

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

THÈSE PRÉSENTÉE À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DU
DOCTORAT EN GÉNIE
PH.D.

PAR
ASMA SELLAMI

PROCESSUS DE VÉRIFICATION DES MESURES DE LOGICIELS
SELON LA PERSPECTIVE DE MÉTROLOGIE

MONTRÉAL, 24 OCTOBRE 2005

(c) droits réservés de Asma Sellami

CETTE THÈSE A ÉTÉ ÉVALUÉE
PAR UN JURY COMPOSÉ DE :

M. Alain Abran, directeur de thèse
Département de génie logiciel et des technologies de l'information
à l'École de technologie supérieure

M. Pierre Bourque, président du jury
Département de génie logiciel et des technologies de l'information
à l'École de technologie supérieure

M. Witold Suryn, membre du jury
Département de génie logiciel et des technologies de l'information
à l'École de technologie supérieure

Mme Maya Daneva, membre du jury
Informatique Department, University of Twente au Pays Bas

ELLE A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC
LE 15 SEPTEMBRE 2005
À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

À ma mère

PROCESSUS DE VÉRIFICATION DES MESURES DE LOGICIELS SELON LA PERSPECTIVE DE MÉTROLOGIE

Asma Sellami

SOMMAIRE

Tandis que le domaine de la mesure et des instruments de mesure est à la fois fondamental et déjà très mature dans les diverses disciplines de l'ingénierie, il est à peine en émergence dans la nouvelle discipline du génie logiciel. En génie logiciel, les propositions actuelles de mesures sont en effet limitées à des concepts se rapprochant non pas à la sous-discipline classique de la métrologie, mais plutôt à la description générale d'algorithmes, sans système de référence, sans instrumentation de mesure et sans notion de protocoles de mesures. Notre recherche s'insère dans ce contexte afin de transposer au domaine du génie logiciel les principaux concepts classiques de la métrologie et ce, tant pour la conception de mesure de logiciels que pour l'analyse de la qualité des instruments de mesure qui en découlent.

Cette thèse propose un cadre intégré de vérification des méthodes de mesure en se basant principalement sur les concepts de mesure identifiés dans le Vocabulaire International des termes fondamentaux et généraux de Métrologie (VIM) d'ISO, le processus de mesure d'Abran *et al.*, ainsi que le cadre de vérification proposé par Kitchenham *et al.*

Le cadre intégré proposé pour la vérification des méthodes de mesure rencontre les critères usuels du domaine de l'instrumentation de mesures en science et en génie et tient compte des caractéristiques spécifiques du produit intellectuel qu'est un logiciel. Ce cadre pourra constituer un modèle d'une méta-norme de mesure pour aider l'industrie à évaluer la qualité des mesures proposées, d'une part, et pour bâtir les futures normes ISO en mesure en génie logiciel, d'autre part.

Pour rencontrer les objectifs de notre recherche, la méthodologie suivante a été suivie :

- modéliser les concepts du vocabulaire ISO de la métrologie (VIM), afin de faciliter leur compréhension et leur utilisation;
- développer un cadre de référence de vérification de mesure en se basant sur le modèle du VIM et sur les éléments du processus de mesure d'Abran *et al.* pour obtenir une envergure plus large de vérification;
- illustrer l'utilisation pratique du cadre de référence de vérification de mesure en :

- analysant les normes internationales ISO et le domaine de connaissance de la mesure en génie logiciel pour identifier leurs forces et leurs faiblesses;
- procédant à la transposition des travaux expérimentaux faits sur une mesure spécifique, à leur utilisation à plus grande échelle sur l'ensemble des mesures en génie logiciel.
- développer un cadre intégré de mesure en utilisant principalement le cadre de référence de mesure et les éléments du cadre de vérification de mesure de Kitchenham *et al.*;
- identifier les critères pertinents de vérification des mesures de la perspective de la métrologie.

Les résultats de ces travaux de recherche pourront servir de base pour d'autres recherches subséquentes afin de concevoir de nouvelles mesures du logiciel qui respecteraient les concepts fondamentaux de la métrologie, et pour l'amélioration des mesures du logiciel déjà proposées à l'industrie.

Enfin, l'ensemble des modèles développés n'est pas restreint uniquement au génie logiciel mais en général à l'ensemble des domaines de la mesure.

Commentaire [MSOffice1] : Normal, justifié.

VERIFICATION PROCESS OF SOFTWARE MEASUREMENT WITHIN METROLOGY PERSPECTIVE

Asma Sellami

ABSTRACT

Commentaire [MSOffice2] : N'est pas un titre. Majuscule, centré, gras.

Commentaire [MSOffice3] : Titre sommaire

While the field of measurement and measuring instruments is both fundamental and very mature in the traditional engineering disciplines, it remains an emerging one in the software engineering discipline, to the extent that current proposals for measurement are limited to a general description of algorithms without any reference to measurement instrumentation or measurement method; in practice, these measurement proposals use very few metrology concepts. This research maps the fundamental concepts of metrology into the software engineering discipline, and does so not only for the design of software measures, but also for the analysis of the quality of the related measuring instruments.

In this R&D project, we have proposed an integrated measurement verification framework based on measurement concepts as documented in the ISO International Vocabulary of Basic and General Terms in Metrology (VIM), and also on the elements of measurement process model of Abran et al. and the software measurement verification framework of Kitchenham et al.

This integrated framework can be useful in a model of a measurement meta-standard to verify software measurement methods, to improve current ISO software engineering standards on measurement and to build future ISO measurement standards in software engineering.

The methodology designed to pursue the objectives of this research consists of the following:

- modeling the measurement concepts of the ISO International Vocabulary in Metrology (VIM) to facilitate their understanding and their use;
- developing a reference framework for measurement verification according to the detailed model of the VIM and the measurement process elements of Abran et al. to provide a broader scope of verification;
- illustrating the practical uses of this reference measurement verification framework by:

- analyzing the measurement-related ISO standards on software engineering and the software measurement body of knowledge in order to identify their strengths and weaknesses,
- transposing the experimental work associated with a specific measure to the larger scale of the set of measurements in software engineering;
- developing an integrated framework for measurement verification using mainly the measurement reference framework and the elements of the software measurement validation framework of Kitchenham et al.;
- identifying the relevant criteria for measurement verification from the metrology perspective.

Industry and researchers can use our research results to design new software measures which are aligned with the concepts of metrology, and to improve the software measures already suggested to industry.

Finally, it is to be noted that the metrology models developed are not restricted to software engineering, but are more generic in they can be applied throughout the measurement domain.

Commentaire [MSOffice4] : Normal, justifié

REMERCIEMENTS

Tout d'abord, je voudrais exprimer ma gratitude la plus profonde au professeur Alain Abran, mon directeur de thèse, pour ses conseils judicieux, ses suggestions d'éclaircissement et son support continu tout au long de la préparation de cette thèse. Ses discussions utiles m'ont aidé à clarifier mes idées et à affiler mes arguments. Je suis particulièrement reconnaissante pour sa compréhension et sa bonne volonté à m'inclure dans ses priorités.

Ensuite, je tiens à témoigner ma gratitude au professeur Pierre Bourque pour avoir accepté la présidence du jury et avec lequel j'ai eu le plaisir d'avoir des échanges scientifiques significatifs et enrichissants. Je voudrais également remercier le professeur Witold Suryn pour ses conseils techniques et enthousiastes. Je tiens aussi à remercier professeur Maya Daneva d'avoir bien voulu être membre externe sur ce jury. Par ailleurs, je tiens à remercier Mme Marthe Lacroix pour les bons conseils concernant la qualité linguistique du texte, et tous les membres du Laboratoire de recherche en génie logiciel de l'ÉTS qui m'ont aidé à mener à bien ce travail.

Une dette spéciale de gratitude va à mon mari qui n'a épargné aucun effort pour m'aider à faire ce travail jusqu'à sa conclusion. J'ai pu également compter sur l'encouragement et le support continu de ma mère qui m'a aidé à atteindre mes objectifs.

Cette thèse est dédiée à tous les membres de ma famille, à mes amis et à mes collègues. Elle est dédiée également avec gratitude à tous mes professeurs de la Faculté des sciences économiques et de gestion de Sfax en Tunisie, de l'Université du Québec à Montréal et de l'École de technologie supérieure. Chacun d'eux a contribué à ce travail avec des perspectives importantes.

Merci à tous.

TABLE DES MATIÈRES

Commentaire [MSOffice6] : TITRE

Page

Commentaire [MSOffice7] : Normal, justifié à droite

Commentaire [MSOffice8] : M1: Titre, Titre 1; TM2: Titre2, Titre3,

SOMMAIRE	i
ABSTRACT	iii
REMERCIEMENTS	v
TABLE DES MATIÈRES	vii
LISTE DES TABLEAUX	xi
LISTE DES FIGURES	xiv
LISTE DES ABRÉVIATIONS ET DES SIGLES	xvi
CHAPITRE 1 INTRODUCTION	17
1.1 Mesure en ingénierie et en sciences	18
1.1.1 Nature de mesurage	19
1.1.2 Étalon de mesure	19
1.1.3 Mesures directes et mesures indirectes	21
1.2 Mesure en génie logiciel	21
1.2.1 Bref historique	21
1.2.2 Contexte prévu de la mesure en génie logiciel	23
1.2.3 Problématique de la mesure en génie logiciel	24
1.2.4 Manque de consensus sur la place de la mesure en génie logiciel	26
1.2.5 Théorie de la mesure et métrologie en génie logiciel	33
1.2.6 Les types d'échelle de mesure en génie logiciel	34
1.3 Problématique et objectifs	38
1.4 Choix d'une approche de recherche : la métrologie	41
1.4.1 Problèmes de mesure au niveau pratique	42
1.4.2 Contexte du projet de recherche	44
1.5 Avertissement terminologique : vérification et validation	45
1.6 Structure du rapport	46
CHAPITRE 2 LES OUTILS ANALYTIQUES DE VÉRIFICATION DE MESURE ..	48
2.1 Introduction	48
2.2 Le domaine de connaissance de la mesure du VIM	49
2.3 Modèle du processus de mesure de Abran et Jacquet	50

2.4	Classification des connaissances de l'ingénierie de Vincenti	51
2.5	Corpus de connaissances de Schneidewind.....	53
2.6	Cadre de validation proposé par Kitchenham <i>et al.</i>	54
2.7	Vers une perspective métrologique de la mesure du logiciel	56
2.8	Sommaire et observations.....	57
CHAPITRE 3 MÉTHODOLOGIE DE RECHERCHE.....		59
3.1	Introduction	59
3.2	Méthodologie : Démarche globale	59
3.2.1	Volet théorique	59
3.2.2	Volet pratique	60
3.2.3	Volet intégrateur	61
3.3	Démarche détaillée	62
3.3.1	Volet théorique	63
3.3.2	Volet pratique	65
3.3.3	Volet intégrateur	66
3.4	Vérification.....	69
CHAPITRE 4 MODÉLISATION DES CONCEPTS DE MESURE DANS VIM'93 ...		71
4.1	Introduction	71
4.2	PARTIE I : Modèle initial du VIM'93	73
4.2.1	Grandeurs et unités	74
4.2.2	Mesurages.....	75
4.2.3	Résultats de mesure	77
4.2.4	Instruments de mesure	77
4.2.5	Caractéristiques des instruments de mesure	81
4.2.6	Étalons	82
4.3	PARTIE II : Modèle détaillé du VIM'93	83
4.3.1	Grandeurs et unités	84
4.3.2	Mesurages.....	96
4.3.3	Résultats de mesure	98
4.3.4	Instruments de mesure	100
4.3.5	Caractéristiques des instruments de mesure	104
4.3.6	Étalons	104
4.4	Sommaire et observations.....	108
CHAPITRE 5 CADRE DE RÉFÉRENCE DE VÉRIFICATION D'UNE MESURE.		111
5.1	Introduction	111
5.2	Alignement des principaux concepts du VIM avec les étapes du processus de mesure.....	111
5.3	Sommaire et observations.....	115
CHAPITRE 6 UTILISATION DU CADRE DE RÉFÉRENCE POUR ANALYSER LES NORMES ISO DE MESURE		116

