluation	3	0,006	0	0,00 %

Suite sur la page suivante

FollowKnee : Score de maturité selon les réponses de l'ingénieur 2 (suite)

Types	Id	SEA	Nb de citations	Poids	Dans le SdMT?	Score
②	26	Library Identifier	3	0,006	0	0,00 %
②	27	Requirement Traceability	3	0,006	0	0,00 %
②	28	Software Architecture Documentation	3	0,006	1	0,56 %
②	29	Software Detailed Design	3	0,006	0	0,00 %
②	30	Software Development Life Cycle Model	3	0,006	1	0,56 %
②	31	Software Safety Classification	3	0,006	0	0,00 %
②	32	Software Unit Implementation	3	0,006	1	0,56 %
②	33	Software Unit Integration	3	0,006	1	0,56 %
②	34	Test Documentation	3	0,006	0	0,00 %
②	35	Test Report	3	0,006	0	0,00 %
②	36	Anomaly Documentation	2	0,004	0	0,00 %
②	37	Configuration Tool Documentation	2	0,004	0	0,00 %
②	38	Documentation Information	2	0,004	0	0,00 %
②	39	Hardware Item Integration	2	0,004	0	0,00 %
②	40	Issue Report	2	0,004	0	0,00 %
②	41	Library Functional Requirement	2	0,004	0	0,00 %
②	42	Library Hardware Requirement	2	0,004	0	0,00 %
②	43	Library Performance Requirement	2	0,004	0	0,00 %
②	44	Library Risk Analysis	2	0,004	0	0,00 %
②	45	Library Software Requirement	2	0,004	0	0,00 %

Suite sur la page suivante

FollowKnee : Score de maturité selon les réponses de l'ingénieur 2 (suite)

Types	Id	SEA	Nb de citations	Poids	Dans le SdMT?	Score
②	46	Software Architecture Verification	2	0,004	0	0,00 %
Score obtenu avec les SEA ② présents : 13,22 %						
Score de maturité de la SdMT "Planning" du projet FollowKnee : 30,91 %						

BIBLIOGRAPHIE

- BAGARIA, Vaibhav et Kshitij CHAUDHARY (2017). « A paradigm shift in surgical planning and simulation using 3Dgraphy : Experience of first 50 surgeries done using 3D-printed biomodels ». In : *Injury* 48.11, p. 2501-2508. ISSN : 0020-1383. DOI : [10.1016/j.injury.2017.08.058](https://doi.org/10.1016/j.injury.2017.08.058). URL : <https://www.sciencedirect.com/science/article/pii/S0020138317305776>.
- BLEVINS, Katherine S., Dan E. AZAGURY, James K. WALL, Venita CHANDRA, Elisabeth K. WYNNE et Thomas M. KRUMMEL (2018). « Chapter 12 - How Good Ideas Die : Understanding Common Pitfalls of Medtech Innovation ». In : *Medical Innovation*. Sous la dir. de Kevin E. BEHRNS, Bruce GINGLES et Michael G. SARR. Academic Press, p. 117-127. ISBN : 978-0-12-814926-3. DOI : <https://doi.org/10.1016/B978-0-12-814926-3.00012-7>. URL : <https://www.sciencedirect.com/science/article/pii/B9780128149263000127>.
- BROMMELAND, T., R. KLOSTER et T. INGEBRIGTSEN (2002). « A four-year experience with a stereotactic computer in a small neurosurgical department ». In : *Surgical Neurology* 57.3, p. 190-194. ISSN : 0090-3019. DOI : [10.1016/S0090-3019\(01\)00700-5](https://doi.org/10.1016/S0090-3019(01)00700-5). URL : <https://www.sciencedirect.com/science/article/pii/S0090301901007005>.
- BUCHNER, Georg A., J. STEPPUTAT KAI, W. ZIMMERMANN ARNO et Reinhard SCHOMÄCKER (mai 2019). « Specifying Technology Readiness Levels for the Chemical Industry ». In : *Ind. Eng. Chem. Res.* 58.17, p. 6957-6969. ISSN : 0888-5885. DOI : [10.1021/acs.iecr.8b05693](https://doi.org/10.1021/acs.iecr.8b05693).
- CIMIT (2018). *NAVIGATING THE HEALTHTECH INNOVATION CYCLE*. URL : <https://cimit.org/documents/173804/228699/Navigating%20the%20HealthTech%20Innovation%20Cycle.pdf/2257c90b-d90b-3b78-6dc9-745db401fbc6>.
- CINQUIN, P. et al. (1995). « Computer assisted medical interventions ». In : *IEEE Engineering in Medicine and Biology Magazine* 14.3, p. 254-263. ISSN : 1937-4186. DOI : [10.1109/51.391779](https://doi.org/10.1109/51.391779).
- COOK, David A. et Thomas J. BECKMAN (2006). « Current Concepts in Validity and Reliability for Psychometric Instruments : Theory and Application ». In : *The American Journal of Medicine* 119.2, 166.e7-166.e16. ISSN : 0002-9343. DOI : [10.1016/j.amjmed.2005.10.036](https://doi.org/10.1016/j.amjmed.2005.10.036). URL : <https://www.sciencedirect.com/science/article/pii/S0002934305010375>.
- CTSI OF THE UNIVERSITY OF MINNESOTA (2023). *What is Translational Research?* En. Clinical and Translational Science Institute. Clinical et Translational Science Institute. URL : <https://ctsi.umn.edu/about/about-ctsi/what-translational-research>.
- DEEP, Kamal, Shivakumar SHANKAR et Ashish MAHENDRA (2017). « Computer assisted navigation in total knee and hip arthroplasty. » eng. In : *SICOT-J* 3, p. 50.
- DERATHÉ, Arthur, Fabian RECHE, Alexandre MOREAU-GAUDRY, Pierre JANNIN, Bernard GIBAUD et Sandrine VOROS (2020). « Predicting the quality of surgical exposure using spatial and procedural features from laparoscopic videos ». In : *International*