6.1	Introduction	116
6.2	ISO/IEC 9126 : Qualité d'un produit logiciel	117
6.2.1	Survol d'ISO 9126.....	117
6.2.2	Analyse de haut niveau de ISO 9126-2 à 4	121
6.2.3	Analyse détaillée de ISO 9126-2 à 4	122
6.3	Requis de métrologie.....	131
6.4	ISO/IEC 14143 : Méthodes de mesure de la taille fonctionnelle du logiciel .	132
6.4.1	Survol de ISO/IEC 14143.....	132
6.4.2	Analyse de ISO/IEC 14143	135
6.5	ISO/IEC 19761 : COSMIC-FFP.....	137
6.5.1	Survol de COSMIC-FFP : ISO 19761	138
6.5.2	Analyse du design de COSMIC-FFP.....	140
6.5.3	Analyse de mesurage avec COSMIC-FFP	143
6.5.4	Analyse de COSMIC-FFP : instruments de mesure et résultats de mesure ...	145
6.5.5	Caractéristiques des instruments de mesure	149
6.6	Sommaire et observations.....	153
CHAPITRE 7 ANALYSE DE ISO 15939 ET DE COCOMOII.....		156
7.1	Introduction	156
7.2	Processus de mesure de logiciels (ISO 15939).....	156
7.2.1	Survol de la norme ISO 15939	158
7.2.2	Modèle du processus de mesure du logiciel de la norme ISO 15939.....	159
7.2.3	Analyse de la norme ISO 15939.....	160
7.3	Interprétation pratique : analyse de COCOMO II	166
7.4	Sommaire.....	171
CHAPITRE 8 ANALYSE DU CADRE PROPOSÉ PAR KITCHENHAM <i>et al.</i>		173
8.1	Introduction	173
8.2	Cadre de vérification dans [82] et entités types correspondantes.....	174
8.3	Analyse structurée du cadre de vérification proposé dans [82].....	176
8.3.1	Design d'une méthode de mesure : éléments à vérifier.....	176
8.3.2	Vérification de l'application d'une méthode de mesure.....	180
8.3.3	Éléments de vérification pour les résultats obtenus en mesure	182
8.3.4	Vérification de l'utilisation des résultats de mesure.....	183
8.4	Résumé des observations et recommandations	185
8.5	Sommaire.....	188
CHAPITRE 9 ANALYSE DU CORPUS DES CONNAISSANCES DE LA MESURE EN GÉNIE LOGICIEL.....		189
9.1	Introduction	189
9.2	Survol du guide SWEBOK.....	189
9.3	Méthodologie de l'analyse du domaine de connaissance de la mesure.....	193
9.3.1	Taxonomie initiale du KA pour les mesures de logiciels dans SWEBOK.....	193
9.3.2	Classification préliminaire des concepts du BOK de mesure.....	197

9.3.3	Mise en correspondance de la mesure dans SWEBOK avec le cadre de référence.....	202
9.4	Sommaire.....	206
9.5	Recommandations	207
CHAPITRE 10 PROPOSITION D'UN CADRE INTÉGRÉ DE VÉRIFICATION D'UNE MESURE		208
10.1	Introduction	208
10.2	Cadre intégré de vérification de mesure.....	208
10.3	Étape 1 : Design d'une méthode de mesure	209
10.3.1	Définition du contexte de design d'une mesure	209
10.3.2	Identification des concepts d'intrant au processus de mesurage	210
10.3.3	Modélisation du monde empirique.....	212
10.3.4	Définition du monde numérique (mathématique)	212
10.4	Étape 2 : Application d'une méthode de mesure.....	213
10.4.1	Utilisation de l'instrument de mesure.....	214
10.4.2	Identification des caractéristiques de l'instrument de mesure.....	214
10.5	Étape 3 : Vérification des résultats de mesure.....	215
10.6	Identification des critères du cadre intégré de vérification d'une mesure.....	216
10.6.1	Vérification du design de la méthode de mesure.....	217
10.6.2	Vérification de l'application de la méthode de mesure	217
10.6.3	Vérification des résultats obtenus.....	218
10.7	Conclusion.....	218
CHAPITRE 11 CONCLUSION		219
11.1	Bilan du travail	219
11.2	Contribution dans d'autres travaux de recherche	229
11.3	Perspectives d'avenir.....	230
ANNEXE A Liste des publications		233
ANNEXE B Travaux ISO en cours sur la métrologie		242
ANNEXE C Termes (et concepts relatifs) de métrologie documentés dans VIM'93 ...		261
RÉFÉRENCES		266

LISTE DES TABLEAUX

Commentaire [MSOffice9] : TITRE

Page

Commentaire [MSOffice10] : normal, justifié à droite

Tableau I	Liste des principes fondamentaux candidats [23].....	27
Tableau II	Types d'échelle	35
Tableau III	Matrice de complexité fonctionnelle (IFPUG –Guide 85).....	36
Tableau IV	Éléments de vérification d'une méthode de mesure du logiciel	46
Tableau V	Catégorie des termes dans VIM'93.....	49
Tableau VI	Classification de Vincenti des catégories de connaissance de l'ingénierie.....	52
Tableau VII	Le BOK en mesure de Schneidewind [101].....	53
Tableau VIII	Structure du cadre de validation présenté dans [82]	55
Tableau IX	Étape à suivre (inspiré de [44]).....	69
Tableau X	Topologie détaillée de l'ensemble des concepts « grandeurs et unités »	75
Tableau XI	Topologie détaillée de la catégorie « résultat d'un mesurage ».....	77
Tableau XII	Topologie détaillée des échelles d'un appareil de mesure	80
Tableau XIII	Opérations sur les instruments de mesure.....	80
Tableau XIV	Topologie détaillée des « caractéristiques des instruments de mesure »	81
Tableau XV	Topologie détaillée d'un étalon.....	82
Tableau XVI	Caractéristiques d'une grandeur.....	85
Tableau XVII	Grandeur de base.....	87
Tableau XVIII	Grandeur dérivée.....	88
Tableau XIX	Valeur d'une grandeur.....	94
Tableau XX	Échelle de repérage	94
Tableau XXI	Termes additionnels dans la catégorie « grandeurs et unités »	96
Tableau XXII	Modes de vérification des résultats de mesure et leurs conditions	99
Tableau XXIII	Types d'appareils de mesure.....	100

Tableau XXIV	Détails d'un appareil de mesure afficheur/ indicateur	101
Tableau XXV	Hiérarchie des étalons [43]	105
Tableau XXVI	Classification des caractéristiques d'un étalon	107
Tableau XXVII	Alignement des concepts de mesure	113
Tableau XXVIII	Différentes entités de mesure sujettes aux critères de vérification.....	114
Tableau XXIX	Structure des caractéristiques et des sous-caractéristiques de la qualité dans ISO/IEC 9126.....	118
Tableau XXX	Mise en correspondance de haut niveau de ISO 9126-2 à 4 et cadre de référence.....	122
Tableau XXXI	Mesure du « temps de la tâche » pour la caractéristique de « productivité » dans 9126-4.....	124
Tableau XXXII	Mesure de « l'efficacité de la tâche » pour la « productivité » dans TR 9126-4.....	127
Tableau XXXIII	Mesure de « l'efficacité de la tâche M_1 » pour la caractéristique « efficacité » dans 9126-4.....	128
Tableau XXXIV	Récapitulatif de l'analyse détaillée de TR 9126-2 à 4	130
Tableau XXXV	Positionnement de ISO 14143-1 à 5 selon le cadre de référence.	136
Tableau XXXVI	Concepts « grandeurs et unités » de métrologie dans COSMIC-FFP	142
Tableau XXXVII	Concepts de métrologie « mesurage » dans COSMIC-FFP.....	145
Tableau XXXVIII	Concepts des « instruments de mesure » et COSMIC/ RUP.....	147
Tableau XXXIX	Concepts des « résultats de mesure » et prototype COSMIC/RUP	149
Tableau XL	Concepts de caractéristiques d'utilisation et prototype COSMIC/RUP	151
Tableau XLI	Concepts des caractéristiques de contrôle et prototype COSMIC/RUP	153
Tableau XLII	Positionnement de l'utilisation des mesures pour un projet logiciel dans ISO 15939.....	160
Tableau XLIII	Facteurs d'échelle et multiplicateurs de l'effort du modèle COCOMO II [22].....	168
Tableau XLIV	Les facteurs du COCOMO II et leurs coefficients.....	170

Tableau XLV	Alignement du modèle COCOMO II [22] avec le cadre de référence	171
Tableau XLVI	Mise en correspondance du cadre de vérification dans [82] avec le cadre de référence	174
Tableau XLVII	Design d'une méthode de mesure – critères de vérification	180
Tableau XLVIII	Classification des éléments de vérification des méthodes de mesure du logiciel	187
Tableau XLIX	Guide SWEBOK : KA [9]	191
Tableau L	Mise en correspondance des thèmes de mesure de Schneidewind par rapport à la taxonomie de mesure du logiciel	195
Tableau LI	Classification de Vincenti des connaissances de l'ingénierie et des critères associés.....	198
Tableau LII	Classification préliminaire des concepts du BOK de la mesure [27] et ceux de Vincenti [109]	200
Tableau LIII	Sommaire préliminaire des thèmes de mesure de Vincenti et BOK	201
Tableau LIV	Classification du thème de la mesure dans SWEBOK avec le cadre de référence	203
Tableau LV	Cadre intégré de vérification de mesure.....	208
Tableau LVI	Catégories des termes en Métrologie.....	244
Tableau LVII	Topologie détaillée de l'ensemble des concepts « grandeurs et unités ».....	247
Tableau LVIII	Exemples du VIM	250
Tableau LIX	Measurement Foundations Concepts	252
Tableau LX	Concepts en relation avec le « résultat de mesure ».....	253
Tableau LXI	Composants des dispositifs pour mesurage et conditions opérationnelles des systèmes de mesure	256
Tableau LXII	Topologie détaillée des « caractéristiques d'un système de mesure »	257
Tableau LXIII	Topologie détaillée des « Étalons »	258

LISTE DES FIGURES

Commentaire [MSOffice11] :
ITRE

Page

Commentaire [MSOffice12] :
ormal, justifié à droite

Figure 1	Évolution d'une discipline de génie [103]	33
Figure 2	Processus de Mesure – Modèle de Haut Niveau [6;76]	51
Figure 3	Modèle structurel de mesure (tiré de Kitchenham <i>et al.</i> [82])	56
Figure 4	Démarche globale de recherche	62
Figure 5	Méthodologie détaillée de recherche	68
Figure 6	Modèle des processus des catégories des termes VIM'93	72
Figure 7	Topologie de haut niveau de la catégorie « grandeurs et unités »	74
Figure 8	Fondements de la mesure	76
Figure 9	Processus de mesure –Topologie détaillée des sous-concepts.....	76
Figure 10	Chaîne de mesure	78
Figure 11	Système de mesure.....	78
Figure 12	Appareil intégrateur	78
Figure 13	Détail d'un instrument de mesure	79
Figure 14	Modèle d'un appareil de mesure totalisateur	79
Figure 15	Catégories de grandeurs	85
Figure 16	Grandeur de dimension un/ Grandeur sans dimension	88
Figure 17	Système de grandeurs.....	90
Figure 18	Système de grandeurs, système d'unités et système cohérent d'unités...	91
Figure 19	Unité de mesure hors système, multiple d'une unité et sous-multiple d'une unité.....	92
Figure 20	Unité de mesure.....	92
Figure 21	Lien entre grandeur, unité et étalon.....	93
Figure 22	Valeur numérique obtenue par comptage	94
Figure 23	Topologie détaillée de « mesurage ».....	97
Figure 24	Phénomène de transformation d'un signal de mesure	97
Figure 25	Instrument de mesure et dispositifs associés.....	101

Figure 26	Modèle général - Succession des concepts de la catégorie « instrument de mesure » selon un ordre approximatif de complexité	102
Figure 27	Composants d'un système de mesure général [43]	102
Figure 28	Liaison entre étalons	106
Figure 29	Processus de mesure – modèle détaillé [6]	112
Figure 30	Relation entre les types de mesure [60]	119
Figure 31	Un modèle général pour mesurer la taille fonctionnelle du logiciel [4;69]	139
Figure 32	Les types de mouvements de données et certaines de leurs relations [4;69]	140
Figure 33	Modèle du processus de mesure du logiciel (traduit de la norme ISO/IEC 15939 [67])	159
Figure 34	Modèle de l'information de mesure (traduit de la norme ISO/IEC 15939 [67])	163
Figure 35	Modèle hiérarchique des concepts de design d'une méthode de mesure, comme discuté dans [82]	177
Figure 36	Modèle d'association entre attributs sur la base du texte dans [82]	178
Figure 37	« Modèle de définition de l'unité » sur la base du texte dans [82]	181
Figure 38	« Processus de mesure » sur la base du texte dans [82]	181
Figure 39	Éléments de vérification d'un instrument de mesure	182
Figure 40	Attributs des composants sur la base du texte de [82]	183
Figure 41	Organisation de la description du domaine de connaissance dans SWEBOK	191
Figure 42	Méthodologie d'analyse	193
Figure 43	Taxonomie pour KA de mesure de logiciels dans SWEBOK [27]	194
Figure 44	Nouvelle taxonomie pour un corpus de connaissance de mesure du logiciel [2]	205
Figure 45	Modèle du processus des catégories des termes VIM'2004	244
Figure 46	Niveaux des grandeurs	248
Figure 47	Modèle structurel de la catégorie « grandeurs et unités »	249
Figure 48	Topologie de haut-niveau du processus de mesure	251
Figure 49	Éléments d'une chaîne de mesure	255
Figure 50	Système de mesure	256

LISTE DES ABRÉVIATIONS ET DES SIGLES

BoK	Body of Knowledge
Cfsu	<u>C</u> osmic <u>f</u> unctional <u>s</u> ize <u>u</u> nit
COSMIC	Common Software Measurement International Consortium
FFP	Full Function Point
FPA	Function Point Analysis
FUR	Functional User Requirements
GÉLOG	Laboratoire de recherche en génie logiciel
IFPUG	International Function Point Users Group
KA	Knowledge Area
MKII	MKII Function Point Analysis
MIS	Management Information System
MTF	Mesure de la Taille Fonctionnelle
NESMA	Netherlands Software Metrieken Gebruikers Associatie
PMI	Project Management Institute
PSM	Practical Software Measurement
SLCP	Software Life Cycle Phase
SWEBOK	Software Engineering Body of Knowledge
V&V	Verification & Validation
VIM	Vocabulaire International des Termes Fondamentaux et Généraux de Métrologie

CHAPITRE 1

INTRODUCTION

La mesure a toujours joué un rôle important dans tous les domaines de l'ingénierie : en génie électrique, mécanique, chimique et autres, les procédés de mesure ont été développés au cours des XVIII^e et XIX^e siècles. Leur évolution en discipline mature a mis près de deux siècles [103]. Les progrès et l'évolution des sciences apportent souvent des informations nouvelles et plus précises sur certains aspects de la connaissance de la mesure, notamment sur les mécanismes de l'instrumentation de mesure. Le génie logiciel doit, entre autres, adopter maintenant tous ces éléments de mesure pour devenir une branche solide de l'ingénierie.

Deux aspects importants font principalement l'objet des analyses des mesures; le premier consiste à examiner les processus des méthodes de mesure actuellement utilisées et à en analyser les éléments en interrelation, ce qu'on peut appeler l'aspect structurel de la mesure. Le second consiste à identifier les connaissances empiriques et l'explication des représentations théoriques, ce qu'on peut appeler l'aspect opérationnel ou pratique de la mesure. Les deux aspects sont nécessaires sans privilégier l'un au détriment de l'autre. Fenton *et al.* [41] mentionnent qu'« il est difficile d'imaginer le génie électrique, mécanique ou civil sans un rôle central de la mesure » tandis que « la mesure est considérée comme un luxe en génie logiciel (...) où les mesures sont habituellement accomplies de manière peu fréquente, inconsistante et incomplète ».

En génie logiciel, durant les quatre dernières décades la plupart des travaux de recherche sur les mesures se sont basés sur le développement des mesures pour la compréhension, l'évaluation, le contrôle et la prédiction des attributs du logiciel tels que l'effort de développement, la fiabilité du logiciel et la productivité du personnel dans l'objectif d'améliorer principalement le processus de développement et de maintenance de

logiciels. Cependant, actuellement ces travaux de recherche en mesure de logiciels ne semblent pas encore aboutir aux objectifs prévus. Cela ne veut pas dire que les mesures utilisées en génie logiciel sont entièrement fausses et à rejeter totalement, mais plutôt qu'elles sont probablement insuffisantes pour évaluer et pour contrôler les différents aspects d'un logiciel.

Y a-t-il donc une logique ou un cheminement standard pour la compréhension des mesures de logiciels? Quels sont les critères de vérification de la qualité des mesures proposées en génie logiciel? Ce travail de recherche explore ces questions et propose une nouvelle perspective de vérification de mesure qui permet d'explorer tant la structure de design d'une mesure en général que son application dans un cas spécifique. Ce chapitre introduit les fondements de mesure en ingénierie et en science de même que les problèmes évoqués en mesure de logiciels et les objectifs à atteindre au cours de cette recherche.

1.1 Mesure en ingénierie et en sciences

Dans cette section, nous présentons l'importance de l'utilisation de la mesure en ingénierie et en sciences. Il est incontestable que la mesure contribue au progrès dans toutes les sciences. Le progrès scientifique en ingénierie se base sur un processus commençant par des observations et des généralisations fondées sur des données quantitatives et des mesures, la dérivation de théories comme hypothèses d'explication et par la suite la confirmation ou le rejet de la théorie via des hypothèses de test basées sur d'autres données empiriques. Ainsi, certaines normes peuvent être accomplies à partir de la confirmation des hypothèses par des experts, c'est-à-dire d'abord à partir de la déduction d'une théorie, puis par une expérience dans la pratique qui confirme le résultat de la déduction ainsi que la théorie. Mais, il peut arriver que cette norme soit établie par une voie guidée par une hypothèse générale et ensuite déduite logiquement d'une théorie. Autrement dit, la théorie précède parfois la pratique et la pratique précède parfois la théorie.

1.1.1 Nature de mesurage

Le mesurage est l'ensemble des opérations destinées à déterminer une valeur d'une grandeur. Les opérations de mesure visent ainsi à répondre à une question commençant par le mot « combien » ; elles consistent à évaluer une grandeur selon son rapport avec une grandeur de même nature, c'est-à-dire à exprimer quantitativement une propriété (attribut) d'un phénomène, d'un corps ou d'une substance. Mesurer, c'est en fait établir une correspondance biunivoque (correspondance un à un) entre un mesurande et un ensemble de nombres entiers, rationnels, irrationnels, réels ou complexes. En d'autres termes, c'est mettre chaque partie ou portion du mesurande en relation avec un nombre différent de l'ensemble des nombres qui est utilisé, en respectant la relation d'ordre en question. Mesurer le poids, c'est mettre ce dernier en correspondance avec la série des nombres entiers positifs. Il en est de même quand on mesure une longueur, une durée ou un volume, sauf dans le cas où le mesurande n'est pas naturellement divisé en unités distinctes et égales [44]. Pour cela, le mesureur utilise une unité de mesure qui est choisie pour des raisons de commodité et par convention eu égard aux circonstances dans lesquelles la mesure est faite : la réalisation de la définition de l'unité mètre (l'étalon mètre) par exemple, est certainement commode pour mesurer les dimensions d'une table ou d'une maison, mais il serait inopérant pour évaluer les dimensions d'une particule infra-atomique, etc.

1.1.2 Étalon de mesure

L'étalon est un concept important dans tous les domaines scientifiques et d'ingénierie. L'étalon consiste en un système de mesure, une mesure matérialisée ou un matériau de référence destiné à définir, à réaliser, à conserver ou à reproduire une unité ou plusieurs valeurs d'une grandeur pour servir de référence [74].

Par exemple, l'étalon mètre définit l'unité mètre. Cet étalon a été défini à l'origine comme étant une fraction de la longueur du méridien de l'équateur au pôle nord, la ligne

imaginaire qui fait le tour de la terre à mi-chemin entre les deux pôles. Il s'est cependant avéré impossible, à la longue, de maintenir une évaluation stable de cette longueur, en raison des mouvements de l'écorce terrestre et du progrès des techniques de mesure. D'où la nécessité, plus tard, d'améliorer cet étalon en définissant un nouveau mètre-étalon comme une barre de platine-iridium faite longitudinalement en forme de X et déposée au Bureau des poids et mesures de Sèvres. Basé initialement sur une longueur naturelle, le mètre est ainsi devenu un étalon de mesure tout à fait artificiel [44]. Le platine-iridium garantissait une variation minimale aux changements de température, la forme longitudinale en X garantissait le maximum de rigidité et, donc, le minimum de déformation par courbure. Plus récemment, le mètre-étalon a été redéfini comme étant la longueur d'onde dans le vide de la radiation de la lumière entre les niveaux $2p_{10}$ et $5d_5$ de l'isotope $86/36$ du krypton (c'est-à-dire $1\,650\,763,73$ de ces longueurs équivalent à un mètre¹). Celui-ci, est plus stable que la barre de platine-iridium et ses subdivisions (le centimètre, le millimètre, etc.) comme étalon standard de longueur, tout au moins pour la mesure des très petites longueurs.

Un étalon de mesure est une réalisation de la définition de l'unité de mesure (avec une valeur déterminée et une incertitude de mesure) utilisé par convention pour effectuer un mesurage. Généralement, pour des raisons d'opération de mesurage, l'étalon est maintenu indivisé, bien qu'il soit en principe divisible. Il doit être stable de telle sorte que les modifications de l'environnement ne provoquent pas chez lui des variations quantitatives appréciables. Il doit aussi être de préférence homogène par rapport au mesurande, pour une plus grande précision et une plus grande régularité des mesures [44]. Ainsi, il vaut mieux mesurer une longueur par une unité de longueur, un poids par une unité de poids, etc. Cela n'est cependant pas toujours possible pour les mesures indirectes ou pour les mesures de ce qui est qualitatif et qu'il faut quantifier artificiellement.

¹ <http://pro.wanadoo.fr/pierre.gay/PagesFra/SysMetFR>

1.1.3 Mesures directes et mesures indirectes

Une mesure directe est établie lorsque l'unité de mesure est homogène par rapport au mesurande. Mais si l'unité de mesure n'est pas homogène par rapport au mesurande, nous obtenons une mesure indirecte. Dans ce cas, le mesurande représente un phénomène qualitatif qui est mesuré indirectement, selon son rapport avec un autre phénomène, quantitatif, mesuré directement. Par exemple, grâce à un thermomètre au mercure, on mesure directement la longueur d'une colonne de mercure, et indirectement la température.

Tandis que les résultats des mesures directes sont des nombres cardinaux qui se prêtent aux quatre opérations arithmétiques élémentaires d'addition, de soustraction, de multiplication et de division, les résultats de la mesure indirecte sont des nombres ordinaux qui ne se prêtent pas à ces opérations et dépendent souvent des opinions des mesureurs. D'autre part, les portions égales de l'intermédiaire quantitatif qui est mesuré directement peuvent ne pas correspondre à des différences uniformes au sein du phénomène qualitatif qui est mesuré indirectement, par exemple, la variation de volume de mercure sous l'effet de la température ainsi que lorsqu'on mesure des températures très hautes ou très basses. Il est préférable d'utiliser d'autres instruments que le thermomètre à mercure.

1.2 Mesure en génie logiciel

1.2.1 Bref historique

Les premiers travaux de recherche sur le développement des mesures en génie logiciel ont été évoqués par Rubey [98] en 1968. Aucune référence précédente sur les mesures de logiciels ne se trouve dans son article. Par la suite, en 1976, McCabe a proposé une approche de mesure pour évaluer la complexité structurelle du code source d'un logiciel [84]. Cette approche de mesure se base sur les concepts de la théorie des graphes. McCabe a considéré la complexité structurelle d'un programme comme étant le nombre

cyclomatique du graphe de flux de contrôle correspondant au programme. En 1977, Halstead a publié son livre intitulé *Elements of Software Science* [46]. Halstead visait à fournir dans ce livre tous les éléments de base pour évaluer les différents attributs d'un logiciel tels que l'effort de développement, la complexité, la durée du codage et la difficulté d'un programme. En 1979, Albrecht a proposé une approche d'analyse des points de fonction (FPA) pour évaluer la taille d'un logiciel de point de vue de l'utilisateur et indépendamment de l'implantation [15]. Cette approche a pour avantage la possibilité d'être appliquée dès le début du cycle de développement du logiciel afin de prédire certains attributs du logiciel tels que l'effort et le temps de développement. En 1984, IFPUG (*International Function Point Users' Group*) a été établi pour maintenir l'approche FPA de Albrecht. Le groupe IFPUG a publié le manuel *Counting Practices Manuals* qui fournit les règles standards pour l'application de FPA [54;55]. Ce manuel avait pour objectif de clarifier les règles de l'approche originale d'Albrecht. Par la suite, en 1988, Symons [107] a introduit plusieurs critiques et difficultés de l'utilisation de FPA et il a proposé une nouvelle approche, Mark II. En 1997, Saint-Pierre *et al.* [105] ont proposé une approche des points de fonction étendus (FFP) comme une extension de (IFPUG FPA 4.0). L'objectif derrière cette extension est de capturer la taille fonctionnelle des applications à temps réel. Par la suite, le groupe COSMIC (*Common Software Measurement International Consortium*) a été formé en 1998 pour développer une méthode améliorée de mesure de la taille fonctionnelle sur la base de l'approche FFP [3]. En mars 2003, la méthode COSMIC-FFP est devenue une norme internationale (ISO/IEC 19761) [69].