- Journal of Computer Assisted Radiology and Surgery* 15.1, p. 59-67. ISSN : 1861-6429. DOI : [10.1007/s11548-019-02072-3](https://doi.org/10.1007/s11548-019-02072-3). URL : <https://doi.org/10.1007/s11548-019-02072-3>.
- DUPUY, T., C. BEITONE, J. TROCCAZ et S. VOROS (2023). « 2D/3D Deep Registration Along Trajectories With Spatiotemporal Context : Application To Prostate Biopsy Navigation ». In : *IEEE Transactions on Biomedical Engineering*, p. 1-12. ISSN : 1558-2531. DOI : [10.1109/TBME.2023.3243436](https://doi.org/10.1109/TBME.2023.3243436).
- ENGEL, Rainer ME (2003). « Philipp Bozzini—the father of endoscopy ». In : *Journal of Endourology* 17.10, p. 859-862.
- EUR-LEX (2017). *European Medical Device Regulations 2017/745*. EN. URL : <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32017R0745>.
- FDA (2018). *SaMD*. URL : <https://www.fda.gov/medical-devices/digital-health-center-excellence/software-medical-device-samd>.
- FERNANDEZ-MOURE, Joseph S. (2016). « Lost in Translation : The Gap in Scientific Advancements and Clinical Application ». In : *Frontiers in Bioengineering and Biotechnology* 4. ISSN : 2296-4185. DOI : [10.3389/fbioe.2016.00043](https://doi.org/10.3389/fbioe.2016.00043). URL : <https://www.frontiersin.org/articles/10.3389/fbioe.2016.00043>.
- FIARD, Gaelle, Sonia-Yuki SELMI, Emmanuel PROMAYON, Jean-Luc DESCOTES et Jocelyne TROCCAZ (déc. 2020). « Simulation-based training for prostate biopsies : towards the validation of the Biopsym simulator ». In : *Minimally Invasive Therapy & Allied Technologies* 29.6, p. 359-365. ISSN : 1364-5706. DOI : [10.1080/13645706.2019.1653926](https://doi.org/10.1080/13645706.2019.1653926). URL : <https://doi.org/10.1080/13645706.2019.1653926>.
- FICHTINGER, Gabor, Jocelyne TROCCAZ et Tamas HAIDEGGER (2022). « Image-Guided Interventional Robotics : Lost in Translation? » In : *Proceedings of the IEEE* 110.7, p. 932-950. DOI : [10.1109/JPROC.2022.3166253](https://doi.org/10.1109/JPROC.2022.3166253).
- FONTANAROSA, Phil B. et Catherine D. DEANGELIS (avr. 2002). « Basic Science and Translational Research in JAMA ». In : *JAMA* 287.13, p. 1728-1728. ISSN : 0098-7484. DOI : [10.1001/jama.287.13.1728](https://doi.org/10.1001/jama.287.13.1728). eprint : <https://jamanetwork.com/journals/jama/articlepdf/194783/jed20018.pdf>. URL : <https://doi.org/10.1001/jama.287.13.1728>.
- GEBHARD, Florian T., Michael D. KRAUS, Eugen SCHNEIDER, Ulrich C. LIENER, Lothar KINZL et Markus ARAND (2006). « Does Computer-Assisted Spine Surgery Reduce Intraoperative Radiation Doses? » In : *Spine* 31.17. ISSN : 0362-2436. URL : https://journals.lww.com/spinejournal/fulltext/2006/0810/does_computer_assisted_spine_surgery_reduce.22.aspx.
- GONÇALVES, Jesús M., M. JOSÉ SÁNCHEZ-LEDESMA, Pablo RUISOTO, Mauricio JARAMILLO, J. Javier JIMÉNEZ et Juan A. JUANES (2018). « Neuronavigation Software to visualize and surgically approach brain structures ». In : *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality. TEEM'18*. Salamanca, Spain : Association for Computing Machinery, p. 405-409. ISBN : 9781450365185. DOI : [10.1145/3284179.3284248](https://doi.org/10.1145/3284179.3284248). URL : <https://doi.org/10.1145/3284179.3284248>.
- GORDON, William J. et Ariel D. STERN (2019). « Challenges and opportunities in software-driven medical devices ». In : *Nature biomedical engineering* 3.7, p. 493. DOI : [10.1038/s41551-019-0426-z](https://doi.org/10.1038/s41551-019-0426-z).
- GWET, Kilem L (2014). *Handbook of inter-rater reliability : The definitive guide to measuring the extent of agreement among raters*. Advanced Analytics, LLC.