Bien que dans la littérature du génie logiciel, des milliers de mesures aient été proposées dans un objectif d'évaluation, de contrôle et/ou de prédiction des différents attributs d'un produit logiciel, le besoin d'une approche rigoureuse s'est rapidement fait sentir au sein de la communauté du logiciel. Il n'y a que quatre méthodes de mesure (IFPUG [54], MkII [87], NESMA [89] et COSMIC-FFP [69]) qui ont fait l'objet de normalisation ISO

pour les meilleures pratiques de la mesure en génie logiciel. Mais, résisteront-elles au fil du temps comme méthodes de mesure? Y a-t-il de « bons » instruments de mesure?

1.2.2 Contexte prévu de la mesure en génie logiciel

Le recours à la mesure est explicitement mentionné dans la définition du génie logiciel selon la norme IEEE 610-12 : « l'application d'une approche systématique, disciplinée et quantifiable au développement, à l'opération et à la maintenance du logiciel, c'est-à-dire l'application du génie au logiciel » [52].

Il découle de cette définition de l'IEEE que l'application de l'ingénierie au logiciel requiert l'utilisation d'une approche quantitative et, de ce point de vue, la mesure est obligatoire. En effet, l'utilisation des mesures et des modèles quantitatifs a été indispensable en ingénierie tout au long de son histoire. Par exemple, Kirby *et al.*, en parlant des Égyptiens, mentionnent : *To place upright an obelisk of several hundred tons weight is an engineering feat that requires nicety of calculations and special equipment even in modern times* [81]. Cependant, en génie logiciel, il y a un manque de consensus sur la mesure. Cette affirmation est basée sur des commentaires collectés lors d'une étude réalisée pour identifier et pour atteindre un consensus sur les principes fondamentaux du génie logiciel [111].

D'autre part, Rombach [97] ajoute que la mesure du logiciel est un élément essentiel pour toute technologie mature qui peut aider à contribuer à mieux construire les modèles de qualité des produits et des processus logiciels. Selon Gile [45], l'avantage de mesurer d'un point de vue de l'ingénierie peut se résumer comme suit : « Permettre d'obtenir un moyen de contrôle des produits et des processus, déterminer la productivité, l'efficacité et la qualité d'un travail, obtenir la crédibilité et le respect du client avec la gestion ainsi qu'identifier où il faut apporter des améliorations ».

Pfleeger [92] ajoute également que la mesure devrait faire partie intégrante des activités de développement et de maintenance du logiciel et que la mesure est utilisée non seulement pour comprendre, pour contrôler et pour améliorer le développement des logiciels, mais aussi, pour déterminer la meilleure façon d'aider les praticiens et les chercheurs. Comme dans toutes les branches des sciences, il est nécessaire de passer par une succession de mesures; tel est le cas de la découverte des mesures de température et la progression vers des échelles, des outils et des techniques de plus en plus sophistiquées. Il en sera de même pour aboutir à la maturation des mesures de logiciels de la même façon que dans les autres disciplines du génie.

Lord Kelvin énonce que : « *when you can measure what you are speaking about, and express it in numbers, you know something about it, but when you cannot measure it, when you can not express it in numbers, your knowledge is of meagre and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the stage of science* » [79]. Cet énoncé de Kelvin met en relief l'aspect de mesure qui est nécessaire dans le progrès en sciences. Mais, selon Zuse: « *Software measurement alone does not guarantee an excellent software product. Software measurement is essential in areas, where empirical views, like quality, cost estimation, understandability, etc., are important (...) theory building is the major task of science and measurement in all areas is an important tool for that* » [114]. La mesure en elle-même ne permet pas d'améliorer le produit. Mais elle va servir comme un outil pour comprendre et pour contrôler différentes visions empiriques.

1.2.3 Problématique de la mesure en génie logiciel

Une problématique de toutes les sciences est l'attribution des caractéristiques empiriques de la signification d'une expérience. Dans le domaine des sciences empiriques, les théories réfèrent toujours à quelque chose qui peut être expérimentée. Toutefois, pour les mathématiques et la logique la vérité peut être établie indépendamment de la nature de n'importe quelle réalité. La fonctionnalité d'une expérience est ainsi considérée comme

un test final de vérification de ces états appelés « science ». Actuellement, la plupart des scientifiques s'entendent sur le fait que l'observation implique toujours certaines propositions, concepts, etc. En d'autres termes, elle est menée par la théorie.

La physique, qui est devenue une science expérimentale et mathématique, s'est développée grâce à l'amélioration des systèmes de mesure et à l'élaboration des théories mathématiques basée sur des lois et sur un ensemble cohérent des principes bien définis et bien formulés. Ces transformations des sciences physiques et chimiques en disciplines solides se sont faites sur plusieurs siècles. Les concepts de métrologie des sciences physiques sont appliqués avec succès dans les autres domaines scientifiques et d'ingénierie [1;14]. Le principal objectif dans toute mesure consiste à établir une valeur des attributs de certaines entités. Ces valeurs devraient être significatives : en se basant sur des méthodes de mesure adéquates ainsi que sur des instruments de mesure basés sur des références de mesure que l'on peut retrouver.

Comparativement, le génie logiciel est une discipline relativement récente et ne dispose pas encore de mesures rigoureuses faisant l'objet d'un consensus général. Malgré l'existence de plusieurs propositions dans la littérature, peu de travaux ont été effectués pour identifier et pour développer un consensus dans le domaine des connaissances des mesures en logiciel.

Dans la littérature du génie logiciel, les designs de la plupart des mesures proposées ne sont pas appuyés sur des approches vérifiables [102]. Ces mesures ne rencontrent pas l'ensemble des concepts de mesure dans les autres disciplines de l'ingénierie.

Pourquoi y a-t-il toujours des problèmes de mesure en génie logiciel? D'après Zuse [114], un des problèmes majeurs de mesure en génie logiciel est le scepticisme dans l'utilisation des valeurs numériques : dans certains cas, il y a même des difficultés dans l'interprétation des nombres et il y a un manque de sémantique des valeurs. Dans la communauté du génie logiciel, Zuse, Fenton, Pfleeger et autres notent une confusion

entre « métriques » et « nombres » dans un contexte général où souvent les nombres sont attribués à des hypothèses sans connaître leur évidence empirique [41;114]. Zuse considère les nombres comme des éléments d'une échelle, et ils devraient faire l'objet d'une correspondance homomorphe d'un système empirique avec un système relationnel numérique et vice versa. Cependant, cet aspect est souvent négligé dans la pratique des auteurs qui proposent des soi-disant « métriques » dans le domaine du logiciel.

1.2.4 Manque de consensus sur la place de la mesure en génie logiciel

Nous allons maintenant explorer l'absence de consensus dans cette communauté du génie logiciel sur la nécessité de mesurer. En effet, malgré une abondante littérature spécialisée dans la mesure en génie logiciel, elle est loin d'être unanimement utilisée et ne peut être qualifiée comme généralement acceptée.

Pour comprendre pourquoi la mesure n'a pas une telle reconnaissance, nous utilisons les commentaires collectés lors d'une étude conduite par des experts internationaux et des praticiens expérimentés dans le domaine du génie logiciel. Cette étude a été suggérée par le *Software Engineering Standards Executive Committee* de IEEE, le Delphi ainsi qu'une étude Web afin d'identifier et de développer un consensus sur les principes fondamentaux de la discipline [75]. Le résultat de cette étude dresse une liste de 15 principes fondamentaux (classés selon un ordre alphabétique) du génie logiciel (Tableau D). Cette liste de principes a permis d'illustrer et de clarifier les opinions de ces experts sur le rôle et sur l'importance de la mesure en génie logiciel.

Cette liste de principes, tirée des travaux de Bourque *et al.* [23], est fondée essentiellement sur les opinions des experts et des praticiens expérimentés dans le domaine qui ont participé aux différentes phases du projet.

Tableau I

Liste des principes fondamentaux candidats [23]

A.	Appliquer et utiliser des mesures quantitatives dans la prise de décision
B.	Construire en réutilisant et pour être réutilisé
C.	Contrôler la complexité par de multiples perspectives et de multiples niveaux
D.	Définir les artefacts logiciels de façon rigoureuse
E.	Mettre en place un processus logiciel flexible
F.	Implanter un processus rigoureux et l'améliorer continuellement
G.	Consacrer les ressources nécessaires à la compréhension du problème
H.	Gérer d'une manière aussi formelle que possible la qualité durant tout le cycle de vie
I.	Minimiser les interactions entre les composants logiciels
J.	Produire le logiciel par étapes
K.	Établir des objectifs de qualité pour chaque produit à livrer
L.	Comme le logiciel est, de par sa nature même, sujet au changement, il faut planifier et gérer ce changement
M.	Les compromis étant inhérents au génie logiciel, il faut les rendre explicites et les documenter
N.	Pour améliorer la conception, étudier les solutions antérieures à des problèmes similaires
O.	L'incertitude est inévitable en génie logiciel. Il faut identifier et gérer cette incertitude

Cependant, on constate que lors de la collection des commentaires et des opinions, il y avait, parmi ces experts, des controverses sur le thème de la mesure. Certains parmi ceux-ci ont argumenté que la mesure est fondamentale en ingénierie et que sans celle-ci il n'y a pas d'ingénierie en elle-même. Selon leur point de vue, la mesure devrait toujours être appliquée et, si nécessaire, qualifiée. D'autres experts ont par contre exprimé l'opinion que l'application et l'utilisation des mesures et des modèles quantitatifs en génie logiciel portent trop à controverse pour être universellement applicables : par exemple, toujours selon ces experts, la mesure du logiciel n'est pas toujours applicable à cause de son coût excessif ou à cause du niveau faible de maturité du domaine du génie logiciel. D'autres experts pensent aussi que certains ingénieurs logiciels peuvent fortement focaliser sur la mesure au détriment de leur meilleur jugement et que la mesure ne constitue qu'une sorte d'entrée au processus de prise de décision dans les projets logiciels [24;111].

Ces différences d'opinions parmi le groupe d'experts dans le domaine du génie logiciel sont préoccupantes et nécessitent d'être analysées plus en détail. Une analyse systématique des différences des opinions par Wolf [111] a permis de formuler plusieurs questions sur la mesure du logiciel. Ces questions représentent les problématiques qui doivent être adressées pour obtenir à terme une reconnaissance large et l'usage de la mesure en génie logiciel.

Il est à noter aussi que l'ensemble des commentaires collectés durant l'étude des principes fondamentaux du génie logiciel est considéré riche en étude car couvrant la discipline du génie logiciel dans son ensemble et provenant d'un corpus international de participants compétents (membres des comités du génie logiciel de l'IEEE *Computer Society* et des praticiens expérimentés). Ces commentaires peuvent être négatifs, positifs, critiques, représenter une opinion, dénoncer un chevauchement de principes, etc.

Pour analyser les commentaires reliés à la mesure, un cadre a été adopté dans [24;111] en se basant sur les sept raisons pour lesquelles il faut mesurer en génie logiciel. Les six premières raisons sont identifiées selon le document d'Oman *et al.* [90] (Mesurer pour la compréhension, l'expérimentation, le contrôle de projet, l'amélioration du processus, l'amélioration du produit, et pour faire des prévisions). Une autre raison (mesurer pour l'évaluation) a été identifiée en se basant sur le livre de Fenton *et al.* [41]. À partir de l'analyse des commentaires, plusieurs questions ont été soulevées dans chaque catégorie, raison pour laquelle il faut mesurer en génie logiciel.

Mesure pour la compréhension : pour une meilleure compréhension des activités de développement et de maintenance des logiciels, et pour établir un guide afin de prédire les futurs comportements.

Un seul commentaire a été identifié concernant cette raison mentionnant que : le recours à la mesure est un indice que les problèmes, les artefacts ou la technologie n'ont pas été bien compris. Ceci indique : soit qu'il est clair pour tout le monde que la mesure aide à

la compréhension des problèmes, des artefacts et des technologies, soit à l'inverse que la mesure n'aide pas en ce sens. Les questions dérivées de l'analyse de ce commentaire sont les suivantes :

- quel est le niveau de compréhension des problèmes, des artefacts et des technologies du génie logiciel?
- quelle est l'étendue à laquelle la mesure permet d'aider à comprendre les problèmes, les artefacts et les technologies du génie logiciel?
- quels sont les aspects de ces entités du logiciel (exemple : problèmes, artefacts et technologies) les plus importants à mesurer?
- quel poids faut-il donner à chacun de ces aspects?