- HABERBUSCH, Jean-Loup, Guillaume DARDENNE et Emmanuel PROMAYON (2023). « Maturity assessment of software-driven medical technologies : a quantitative score derived from a quality standard for the research phase * ». In : *2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, p. 1-4. DOI : [10.1109/EMBC40787.2023.10341142](https://doi.org/10.1109/EMBC40787.2023.10341142).
- HAGEDORN, Thomas J., Ian R. GROSSE et Sundar KRISHNAMURTY (2015). « A concept ideation framework for medical device design ». In : *Journal of Biomedical Informatics* 55, p. 218-230. ISSN : 1532-0464. DOI : <https://doi.org/10.1016/j.jbi.2015.04.010>.
- HARTL, Dominik et al. (2021). « Translational precision medicine : an industry perspective ». In : *Journal of Translational Medicine* 19.1, p. 245. ISSN : 1479-5876. DOI : [10.1186/s12967-021-02910-6](https://doi.org/10.1186/s12967-021-02910-6). URL : <https://doi.org/10.1186/s12967-021-02910-6>.
- HAUSCHILD, Anne-Christin, Roman MARTIN, Sabrina Celine HOLST, Joachim WIENBECK et Dominik HEIDER (2022). « Guideline for software life cycle in health informatics ». In : *iScience* 25.12, p. 105534. ISSN : 2589-0042. DOI : <https://doi.org/10.1016/j.isci.2022.105534>. URL : <https://www.sciencedirect.com/science/article/pii/S2589004222018065>.
- HAUTE AUTORITÉ DE SANTÉ (juill. 2021). *Parcours du dispositif médical en France*. Guide disponible sur le site de la Haute Autorité de Santé. Mis à jour. URL : https://www.has-sante.fr/upload/docs/application/pdf/2009-12/guide_pratique_dm.pdf.
- HÉDER, Mihály (2017). « From NASA to EU : the evolution of the TRL scale in Public Sector Innovation ». In : *THE INNOVATION JOURNAL* 22.2, p. 1-23.
- HENDERSON-SELLERS, Brian, Tom McBRIDE, Graham LOW et Cesar GONZALEZ-PEREZ (2013). « Ontologies for International Standards for Software Engineering ». In : *Conceptual Modeling*. Sous la dir. de Wilfred Ng, Veda C. STOREY et Juan C. TRUJILLO. Berlin, Heidelberg : Springer Berlin Heidelberg, p. 479-486. ISBN : 978-3-642-41924-9.
- HERMON, Rebecca, Patricia A. H. WILLIAMS et Vincent B. McCAULEY (2021). « Software as a Medical Device (SaMD) : Useful or Useless Term ? ». In : *54th Hawaii International Conference on System Sciences*, p. 3722-3731.
- IBRAHIM, Mohd Salami, Myat Moe Thwe AUNG, San San Oo, Mohd Ihsanuddin ABAS et Nur Azree Ferdaus KAMUDIN (2022). « Content validation template (CVT) ». In : *Journal of Statistical Computation and Simulation* 0.0, p. 1-18. DOI : [10.1080/00949655.2022.2128799](https://doi.org/10.1080/00949655.2022.2128799). eprint : <https://doi.org/10.1080/00949655.2022.2128799>. URL : <https://doi.org/10.1080/00949655.2022.2128799>.
- IMDRF (2013). *Software as a Medical Device (SaMD) : Key Definitions*. International Medical Device Regulators Forum.
- (déc. 2016). *Software as a Medical Device (SaMD) : Clinical Evaluation*. EN. URL : <https://www.imdrf.org/consultations/software-medical-device-samd-clinical-evaluation>.
- INTUITIV (2023). *Intuitiv Da Vinci*. URL : <https://www.intuitive.com/en-us/products-and-services/da-vinci>.
- ISIR, ICUBE, LIRMM (2016). *Surgical Cockpit*. EN. URL : <https://cami-labex.fr/surgical-cockpit/>.