Mesure pour l'évaluation : pour comprendre ce qui existe maintenant ou ce qui s'est déroulé dans le passé. Les commentaires reliés à cette raison concernent l'utilisation de l'évaluation pour minimiser l'incertitude des projets en génie logiciel. Certaines questions dérivées de l'analyse de ces commentaires sont les suivantes :

- la discipline du génie logiciel possède-t-elle un plus haut niveau d'incertitude en comparaison avec les autres disciplines du génie ou simplement parce que les autres disciplines ont appris à mieux contrôler leurs incertitudes à travers le temps?
- l'utilisation de la mesure dans certaines activités de logiciel (telles que les études de faisabilité, la définition d'une architecture, l'identification et la réduction des risques, le planning et l'estimation du coût et l'effort) pourrait-elle minimiser réellement le niveau d'incertitude dans les projets du génie logiciel?

Mesure pour l'expérimentation : pour améliorer les méthodes de développement de logiciels, pour mieux comprendre les effets des technologies diverses, pour identifier les champs nécessitant le plus d'amélioration et pour tester des hypothèses et en bâtir de nouvelles.

Aucun commentaire sur cette raison n'a été identifié parmi l'ensemble des concepts collectés. Cette absence est peut-être due au fait que l'expérimentation n'est pas présente à un degré suffisant en génie logiciel, ou encore parce qu'on s'intéresse à des analyses théoriques qui ne sont pas appliquées par la suite dans la pratique.

Mesure pour le contrôle de projet : pour faciliter l'évaluation de l'état d'un projet et la prédiction des événements futurs dans un projet. Un participant affirme que chaque projet devrait planifier et spécifier quantitativement ses stratégies de fiabilité en se basant sur ses objectif(s) de fiabilité. Un autre affirme que chaque projet logiciel devrait fixer un objectif ou des objectifs de fiabilité pour son produit livrable, et qu'il devrait être estimé à différentes étapes du projet et comparé aux résultats obtenus avec les objectif(s) du projet. Les questions dérivées de l'analyse de ces commentaires sont les suivantes :

- est-il réellement utile d'utiliser des mesures quantitatives dans la planification et le contrôle du projet?
- est-ce que le contrôle n'implique pas la nécessité de mesure?
- est-ce que la mesure quantitative est nécessaire pour le contrôle de tous les types des projets en génie logiciel?

Mesure pour l'amélioration du processus : pour améliorer la qualité des processus en génie logiciel, par une meilleure évaluation et par une meilleure compréhension des impacts des changements qui lui sont apportés. Un participant suggère que l'utilisation des données quantitatives devrait non seulement guider la gestion des projets mais également calibrer les progrès dans l'amélioration du processus au sein de l'organisation. Un autre affirme que l'activité de mesure des progrès et la performance des processus en génie logiciel sont essentielles aux améliorations nécessaires pour satisfaire la demande de l'industrie. Les questions dérivées sont les suivantes :

- les mesures utilisées pour l'amélioration des processus en génie logiciel sont-elles valides?

- y a t-il des mesures permettant de contrôler l'amélioration apportée aux processus grâce à l'utilisation de certaines mesures?

Mesure pour l'amélioration du produit : pour fournir les raisons sur les façons dont les processus, les produits, les ressources, les méthodes et les technologies de développement de logiciels sont liés; en résumé, pour permettre l'amélioration du produit. Un participant affirme que l'évaluation de la qualité du logiciel doit être scientifique et quantitative. Cependant, les résultats doivent être utiles et les coûts ne doivent pas dépasser les avantages. La question dérivée est la suivante : Faut-il à tout prix viser à utiliser des mesures permettant l'amélioration du produit, ou est-il important de calculer le retour sur les investissements obtenus à partir de l'utilisation de ces mesures?

Mesure pour la prédiction : pour de nouvelles activités, il est nécessaire de prévoir l'effort requis et les coûts de développement ainsi que d'autres facteurs. La mesure fournit un guide pour prédire ces activités. Attendre jusqu'à la fin du projet pour mesurer les attributs de coût et de temps est clairement irréaliste. Quelques commentaires suggèrent que dans certains projets, il est important de faire des estimations à l'avance du coût et de la durée, car il est simplement inacceptable d'attendre jusqu'à la fin du projet pour connaître ces valeurs. Il est aussi souvent nécessaire de connaître si un produit logiciel donné peut atteindre l'objectif de qualité requis, tel que la fiabilité. Il est ainsi important d'avoir des mesures pour pouvoir prédire ce qui va se passer si le projet est réalisé. Les questions dérivées sont les suivantes :

- dans les cas de prédictions, à quoi faut-il le plus se fier : aux mesures prévues au départ ou aux mesures prélevées en cours de projet?
- est-ce que l'utilisation de mesures prises pour des projets antérieurs peut permettre de faire de meilleures prédictions sur des projets à venir?

Par ailleurs, certains commentaires ne peuvent être classés selon le cadre défini, car ceux-ci sont reliés à la mesure d'un point de vue général de mesure. Pour cela, une raison additionnelle a été ajoutée « mesure en général ».

Mesure en général : la mesure joue un rôle important en ingénierie; par conséquent, selon certains commentaires des experts participants, si la mesure n'est pas utilisée systématiquement en génie logiciel, la discipline ne peut être reconnue comme une discipline de l'ingénierie. Cependant, un autre participant affirme que malgré l'importance de la mesure, les pratiques en génie logiciel demeurent encore une combinaison d'art et de science. Les questions dérivées sont les suivantes :

- est-ce que la mesure joue un rôle primordial en génie logiciel?
- si la mesure ne joue pas un rôle primordial en génie logiciel, cela élimine-t-il les chances de cette discipline d'être reconnue comme étant une branche du génie?

Cette analyse des opinions des experts et des praticiens expérimentés sur l'utilisation de la mesure et des modèles quantitatifs en génie logiciel illustre le manque de consensus sur ce thème.

Ce manque de consensus parmi les experts et les praticiens expérimentés est préoccupant d'autant plus que :

- la définition du génie logiciel selon IEEE [52] nécessite explicitement l'application d'une approche quantifiable;
- l'expression « génie logiciel » est largement utilisée dans la recherche et la pratique, et l'utilisation extensive des mesures et des modèles quantitatifs est donnée dans les disciplines reconnues de l'ingénierie.

Ces questions soulèvent particulièrement le rôle et l'importance de la mesure dans la reconnaissance du génie logiciel en tant qu'une discipline du génie, ainsi que les problèmes non encore résolus de la mesure en génie logiciel.

1.2.5 Théorie de la mesure et métrologie en génie logiciel

Comparativement aux autres disciplines du génie (tels que le génie électrique, civil, chimique, etc.), en génie logiciel la plupart des chercheurs spécialisés dans la mesure du logiciel ont utilisé les concepts de la théorie de mesure pour l'évaluation des mesures du logiciel que ce soit pour le contrôle et la prédiction des attributs du logiciel tels que par exemple l'effort de développement et la productivité du personnel. Cependant, l'application de la théorie de la mesure en génie logiciel ne se concrétise pas facilement comme dans les autres domaines du génie et les praticiens et chercheurs ont de la difficulté à utiliser les mesures du logiciel proposées pour évaluer et contrôler les différents attributs du logiciel.

Dans les autres disciplines de l'ingénierie, les méthodes de mesure, les instruments et les techniques de mesure ont été établis depuis longtemps : ils doivent satisfaire à l'ensemble des concepts de la métrologie (VIM) et les critères de qualité des résultats de mesure doivent être connus, telles que par exemple l'exactitude, la répétitivité, la reproductibilité, etc. L'application des concepts de la métrologie classique en génie logiciel pourrait aider à la constitution de cette structure dans le sous-domaine des normes pertinentes à la mesure.

Shaw [103] décrit dans son article *Prospects for an Engineering Discipline of Software*, l'évolution d'une discipline vers le génie comme le montre la Figure 1.

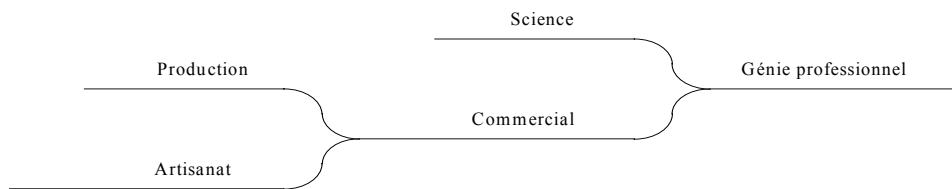


Figure 1 Évolution d'une discipline de génie [103]

La Figure 1 présente l'évolution d'une discipline de génie; les lignes inférieures reflètent le parcours que suit la technologie. Les lignes supérieures montrent l'entrée des habiletés de production et de connaissances scientifiques, contribuant à de nouvelles possibilités au niveau des pratiques de l'ingénierie. D'après Shaw, le domaine du génie logiciel aurait été au début des années 1990 entre l'artisanat et le commercial, dépendamment des aspects qu'il prend en compte, et que ce domaine constitue un objectif qui peut être atteint selon des contraintes requises. Ces contraintes consistent à sélectionner à court terme un ensemble approprié de contributions purement empiriques et pragmatiques, qui aideront à stabiliser les pratiques commerciales et à investir dans des efforts à long terme pour développer et rendre disponible des contributions scientifiques de base [103].

1.2.6 Les types d'échelle de mesure en génie logiciel

D'après Fenton [41], une mesure directe d'un attribut est l'assignation de la correspondance d'un système de relation empirique en un système de relation numérique, de telle sorte que les nombres obtenus représentent les caractéristiques du monde réel. Cependant, les mises en correspondance des mesures ne sont pas toujours les mêmes et peuvent dépendre étroitement de la nature de l'analyse effectuée. La notion de type d'échelle est ainsi importante car elle permet de mettre en évidence les propriétés empiriques associées à l'attribut mesuré. La communauté du génie logiciel utilise souvent les cinq types d'échelle proposés par Stevens en 1946 : nominale, ordinale, intervalle, ratio et absolue [106]. Le Tableau II montre la classification des types d'échelle, l'ensemble des transformations admissibles qui peuvent être appliquées à un type d'échelle donné et des exemples. Les propriétés des échelles de mesure sont hiérarchiques. Chaque échelle possède les propriétés de l'échelle d'un niveau plus bas. Ainsi, l'échelle absolue possède toutes les propriétés des échelles précédentes.

Tableau II
Types d'échelle

Type d'échelle	Transformations admissibles, G	Exemples
Nominale	G unique, est une correspondance 1-1	Étiquettes, couleur, forme
Ordinale	G est une fonction monotone croissante ($x \geq y \Rightarrow G(x) \geq G(y)$)	Préférence, difficulté, qualité, intelligence
Intervalle	$G(x) = a x + b, a > 0$	Température (Fahrenheit, Celsius), temps du calendrier
Ratio	$G(x) = a x, a > 0$	Longueur, masse, distance température absolue (Kelvin)
Absolue	G est l'identité $G(x) = x$	Comptage d'entités

Les types d'échelle sont ainsi définis par les transformations mathématiques admissibles et décrits par les propriétés empiriques de l'attribut mesuré (les opérations statistiques applicables sur les valeurs de l'échelle, les unités associées aux échelles, le type des affirmations significatives dans l'échelle). Le type d'échelle nous indique le type d'opérations qu'on peut accomplir. Par exemple, on peut calculer les moyennes des mesures de type d'échelle ratio, mais pas celles de type ordinal.

La plupart des attributs de logiciels sont mesurés sur une échelle de type ordinal. Exemple : dans la méthode des points de fonction (PF) de mesure de la taille fonctionnelle du logiciel, les cinq éléments de base de la méthode (entrées, sorties, requêtes, fichiers et interfaces) sont évalués en utilisant une échelle ordinale composée des critères simple, moyen et complexe. Chacun de ces critères est représenté par un intervalle selon le nombre de fichiers et d'items référencés (Tableau III).

La taille fonctionnelle est tout d'abord calculée en assignant des poids à chaque fonction individuelle pour obtenir les points de fonction non ajustés (UFP), par la suite la valeur du facteur d'ajustement (VAF) est calculée en utilisant le coefficient prédéfini des

caractéristiques d'un système général (GSC) pour obtenir les points de fonction ajustés (AFP). Il y a 14 GSC avec un VAF = $\pm 35\%$ dans Albrecht'83 qui sont mesurés sur une échelle ordinale.

Tableau III

Matrice de complexité fonctionnelle (IFPUG –Guide 85)

Enregistrements logiques (RET)	Types d'éléments de données (DET)		
	1 à19	20 à 50	51+
1	Simple	Simple	Moyen
2 à5	Simple	Moyen	Complexe
6+	Moyen	Complexe	Complexe

Le problème avec la méthode FPA a été identifié par Abran *et al.* [10], en précisant la validité des opérations mathématiques et les transformations implicites lors du passage d'une échelle à l'autre. Par exemple, il est possible de passer d'une échelle ordinale à une échelle nominale. Mais la transformation d'une échelle ordinale à une échelle ratio n'est pas permise et conduit à une perte d'information.

D'autre part, le positionnement et la correspondance des résultats de DET et RET selon des rangs (Tableau III) ainsi que les résultats obtenus ne peuvent pas être additionnés, mais seulement ordonnés. Ils ne peuvent pas être qualifiés comme une échelle ratio ou un intervalle, mais ils correspondent plutôt à une échelle ordinale. Ainsi, la flexibilité mathématique a été perdue à travers ce changement de type d'échelle. Même si elles semblent intuitivement valides, les opérations mathématiques pour chaque type d'échelle ne sont pas respectées [10;11] de même que pour la « mesure » : le nombre de complexité cyclomatique de McCabe [84]. McCabe définit l'équation (1) en se basant sur la théorie des graphes.

$$v(g) = e - n + 2 \quad (1)$$

où, il y a e arêtes et n nœuds.

L'identification des unités de mesure dans la définition du nombre cyclomatique de McCabe n'a pas été explicitement documentée. Cette constatation a été identifiée par Abran *et al.* [7]. En effet, l'addition et la soustraction des différents types d'unités sont explicitement invalides, dans un sens numérique où les différents types d'unités sont impliqués. Ceci ne peut être valide que si ces unités peuvent être transposées à un niveau plus élevé d'abstraction avec un autre type d'attribut plus générique avec une unité correspondante. De plus, le nombre 2 de l'équation (1) est dérivé d'une addition invalide des différents types d'unité (nombre d'arrête virtuel = 1 et nombre de composants connectés = 1).

McCabe a proposé ainsi une mesure de la complexité du programme : complexité cyclomatique basée sur l'équation (1), et qu'il interprète comme étant la quantité de la logique de décisions dans un seul module du logiciel. En d'autres termes, un module de logiciel est représenté par un graphe de flux de contrôle et sa complexité cyclomatique est définie pour chaque module par l'équation (1) [110]. Cependant, McCabe ne définit pas la complexité. En plus du manque de définition de la complexité et de la caractérisation de l'attribut lui-même, il y a un problème d'identification du type d'échelle correspondant.

Zuse a proposé une démonstration du type d'échelle associée avec le nombre cyclomatique [114]. Il a suggéré une opération de concaténation des diagrammes de flux afin de prouver l'échelle ratio. Cependant, l'utilisation du nombre cyclomatique comme une échelle ratio nécessite de travailler avec des opérations additives spécifiées dans Zuse [114].

Ainsi, l'analyse du nombre de complexité cyclomatique de McCabe selon Abran *et al.* [7] a montré que le nombre cyclomatique $v(g)$ est défini en termes d'unité (nombre de cycles) et qu'il représente une quantification qualifiée par un « comptage », opération qui produit un nombre entier sur une échelle de type ratio. Cependant, l'utilisation de ce

nombre ne semble pas être interprétée sur une échelle ratio, mais plutôt sur une échelle intervalle non-spécifiée, telle que le type d'échelle exponentielle; ce type d'interprétation implique un manque de transparence si les résultats de mesure sont utilisés dans un contexte d'estimation du taux d'erreur ou de l'effort de test : il y a alors une transposition de l'échelle ratio du nombre cyclomatique par les utilisateurs du nombre de McCabe selon leur propre échelle d'interprétation, et leur perception intuitive de la complexité du diagramme de flux (qui est associé au nombre de McCabe) [7].

1.3 Problématique et objectifs

Depuis quelques décennies, plusieurs propositions dites de « mesures » ont été introduites dans le domaine du génie logiciel pour analyser la qualité du produit logiciel et pour utiliser les connaissances acquises dans le but de contrôler le processus de développement et de maintenance des logiciels. Toutefois, ces propositions dites de « mesures » se limitent à un sous-ensemble de l'ensemble des concepts pertinents aux mesures, soit la description générale d'algorithmes sans système de référence, sans instrumentation de mesure et sans notion de protocoles de mesure.

L'expression généralement utilisée dans la littérature du génie logiciel est celle d'un concept inédit en sciences et en ingénierie et mal défini de « métriques » de logiciels, à la place du terme plus précis et classique de « mesures » de logiciels. Il y a une confusion entre les deux termes. En effet, ces « métriques » ont des objectifs divers et ne sont pas nécessairement reliées au même concept de mesure; certaines de ces « métriques » ont comme objectif de mesurer un attribut (exemples : fonctionnalité, fiabilité, maintenabilité [71-73]). D'autres s'intéressent à la modélisation d'un ensemble de relations entre de multiples attributs et ces relations sont fortement influencées par des contextes empiriques très différents (exemple : modèle COCOMO [22]).

Il s'ensuit que les résultats obtenus avec ces « métriques » ne répondent pas en général à la majorité des critères de qualité requis pour les instruments de mesure, comme par

exemple l'exactitude, la répétitivité, la reproductibilité, le seuil de tolérance, la convertibilité, etc.

La mesure joue un rôle important dans la compréhension et dans le contrôle des pratiques de développement et de maintenance du logiciel. Cependant, dans son état actuel en génie logiciel, la mesure présente plusieurs problèmes : il est nécessaire d'améliorer la situation, en particulier de structurer le fondement des mesures utilisées afin de nous assurer de la fiabilité des résultats obtenus, et de permettre à l'industrie de choisir parmi les mesures proposées. En effet, beaucoup d'alternatives de mesure ont été proposées dans la littérature du génie logiciel pour quantifier plusieurs sujets d'intérêt pour l'industrie mais avec peu de consensus qui se sont concrétisés par l'acceptation des mesures de logiciels comme normes ISO.

Cette diversité dans la littérature et le manque de normalisation est typique des technologies émergentes qui se trouvent dans la phase immature de leur cycle de vie. À ce niveau de début de l'évolution technologique, les innovateurs et les chercheurs dans le domaine ont leurs propres visions sur les caractéristiques attendues dans les solutions proposées et utilisent leurs propres terminologies selon leurs propres perspectives. Pendant cette phase d'émergence, ces technologies ne sont ni robustes pour l'industrie, ni totalement fonctionnelles, et elles nécessitent des ajustements ou des améliorations pour rencontrer l'infrastructure existante des technologies les plus matures. Le contexte du génie logiciel rencontre les caractéristiques des technologies immatures qui sont en voie d'émergence. Dans ce contexte, plusieurs mesures de logiciel sont basées sur des approches intuitives non-vérifiées, sont mal définies par leurs auteurs et ne sont pas basées sur des fondements vérifiables. Il existe ainsi plusieurs mesures portant sur des objectifs et des objets différents, telles que les mesures de la complexité des logiciels et des éléments orientés objets, etc.

Pour aider l'industrie à évaluer la qualité des mesures proposées, quelques auteurs en génie logiciel (Kitchenham [82], Zuse [114], Fenton [39;41], Pfleeger [92], El-Emam

[38], Schneidewind [100;101], etc.) ont proposé diverses approches de « validation » de ces mesures, mais selon des perspectives distinctes. Chaque auteur a proposé un ensemble différent de critères de validation selon sa propre perspective et souvent avec exactement le même vocabulaire : or, ces auteurs établissent toutefois des étapes distinctes au processus de mesure [6]. Il en découle ainsi une ambiguïté au niveau des concepts de validation des mesures, concepts qui semblent être incohérents entre les auteurs. Ceci amène une certaine confusion chez les chercheurs et les praticiens quant au choix à prendre entre les diverses approches de validation proposées et les critères pertinents, plutôt que de les aider. D'après Abran, « *If software engineering is to mature into a recognized engineering discipline, it needs to be supported by measures, measurement methods and well tested descriptive and quantitative model* » [1]. Pour que cette discipline devienne mature, elle doit pouvoir compter sur des mesures dont le design et l'instrumentation de mesure se comparent favorablement aux outils des mesures disponibles dans les autres disciplines de l'ingénierie. Bien entendu, tant le domaine du génie logiciel, que son sous-domaine de mesure, sont encore immatures et présentent des problèmes au niveau théorique et pratique.

Pour pallier à cette problématique générale, un cadre de vérification d'une mesure est nécessaire. Un tel cadre pourra aider les chercheurs et les praticiens à comprendre comment vérifier une mesure en-soi (au niveau de son design), comment évaluer les différentes mesures proposées en génie logiciel et comment vérifier le design des instruments de mesure reliés à l'application pratique d'une méthode de mesure donnée. Nos objectifs consistent à :

- introduire les concepts du VIM pour comprendre les fondements des attributs de mesure de logiciels;
- proposer un cadre de vérification basé sur le VIM pour analyser le design des mesures et le design des instruments de mesure de logiciels;
- identifier les forces et les faiblesses des mesures de logiciels;

- proposer un cadre intégré pour identifier les critères pertinents de vérification et unifier les différentes perspectives de mesure pour aboutir à un consensus et à la progression du génie logiciel dans sa maturité.

Dans cette thèse, nous proposons un cadre intégré de vérification des méthodes de mesure en se basant principalement sur les concepts de mesure identifiés dans le VIM, le processus de Abran *et al.*, le cadre de Kitchenham. Ce cadre devra rencontrer les critères usuels du domaine de l'instrumentation de mesures en science et en génie et tenir compte des caractéristiques spécifiques du produit intellectuel qu'est un logiciel afin de permettre d'aider l'industrie à évaluer la qualité des mesures proposées.

1.4 Choix d'une approche de recherche : la métrologie

Dans la littérature du génie logiciel, certains auteurs tels que Curtis [33], Fenton [41], Bieman [20], Schneidewind [100], Henderson [47], Melton [86], Zuse [114]) ont principalement utilisé, pour analyser les mesures proposées, les propriétés théoriques des mesures, concrétisées par la manipulation des chiffres par des formules mathématiques et l'assignation des nombres. D'autres auteurs, pour concevoir leurs mesures du logiciel, se sont généralement basés sur l'interprétation qualitative des nombres obtenus et sur des approches intuitives et des hypothèses de cause à effet qui souvent n'ont pas été vérifiées expérimentalement.

Bien que la définition des concepts de mesure soit une étape importante pour l'élaboration du mesurage, la plupart de ces auteurs se basent sur l'aspect utilisation des mesures et n'abordent que peu les thèmes de mesure identifiés en métrologie (VIM). Le VIM présente l'ensemble des concepts pour le design, l'évaluation et l'utilisation des instruments de mesure et des processus de mesure dans toute discipline de l'ingénierie. Toutefois, en génie logiciel, les concepts du VIM se référant à la métrologie n'ont pas été utilisés par les chercheurs du domaine de la mesure.

Pour faciliter la compréhension et l'utilisation du domaine de connaissance de la mesure ISO du VIM, nous proposons de développer des modèles du VIM. Dans ce travail de recherche, nous explorerons ainsi les concepts de la métrologie afin d'en faire bénéficier la thématique de mesure en génie logiciel. Nous utiliserons aussi bien le document ISO du VIM [57] que d'autres outils analytiques de vérification de mesure pour pouvoir identifier, comprendre, contrôler et améliorer les fondements de mesure en génie logiciel.

1.4.1 Problèmes de mesure au niveau pratique

Pour obtenir un résultat de mesure cohérent, il est nécessaire d'appliquer une méthode de mesure consistante ainsi que de vérifier les concepts reliés à l'instrument de mesure. La mesure de logiciels est considérée comme appartenant au domaine expérimental du génie logiciel : elle doit nous permettre de comprendre les forces et les faiblesses des méthodes et des dispositifs à utiliser pour un objectif spécifique d'un projet logiciel spécifique. Pour concevoir des mesures de logiciels et pour les vérifier comme dans les autres domaines de l'ingénierie, un cadre pour la vérification d'une mesure est requis. Dans le cadre de notre recherche, un cadre de référence sera proposé pour permettre de positionner les éléments de design des méthodes de mesure du logiciel et ceux des instruments reliés à l'application pratique de ces méthodes de mesure. Nous proposerons ainsi l'utilisation de ce cadre pour l'analyse des normes ISO et des domaines de connaissance de la mesure en logiciel (ISO 19759 : SWEBOK [68], ISO 9126 [60;71-73], ISO 14143 [62-66], ISO15939 [67], ISO19761 : COSMIC-FFP [69]). Cette solution permettra ainsi de déterminer les critères requis pour améliorer tant les méthodes de mesure que les normes ISO actuelles (et pour bâtir les futures normes) de mesure du logiciel.