- ISO/IEC (2006). *62304 and updates Medical device software Software life cycle processes*. International Organization for Standardization.
- (2007). *14971 :2007 Medical devices — Application of risk management to medical devices*. International Organization for Standardization.
 - (2015). *IEC 62366-1 :2015 Medical devices — Part 1 : Application of usability engineering to medical devices*. International Organization for Standardization.
- KELLY, Patrick J., Bruce A. KALL, Stephan GOERSS et Franklin EARNEST (1986). « Computer-assisted stereotaxic laser resection of intra-axial brain neoplasms ». In : *Journal of Neurosurgery* 64.3, p. 427-439. DOI : [10.3171/jns.1986.64.3.0427](https://doi.org/10.3171/jns.1986.64.3.0427). URL : <https://thejns.org/view/journals/j-neurosurg/64/3/article-p427.xml>.
- KER, Justin, Lipo WANG, Jai RAO et Tchoyoson LIM (2018). « Deep Learning Applications in Medical Image Analysis ». In : *IEEE Access* 6, p. 9375-9389. DOI : [10.1109/ACCESS.2017.2788044](https://doi.org/10.1109/ACCESS.2017.2788044).
- KIM, DongYeop, Ye-Seul PARK, Byungjeong LEE et Jung-Won LEE (2019). « Ontology-based process integration incorporating reference associations between medical standards from the perspective of medical software developers ». In : *Journal of Ambient Intelligence and Humanized Computing*. ISSN : 1868-5145. DOI : [10.1007/s12652-019-01383-2](https://doi.org/10.1007/s12652-019-01383-2). URL : <https://doi.org/10.1007/s12652-019-01383-2>.
- KIM, dongyeop, Byungjeong LEE et Jung-Won LEE (août 2019). « Building a Rule-Based Goal-Model from the IEC 62304 Standard for Medical Device Software ». In : *KSII Transactions on Internet and Information Systems* 13.8, p. 4174. ISSN : 1976-7277. DOI : [10.3837/tiis.2019.08.020](https://doi.org/10.3837/tiis.2019.08.020).
- LABEx CAMI (2015). *LabEx CAMI*. En. TIMC, ISIR, LaTIM, ICube, LTSI, LIRMM. URL : <https://cami-labex.fr/>.
- LANDIS, J. Richard et Gary G. KOCH (1977). « The Measurement of Observer Agreement for Categorical Data ». In : *Biometrics* 33.1, p. 159-174. ISSN : 0006341X, 15410420. DOI : [10.2307/2529310](https://doi.org/10.2307/2529310). URL : <http://www.jstor.org/stable/2529310>.
- LAPOUGE, Guillaume, Philippe POIGNET et Jocelyne TROCCAZ (2021). « Towards 3D Ultrasound Guided Needle Steering Robust to Uncertainties, Noise, and Tissue Heterogeneity ». In : *IEEE Transactions on Biomedical Engineering* 68.4, p. 1166-1177. DOI : [10.1109/TBME.2020.3022619](https://doi.org/10.1109/TBME.2020.3022619).
- LASRADO, Lester Allan (2018). « Set-Theoretic Approach to Maturity Models ». eng. hdl :10398/9621. Thèse de doct. Frederiksberg. ISBN : 9788793579774. URL : <http://hdl.handle.net/10419/209064>.
- LASRADO, Lester Allan, Ravi VATRAPU et Kim Normann ANDERSEN (2015). « Maturity models development in is research : a literature review ». In : *IRIS Selected Papers of the Information Systems Research Seminar in Scandinavia*. T. 6. IRIS New York, NY, USA.
- LATIM, LTSI, TIMC, LIRM, ISIR (sept. 2013). *CAMMA*. URL : <https://cami-labex.fr/camma/>.
- LAWSHE, Charles H. (1975). « A quantitative approach to content validity ». In : *Personnel psychology* 28.4, p. 563-575.
- LIRMM, TIMC, ICUBE (2016). *FLEXTER*. URL : <https://cami-labex.fr/flexer/>.
- LTSI, ISIR, TIMC (2017). *SIMPLE*. En. URL : <https://cami-labex.fr/simple/>.

- LYNN, Mary R. (1986). « Determination and Quantification Of Content Validity ». In : *Nursing Research* 35.6. ISSN : 0029-6562.
- MANKINS, John C. (1995). « Technology Readiness Levels : A White Paper/6 April 1995 NASA Office of Space Access and Technology, Advanced Concepts Office ». In : 6, p. 1995. URL : https://aiaa.kavi.com/apps/group_public/download.php/2212/TRLs_MankinsPaper_1995.pdf.
- MAYRINK, Nadja N. V., LUÍS ALCOFORADO, Arthur CHIORO, Felipe FERNANDES, Thaisa S. LIMA, Erika B. CAMARGO et Ricardo A. M. VALENTIM (2022). « Translational research in health technologies : A scoping review ». In : *Frontiers in Digital Health* 4. ISSN : 2673-253X. DOI : 10.3389/fdgth.2022.957367. URL : <https://www.frontiersin.org/articles/10.3389/fdgth.2022.957367>.
- MC CAFFERY, Fergal, John BURTON et Ita RICHARDSON (sept. 2009). « Risk management capability model for the development of medical device software ». In : *Software Quality Journal* 18.1, p. 81. ISSN : 1573-1367. DOI : 10.1007/s11219-009-9086-7.
- MCCULLOCH, Peter, Douglas ALTMAN G., Bruce CAMPBELL W., David FLUM R., Paul GLASZIOU, John MARSHALL C. et Jon NICHOLL (2009). « No surgical innovation without evaluation : the IDEAL recommendations ». In : *The Lancet* 374.9695, p. 1105-1112. ISSN : 0140-6736. DOI : [https://doi.org/10.1016/S0140-6736\(09\)61116-8](https://doi.org/10.1016/S0140-6736(09)61116-8). URL : <https://www.sciencedirect.com/science/article/pii/S0140673609611168>.
- MÉNDEZ FERNÁNDEZ, Daniel, Wolfgang BÖHM, Andreas VOGELANG, Jakob MUND, Manfred BROY, Marco KUHRMANN et Thorsten WEYER (2019). « Artefacts in software engineering : a fundamental positioning ». In : *Software & Systems Modeling* 18.5, p. 2777-2786. ISSN : 1619-1374. DOI : 10.1007/s10270-019-00714-3. URL : <https://doi.org/10.1007/s10270-019-00714-3>.
- MOLITERNO, Jennifer A., Toral R. PATEL et Joseph M. PIEPMEIER (2012). « Neurosurgical Approach ». In : *The Cancer Journal* 18.1. ISSN : 1540-336X. URL : https://journals.lww.com/journalppo/fulltext/2012/01000/neurosurgical_approach.5.aspx.
- MORAN, Michael E. (juill. 2007). « Jacques de Vaucanson : The Father of Simulation ». In : *Journal of Endourology* 21.7, p. 679-683. ISSN : 0892-7790. DOI : 10.1089/end.2007.9951. URL : <https://doi.org/10.1089/end.2007.9951>.
- MOREAU-GAUDRY, A. et L. PAZART (2010). « Développement d'une innovation technologique en santé : le cycle CREPS Concept-Recherche-Essais-Produit-Soins ». In : *Irbm* 31.1, p. 12-21.
- OLUWATAYO, James Ayodele (mai 2012). « Validity and Reliability Issues in Educational Research ». In : *Journal of Educational and Social Research* 2. URL : <https://ideas.repec.org/a/bjz/jesrjr/135.html>.
- PFEIFFER, Rolf-Helge (2020). « What constitutes Software? An Empirical, Descriptive Study of Artifacts ». In : *Proceedings of the 17th International Conference on Mining Software Repositories*. MSR '20. Seoul, Republic of Korea : Association for Computing Machinery, p. 481-491. ISBN : 9781450375177. DOI : 10.1145/3379597.3387442. URL : <https://doi.org/10.1145/3379597.3387442>.
- REGULATION, The European Union Medical Device (2022). *Intended-purpose*. Anglais. URL : <https://eumdr.com/intended-purpose/>.