En dépit du fait que la mesure soit considérée comme un concept nécessaire dans toute discipline du génie, la mesure est toujours controversée en génie logiciel. Cette constatation a été notée particulièrement lors de l'analyse des commentaires recueillis

dans un projet visant à identifier et à obtenir un consensus sur les principes fondamentaux de la discipline parmi la communauté internationale du génie logiciel [23]. En effet, cette analyse a montré que les membres de cette communauté ne s'entendaient pas sur l'importance à accorder à la mesure. De plus, en génie logiciel la mesure n'est pas aussi utilisée que dans les autres disciplines du génie. Il y a un manque de consensus sur la mesure en génie logiciel :

- d'un côté, il y a ceux qui pensent que la mesure doit faire partie intégrante du génie logiciel et que, de ce fait, elle représente un concept important;
- d'un autre côté, il y a ceux qui pensent que la mesure aura effectivement un rôle à jouer en génie logiciel mais que, dans son état actuel, il vaut mieux ne pas trop l'utiliser. De plus, dans les cas où la mesure est effectivement utilisée en génie logiciel, le processus correspondant semble manquer de clarté et de rigueur.

Il y a plusieurs chercheurs qui ont traité la problématique de la mesure de logiciels selon différentes perspectives [19;40;41;82;88;94;100;114] avec peu de discussion sur les instruments de mesure, ni sur l'étude de la qualité des instruments quand mentionnés. Les propositions sont souvent décrites au niveau utilisation des mesures. Par exemple :

- Kitchenham *et al.* recommandent un cadre préliminaire de validation de mesure de logiciels [82];
- Schneidewind propose un corpus de connaissance qui consiste en une liste des thèmes (*issues*) de mesure de la qualité du logiciel [101];
- Fenton *et al.* suggèrent qu'une mesure valide doit nécessairement obéir aux conditions de représentation de la théorie de mesure de telle sorte que la compréhension intuitive de certains attributs est préservée lors de sa mise en correspondance dans un système de relation numérique [41];
- Zuse définit les mesures de logiciels par la mise en correspondance des propriétés empiriques aux propriétés numériques par un homomorphisme [114].

La plupart de ces auteurs s'intéressent particulièrement au domaine théorique des connaissances de la mesure. Cependant, les aspects théoriques et pratiques de la mesure sont importants sans avoir à privilégier l'un de l'autre ou chercher à placer l'un ou l'autre en position dominante. Comment peut-on vérifier que les mesures utilisées dans le contexte industriel soient robustes?

L'industrie s'intéresse généralement à l'utilisation des mesures afin d'optimiser la qualité du logiciel, le délai de livraison, améliorer les relations avec les clients, etc. sans tenir compte de la vérification de la mesure en-soi. En effet, les gestionnaires préfèrent manipuler plusieurs chiffres pour avoir une flexibilité au niveau de la gestion des projets logiciels sans tenir compte des fondements sur lesquels les chiffres ont été obtenus. Les approches les plus utilisées de quantification des aspects de qualité du logiciel relatifs au produit, au processus et aux ressources sont les mesures de code de Halstead, la densité-de-défauts de McCabe, les méthodes des points de fonction d'Albrecht, le modèle COCOMO [22], etc. Une question qui se pose : les approches les plus utilisées sont-elles nécessairement les « bonnes » approches de mesure? Et, quels sont les critères de sélection parmi ces approches pour prendre les meilleures décisions?

1.4.2 Contexte du projet de recherche

La problématique générale de la mesure du logiciel est l'une des principales problématiques explorée par les chercheurs du Laboratoire de recherche en génie logiciel de l'École de technologie supérieure - Université du Québec. Les intérêts de ces chercheurs incluent les mesures de logiciels (particulièrement les mesures de la taille fonctionnelle des logiciels), la mesure de la maintenance, les méthodes de vérification des mesures et les programmes de mesure. Au cours des cinq dernières années, les chercheurs de ce laboratoire ont publié plusieurs résultats de recherche transposant au domaine du génie logiciel quelques-uns des concepts classiques de la métrologie, et ce tant pour le design d'une méthode de mesure du logiciel que pour l'analyse de la qualité

des instruments de mesure qui en découle. Ceci a été appliqué en particulier dans le design de la nouvelle méthode de mesure COSMIC-FFP (ISO 19761) [4;69].

Notre projet de recherche constitue le prolongement de ces travaux. Plutôt que d'utiliser uniquement un sous-ensemble des concepts de métrologie, nous utiliserons l'ensemble des concepts de mesure du domaine de la métrologie (VIM) avec les éléments du processus de mesure comme un outil d'analyse, soit pour évaluer les mesures proposées dans le domaine du génie logiciel, soit pour en développer de nouvelles.

1.5 Avertissement terminologique : vérification et validation

Le terme « validation » est fréquemment utilisé dans les textes de beaucoup de chercheurs, mais sans faire de distinction entre le concept de « vérification » et celui de « validation ». La vérification porte en général sur l'évaluation par rapport à certains critères documentés à l'avance, tandis que celui de la validation porte sur l'évaluation par rapport aux besoins.

Selon la norme ISO 14143-3 [64] sur la vérification des méthodes de mesure de la taille fonctionnelle, l'objectif de la vérification est d'obtenir une preuve objective de l'étendue avec laquelle une méthode de mesure démontre certaines propriétés de performance. Le niveau d'acceptabilité peut être dépendant du contexte et de l'objectif du responsable de vérification : par exemple, un niveau peut être acceptable pour un objectif mais inacceptable pour un autre objectif. Dans ce contexte, la vérification dépend du choix des critères déjà définis et de l'objectif de mesure. La vérification ainsi faite conduit au degré qui rencontre ces critères. Nous illustrons les éléments de vérification d'une méthode de mesure du logiciel dans le Tableau IV qui suit :

Tableau IV

Éléments de vérification d'une méthode de mesure du logiciel

<i>Méthode de mesure du logiciel :</i>	Méthode basée sur des critères explicatifs
<i>Principaux concepts du domaine de connaissance de la mesure (VIM) :</i>	Concepts déjà acceptés comme bien établis par la communauté scientifique et qu'il faut combiner aux critères de la méthode de mesure du logiciel pour sa vérification
<i>Facteurs supplémentaires :</i>	Facteurs supplémentaires ajoutés à la méthode qui portent sur le niveau d'expertise de l'utilisateur et sur les procédés utilisés lors de la vérification
<i>Résultats :</i>	Évaluation par rapport aux éléments mentionnés

Dans le contexte de notre recherche et pour des raisons de clarification, nous utiliserons le terme « vérification » et non pas le terme « validation ».

1.6 Structure du rapport

Le présent document est composé de dix chapitres. Dans le chapitre suivant, nous présentons les différents outils analytiques de vérification de mesure dans les domaines du génie et du génie logiciel.

Le chapitre 3 est consacré à la présentation des étapes de la méthodologie de recherche que nous utiliserons pour atteindre nos objectifs.

Dans le chapitre 4, nous allons établir des modèles pour la vérification de mesure selon la perspective de la métrologie. Ces modèles seront déterminés lors de l'analyse du document ISO du VIM. Nous allons ensuite effectuer une analyse détaillée des concepts du VIM. Le but de cette analyse est d'identifier les éléments en liaison avec les concepts de mesure qui ont été mentionnés particulièrement dans les notes et les exemples du VIM, et de chercher les niveaux d'abstraction de ces concepts.

Dans le chapitre 5, le cadre de référence pour la vérification de mesure qui est introduit, permet de fusionner les concepts de la métrologie VIM avec les concepts du processus

de mesure de logiciels d'Abran *et al.* Ce cadre présente, en plus des concepts identifiés pour la mesure de logiciels, les concepts supplémentaires qui proviennent de la métrologie. Ce cadre va servir comme outil analytique pour analyser et vérifier les propositions de mesure en génie logiciel.

Dans le chapitre 6, nous abordons l'analyse des normes ISO (ISO 9126-1 à 4 et ISO 14143-1 à 5) traitant de la mesure en génie logiciel, en utilisant le cadre de référence. Nous analysons également la méthode COSMIC-FFP : ISO 19761 en utilisant le modèle du VIM.

Dans le chapitre 7, nous analysons les différents types de mesure utilisés dans le domaine de la gestion du logiciel, en particulier la norme ISO 15939 et le modèle COCOMO II.

Le chapitre 8 traite de l'analyse du cadre proposé par Kitchenham *et al.* [82] pour clarifier principalement les concepts de mesure.

Le chapitre 9 présente l'analyse du corpus de connaissance de la mesure en génie logiciel pour examiner les fondements de mesure dans SWEBOK.

Le dernier chapitre présente une proposition d'un cadre intégré de vérification de mesure qui est basé principalement sur le modèle du VIM, le cadre de processus de mesure de Abran *et al.* et le cadre de vérification proposé par Kitchenham *et al.* Ainsi, nous présentons les différentes étapes de ce cadre allant de la conception d'une méthode de mesure jusqu'à l'obtention des résultats qu'elle peut produire. Ensuite, nous identifions les critères de vérification d'une mesure en génie logiciel.

La conclusion générale dresse un bilan des travaux entrepris dans cette thèse, nos contributions principales, et enfin nos perspectives dans le domaine de la mesure en génie logiciel.

CHAPITRE 2

LES OUTILS ANALYTIQUES DE VÉRIFICATION DE MESURE

2.1 Introduction

La « mesure du logiciel » est souvent proposée comme un outil d'analyse de mesure dans les études empiriques en génie logiciel. Ce domaine de la « mesure du logiciel » a été souvent évoqué d'une perspective relative aux concepts de la « théorie de mesure ». Cependant, dans toute discipline de l'ingénierie, c'est plutôt le domaine de connaissance relative à la métrologie qui représente la base pour le développement et l'utilisation des instruments de mesure et des processus de mesure.

Dans la littérature du génie logiciel, le domaine traité de la mesure concerne principalement l'utilisation des résultats de mesure dans des modèles d'évaluation ou de prédiction [5;29;34;36;37;41;42;49;114]. Malgré le nombre élevé des mesures proposées, il y a très peu de discussion sur les instruments de mesure par rapport aux disciplines traditionnelles de l'ingénierie. Ceci nous indique qu'un nombre de concepts et de sous-concepts de la métrologie n'ont pas été significativement discutés et utilisés dans la littérature de « mesure du logiciel ». En sciences et en génie, les instruments de mesure sont des éléments clés nécessaires aux études empiriques; ceci met en relief une faiblesse dans les études empiriques actuelles en génie logiciel, et fournit une indication que des améliorations provenant de la métrologie peuvent contribuer à renforcer les études empiriques ultérieures en génie logiciel.

Dans ce chapitre, nous proposons une revue de la littérature sur les différents outils analytiques de vérification d'une méthode de mesure. Pour chaque outil, nous présentons les avantages et les inconvénients.

2.2 Le domaine de connaissance de la mesure du VIM

Tandis que le document ISO du VIM [57] est largement reconnu dans le domaine de la métrologie, il est pratiquement inconnu par les praticiens et les chercheurs en génie logiciel. En effet, plusieurs auteurs en mesure des logiciels [25;29;38;41;49;82;112] ont abordé le problème de la mesure en se basant sur leurs intuitions et sur des analyses de cause à effet qui ne sont pas fondées sur le domaine de connaissance de la mesure (métrologie).

Le VIM représente un consensus international sur une terminologie cohérente et commune pour l'ensemble des concepts de métrologie. Le document ISO du VIM [57] ne suit pas la structure usuelle des dictionnaires ou des glossaires par lesquels tous les termes sont présentés et documentés sur la base de leur ordre alphabétique : dans le VIM, les termes sont tout d'abord subdivisés en un ensemble de six catégories organisées par des chapitres distincts (voir Tableau V).