- RIZEA, R. E., Karina Lidia GHEORGHITA, M. V. SALACEANU et A. V. CIUREA (mars 2019). « New technics for removal of intradural spinal tumours ». In : *roneuro* 33.1, p. 34-36. DOI : [10.33962/roneuro-2019-006](https://doi.org/10.33962/roneuro-2019-006). URL : <https://journals.lapub.co.uk/index.php/roneurosurgery/article/view/328>.
- ROYCE, WINSTON (1970). « Managing the development of large software systems. proceedings of IEEE WESCON ». In : *Los Angeles*, p. 328-388.
- RUIZ, FRANCISCO et José R. HILERA (2006). « Using Ontologies in Software Engineering and Technology ». In : *Ontologies for Software Engineering and Software Technology*. Sous la dir. de Coral CALERO, FRANCISCO RUIZ et Mario PIATTINI. Berlin, Heidelberg : Springer Berlin Heidelberg. Chap. 2, p. 49-102. ISBN : 978-3-540-34518-3. DOI : [10.1007/3-540-34518-3_2](https://doi.org/10.1007/3-540-34518-3_2). URL : https://doi.org/10.1007/3-540-34518-3_2.
- RUST, Peter, Derek FLOOD et Fergal McCAFFERY (2015). « Software process improvement and roadmapping : a roadmap for implementing IEC 62304 in organizations developing and maintaining medical device software ». In : *International Conference on Software Process Improvement and Capability Determination*. Springer, p. 19-30. eprint : http://eprints.dkit.ie/546/1/02_Rust_SPIRoadmappingIEC62304SPICE2015.pdf.
- (2016a). « Creation of an IEC 62304 compliant software development plan ». In : *Journal of Software : Evolution and Process* 28.11, p. 1005-1010. DOI : <https://doi.org/10.1002/smr.1826>. eprint : <https://onlinelibrary.wiley.com/doi/pdf/10.1002/smr.1826>. URL : <https://onlinelibrary.wiley.com/doi/abs/10.1002/smr.1826>.
- (2016b). « Software process improvement roadmaps—using design patterns to aid SME’s developing medical device software in the implementation of IEC 62304 ». In : *International Conference on Software Process Improvement and Capability Determination*. Springer, p. 43-56.
- SANTOS, Isa CT, G Scott GAZELLE, Luis A ROCHA et João Manuel RS TAVARES (2012). « An ontology model for the medical device development process in Europe ». In : *The 1st International Conference on Design and Processes for Medical Devices-PROMED 2012*.
- SEDRAKYAN, Art, Bruce CAMPBELL, Jose G MERINO, Richard KUNTZ, Allison HIRST et Peter McCULLOCH (2016). « IDEAL-D : a rational framework for evaluating and regulating the use of medical devices ». In : *BMJ* 353. DOI : [10.1136/bmj.i2372](https://doi.org/10.1136/bmj.i2372). eprint : <https://www.bmj.com/content/353/bmj.i2372.full.pdf>.
- SELMİ, Sonia-Yuki, Emmanuel PROMAYON, Johan SARRAZIN et Jocelyne TROCCAZ (2014). « 3D Interactive Ultrasound Image Deformation for Realistic Prostate Biopsy Simulation ». In : *Biomedical Simulation*. Sous la dir. de Fernando BELLO et Stéphane COTIN. Cham : Springer International Publishing, p. 122-130.
- SESSA, Luca, Cyril PERRENOT, Song XU, Jacques HUBERT, Laurent BRESLER, Laurent BRUNAUD et Manuela PEREZ (2018). « Face and content validity of Xperience™ Team Trainer : bed-side assistant training simulator for robotic surgery ». In : *Updates in Surgery* 70.1, p. 113-119. ISSN : 2038-3312. DOI : [10.1007/s13304-017-0509-x](https://doi.org/10.1007/s13304-017-0509-x). URL : <https://doi.org/10.1007/s13304-017-0509-x>.
- SEYHAN, Attila A. (2019). « Lost in translation : the valley of death across preclinical and clinical divide - identification of problems and overcoming obstacles ». In :

- Translational Medicine Communications* 4.1, p. 18. ISSN : 2396-832X. DOI : [10.1186/s41231-019-0050-7](https://doi.org/10.1186/s41231-019-0050-7). URL : <https://doi.org/10.1186/s41231-019-0050-7>.
- STAUB, C., S. CAN, B. JENSEN, A. KNOLL et S. KOHLBECHER (2012). « Human-computer interfaces for interaction with surgical tools in robotic surgery ». In : *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechanics (BioRob)*, p. 81-86. DOI : [10.1109/BioRob.2012.6290850](https://doi.org/10.1109/BioRob.2012.6290850).
- STEFINI, Roberto, Stefano PERON, Jaime MANDELLI, Elena BIANCHINI et Paolo ROCCUCCI (2018). « Intraoperative Spinal Navigation for the Removal of Intradural Tumors : Technical Notes ». In : *Operative Neurosurgery* 15.1. ISSN : 2332-4252. URL : https://journals.lww.com/onsonline/fulltext/2018/07000/intraoperative_spinal_navigation_for_the_removal.8.aspx.
- TAHERDOOST, Hamed (avr. 2016). « Validity and Reliability of the Research Instrument; How to Test the Validation of a Questionnaire/Survey in a Research ». In : *International Journal of Academic Research in Management* 5.3, p. 28-36. ISSN : 2296-1747.
- TIMC (2024). URL : <https://www.timc.fr/gmcao>.
- TIMC, LTSI (2016). *Augmented Endoscopy*. URL : <https://cami-labex.fr/augmented-endoscopy/>.
- TROCCAZ, Jocelyne (2009). « Computer and Robot-Assisted Medical Intervention ». In : *Springer Handbook of Automation*. Sous la dir. de Shimon Y. Nof. Berlin, Heidelberg : Springer Berlin Heidelberg, p. 1451-1466. DOI : [10.1007/978-3-540-78831-7_82](https://doi.org/10.1007/978-3-540-78831-7_82). URL : https://doi.org/10.1007/978-3-540-78831-7_82.
- TROCCAZ, Jocelyne, Stéphane LAVALLÉE et Philippe CINQUIN (1996). « Computer-augmented surgery ». In : *Human Movement Science* 15.3, p. 445-475. ISSN : 0167-9457. DOI : [10.1016/0167-9457\(96\)00010-3](https://doi.org/10.1016/0167-9457(96)00010-3). URL : <https://www.sciencedirect.com/science/article/pii/0167945796000103>.
- VALENT, Peter et al. (fév. 2016). « Paul Ehrlich (1854-1915) and His Contributions to the Foundation and Birth of Translational Medicine ». In : *J Innate Immun* 8.2, p. 111-120. ISSN : 1662-811X. DOI : [10.1159/000443526](https://doi.org/10.1159/000443526). URL : <https://doi.org/10.1159/000443526>.
- VARACHIU, N. (2020). « Make Innovation Happen : Scientific and Statistic Tools to Accelerate the Way Toward Technology Readiness Level TRL 9—A Deployed Application ». In : *4th International Conference on Nanotechnologies and Biomedical Engineering*. Sous la dir. d'Ion TIGINYANU, Victor SONTEA et Serghei RAILEAN. Cham : Springer International Publishing, p. 755-760. ISBN : 978-3-030-31866-6.
- VIVALTO-SANTÉ (2024). URL : <https://hautsdefrance.vivalto-sante.com/cmco/>.
- VOROS, S., G. P. HABER, J. F. MENUDET, J. A. LONG et P. CINQUIN (2010). « ViKY Robotic Scope Holder : Initial Clinical Experience and Preliminary Results Using Instrument Tracking ». In : *IEEE/ASME Transactions on Mechatronics* 15.6, p. 879-886. ISSN : 1941-014X. DOI : [10.1109/TMECH.2010.2080683](https://doi.org/10.1109/TMECH.2010.2080683).
- WICHMAN, Chris, Lynette M. SMITH et Fang YU (2021). « A framework for clinical and translational research in the era of rigor and reproducibility ». In : *Journal of Clinical and Translational Science* 5.1, e31. DOI : [10.1017/cts.2020.523](https://doi.org/10.1017/cts.2020.523).
- WIKIPEDIA (2024a). URL : https://fr.wikipedia.org/wiki/Canard_de_Vaucanson. – (2024b). En. URL : https://en.wikipedia.org/wiki/Philipp_Bozzini#.

- WOOLF, Steven H. (jan. 2008). « The Meaning of Translational Research and Why It Matters ». In : *JAMA* 299.2, p. 211-213. ISSN : 0098-7484. DOI : [10.1001/jama.2007.26](https://doi.org/10.1001/jama.2007.26). URL : <https://doi.org/10.1001/jama.2007.26>.
- YOCK, Paul G., Stefanos ZENIOS, Josh MAKOWER, Todd J. BRINTON, Uday N. KUMAR, F. T. Jay WATKINS, Lyn DENEND, Thomas M. KRUMMEL et Christine Q. KURIHARA (2015). *Biodesign : The Process of Innovating Medical Technologies*. 2^e éd. Cambridge : Cambridge University Press. DOI : [10.1017/CB09781316095843](https://doi.org/10.1017/CB09781316095843). URL : <https://www.cambridge.org/core/product/F71806C33C7BFA8C1B9A5660F68B2858>.
- YUSOFF, Muhamad Saiful Bahri (2019). « ABC of content validation and content validity index calculation ». In : *Education in Medicine Journal* 11.2, p. 49-54.
- YUSOFF, Muhamad Saiful Bahri, Wan Nor ARIFIN et Siti Nurma Hanim HADIE (2021). « ABC of questionnaire development and validation for survey research. » In : *Education in Medicine Journal* 13.1.

DECLARATION

Je soussigné(e) Jean-Loup Habermusch, déclare : N'avoir aucun lien ou conflit d'intérêt dans le cadre des travaux de cette thèse.

La Tronche, 19 juillet 2024

Jean-Loup HABERBUSCH