Tableau V

Catégorie des termes dans VIM'93

Catégories	Nombre des termes = 120
Grandeurs et unités	22
Mesurages	9
Résultats de mesure	16
Instruments de mesure	31
Caractéristiques des instruments de mesure	28
Étalon	14

Dans chaque chapitre du VIM, les termes sont présentés dans « un ordre approximatif de complexité croissante » [7 : ligne 7, page 29], et leur définition dans un format de texte. Ceci rend la compréhension générale difficile, puisqu'il n'y a aucune correspondance directe fournie de l'ensemble complet des relations entre les 120 termes reliés aux concepts de mesure. En effet, cette forme de présentation du vocabulaire en rend la lecture d'ensemble difficile parce qu'aucun modèle global n'est fourni au lecteur de

l'information contenue dans ce vocabulaire spécialisé : les lecteurs du document doivent essentiellement étudier soigneusement les descriptions textuelles et reconstruire la logique qui fait la liaison entre les termes des différents aspects de mesure. Cela veut dire que chaque lecteur doit reconstruire la topologie des concepts et des sous-concepts reliés aux différents niveaux d'abstraction.

Une question qui se pose est : comment développer une représentation utile qui permette de comprendre à la fois l'ensemble des concepts du VIM, ainsi que chacun individuellement? Nous proposons dans le chapitre 4 de construire des modèles sur la base du texte de VIM pour faciliter la compréhension des concepts inter-reliés de métrologie d'une part, et permettre leur utilisation en tant qu'un outil analytique d'autre part pour le cadre de vérification que nous voulons construire pour les mesures du logiciel.

2.3 Modèle du processus de mesure de Abran et Jacquet

Dans leur travail comme éditeurs de la norme ISO [64] pour le guide de vérification des méthodes de mesure de la taille fonctionnelle (ISO 14143-3)², Abran et Jacquet ont étudié les différentes propositions des méthodes de « validation de mesure » en génie logiciel [6]. Ils ont constaté que plusieurs auteurs ont abordé ce problème de validation en donnant à cette expression des significations différentes, en présentant des méthodes de validation diverses et en utilisant des termes similaires pour exprimer des concepts associés, mais distincts. Pour clarifier la confusion due à des terminologies contradictoires, Abran et Jacquet ont proposé un modèle de processus de mesure [6;76-78]. Ce modèle présenté à la Figure 2 identifie quatre étapes distinctes, à partir de la conception d'une méthode de mesure jusqu'à l'exploitation des résultats de mesure obtenus. Par la suite, Abran et Jacquet ont positionné les approches des différents

² L'objectif initial du document ISO était un guide de validation, mais étant donné le manque de consensus sur la validation des mesures de logiciels, la portée de ce document ISO a été par la suite limitée à un guide de vérification pour certaines propriétés des méthodes de mesure comme décrites dans le VIM.

auteurs ainsi que les divers concepts de validation selon les types de validation en fonction des critères et des techniques de validation décrits.

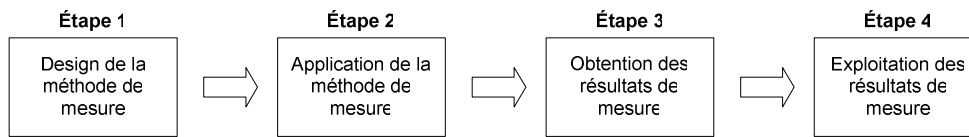


Figure 2 Processus de Mesure – Modèle de Haut Niveau [6;76]

Ce modèle de haut niveau (Figure 2) présente des notions importantes et distinctes de la mesure et met en relief quatre étapes :

1. Design de la méthode de mesure : pour mesurer, il est nécessaire de concevoir une méthode de mesure.
2. Application de la méthode de mesure : consiste à appliquer les règles de la méthode sur le logiciel ou la partie du logiciel à mesurer.
3. Obtention des résultats de mesure : l'application de la méthode de mesure fournit des résultats de mesure qui devraient être vérifiés avant de les exploiter.
4. Exploitation des résultats de mesure : les résultats obtenus par la méthode peuvent être exploités et utilisés dans un modèle qualitatif ou quantitatif.

Ce modèle permet de structurer et de positionner les concepts et les sous-concepts de mesure proposés dans la littérature du génie logiciel.

Le cadre de référence de vérification de mesure, que nous développerons dans cette thèse, adoptera la structure du modèle de processus de mesure. Le chapitre 5 décrira la fusion de ce modèle par rapport à celui du VIM.

2.4 Classification des connaissances de l'ingénierie de Vincenti

Dans son livre *What Engineers Know and How They Know it: Analytical Studies from Aeronautical History* [109], Vincenti a proposé une structure générale des connaissances de l'ingénierie basée sur l'analyse de l'évolution des connaissances en génie

aéronautique. Cette structure identifie six catégories différentes des connaissances de l'ingénierie, présentées dans le Tableau VI, et bien entendu incluant les données quantitatives en tant qu'une catégorie de connaissance du génie.

Tableau VI

Classification de Vincenti des catégories de connaissance de l'ingénierie

#	Catégories de connaissance de Vincenti
1.	Concepts fondamentaux de design
2.	Critères et spécifications
3.	Outils théoriques
4.	Données quantitatives
5.	Considérations pratiques
6.	Instruments de design

Vincenti postule que cette classification de connaissances n'est pas spécifique uniquement au génie aéronautique, mais plutôt générique et applicable au génie dans le sens large. Les données quantitatives en génie indiquent les données descriptives dérivées souvent des expériences bien contrôlées utilisant les concepts reconnus de mesure, des instruments de mesure étalonnés, des protocoles de mesure documentés et des procédures de tests et de réplication pour s'assurer à la fois de la vérification des données en input et de la compréhension du phénomène derrière l'étude pour identifier l'étendue des opérations et ses limites.

Tandis que dans la littérature de gestion de l'ingénierie de logiciels il y a de nombreux articles sur l'estimation, l'ensemble des données disponibles pour la recherche manquent souvent de rigueur dans le sens du génie au niveau des procédures de collecte de données, d'une part, et les résultats d'analyse présentent souvent des limites significatives dans les forces de généralisation, d'autre part. Nous reviendrons sur le problème des modèles d'estimation dans le chapitre 7.

2.5 Corpus de connaissances de Schneidewind

En février 2002, Schneidewind [101] a proposé une version initiale d'un corpus de connaissance (*Body of Knowledge* - BOK) pour la mesure de la qualité d'un produit logiciel, car il est considéré important pour les ingénieurs de produire un produit logiciel de qualité. Le BOK de Schneidewind consiste en un ensemble de neuf thèmes (*issues*) relatifs à la mesure –Tableau VII [101].

Tableau VII

Le BOK en mesure de Schneidewind [101]

Problème	Fonction	Connaissance
1. Objectifs	Analyser les objectifs de qualité et spécifier les besoins de qualité	Ingénierie de la qualité Ingénierie des besoins
2. Coût et risque	Évaluer les objectifs économiques et de risques de la qualité	Analyse économique Analyse du risque
3. Contexte	Analyser l'environnement de l'application	Analyse des systèmes Conception de logiciels
4. Profil opérationnel	Analyser l'environnement de logiciels	Analyse de probabilité et statistique
5. Modèles	Modèle de qualité et validation du modèle	Modèles de probabilité et statistique
6. Exigences des données	Définir le type, la phase, le temps et la fréquence de collection des données	Analyse des données
7. Types de mesures et granularité	Définir les propriétés statistiques des données	Théorie des mesures
8. Évaluation et test du produit et processus	Analyser la relation entre la qualité du produit et la stabilité du processus	Méthodes de test et inspection
9. Prédiction de qualité du produit et processus	Évaluer et prévoir la qualité de logiciel	Outils de mesure

Selon Schneidewind, les mesures obtenues lors des premières phases du cycle de vie du processus de mesure sont généralement moins quantitatives que celles obtenues dans les phases finales. Étant donné l'évolution des processus et des produits durant leur cycle de vie, les objets mesurés durant la phase de test et d'opération ne sont pas les mêmes que

ceux mesurés durant la phase d'analyse des besoins. Schneidewind note également que les approches de conception et les besoins peuvent changer durant le cycle de vie. Le manque de qualité d'un produit logiciel conduit à des effets de coûts considérables tant pour les fournisseurs que pour les clients. Cependant, le corpus proposé ne distingue pas les multiples concepts relatifs à la mesure pour le processus de mesure lui-même. Nous reviendrons sur ce corpus de connaissance dans le chapitre 9.

2.6 Cadre de validation proposé par Kitchenham *et al.*

En génie logiciel, quelques auteurs ont examiné le problème de « validation » de mesure du logiciel, qui se réfère aussi à la « validation de métrique du logiciel » par certains auteurs [41;82;100;104] pour définir le « quoi » et le « comment » valider une mesure en logiciel. Plusieurs auteurs ont proposé plusieurs approches pour répondre à ce problème, mais selon différentes perspectives [41;100;114]. Dans leur article intitulé *Towards a Framework for Software Measurement Validation*, Kitchenham *et al.* mentionnent que « *What has been missing so far is a proper discussion of relationships among the different approaches* » [82]. Ceci montre la nécessité de mettre en place un cadre de « validation » de mesure du logiciel qui clarifie et qui intègre les différentes perspectives de validation proposées par différents auteurs. L'objectif du cadre proposée par Kitchenham *et al.* [82] était de répondre aux questions suivantes :

- une mesure est-elle proprement valide? Permet-elle une caractérisation numérique propre à l'attribut à mesurer?
- une mesure est-elle utilisable? Une mesure parfaite d'un point de vue mathématique n'est intéressante que si elle peut être appliquée.
- une mesure peut-elle prendre une place dans un système de prédiction valide?

La structure de ce cadre proposé par Kitchenham *et al.* [82] présente trois sections :

- structure d'une mesure;
- modèles et mesurage;
- validation des mesures de logiciels.

Plusieurs concepts de mesure ont été traités dans chacune des sections, comme illustré au Tableau VIII.

Tableau VIII

Structure du cadre de validation présenté dans [82]

Sections	Concepts de mesure traités dans chacun des sections
Structure d'une mesure	Entités, attributs et leurs relations
	Unités et types d'échelle et leurs relations
	Valeurs
	Propriétés des valeurs
	Instrument de mesure
	Mesures indirectes
	Unités composées
	Propriétés des mesures indirectes
	Implications pour la validation des mesures
Modèle et mesurage	Modèle de définition de l'unité
	Modèle de l'instrumentation
	Modèle de relation entre attributs
	Problème pratique avec les modèles d'attributs
	Protocole de mesure
	Différences entre les modèles de définition
	Modèle de population de l'entité
Validation des mesures de logiciels	Validation théorique et empirique
	Problèmes de validation théorique de mesure
	Corroboration empirique de mesure
	Validation empirique des relations entre attributs

Ce cadre propose d'aider les chercheurs et les praticiens à valider une mesure du logiciel. Cependant, le comprendre et l'utiliser constitue un défi : en effet, l'expression « validation de mesure » est utilisée de différentes façons avec différentes significations et les auteurs ne mentionnent pas explicitement lequel des sous-concepts de mesure a été utilisé. À cet effet, nous proposons dans le chapitre 8 de réviser ce cadre en utilisant les modèles de métrologie VIM [12] et le processus de mesure [6;76;77] pour mettre en relief certaines clarifications et améliorations. Kitchenham *et al.* [82] proposent particulièrement un modèle structurel de mesure qui décrit les objets impliqués en mesurage et leurs relations au niveau du monde empirique et mathématique. En effet, la

mesure est considérée selon Kitchenham *et al.* comme étant la correspondance d'un attribut empirique à son équivalent dans un monde mathématique (Figure 3).

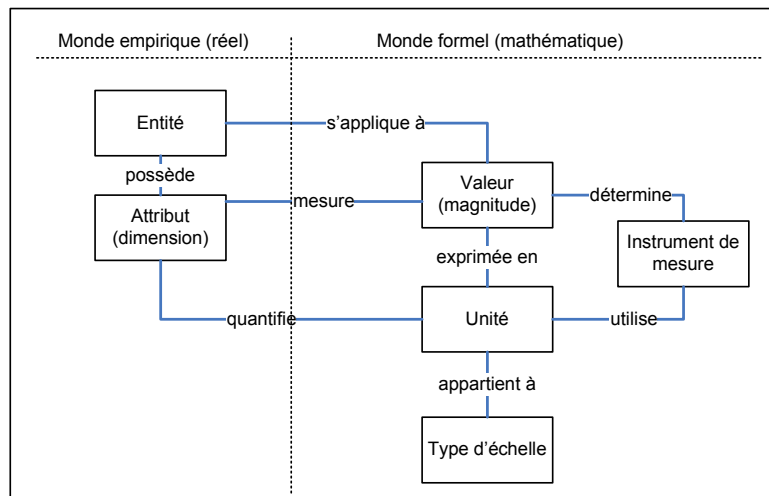


Figure 3 Modèle structurel de mesure (tiré de Kitchenham *et al.* [82])

2.7 Vers une perspective métrologique de la mesure du logiciel

Le processus pour les méthodes de mesure du logiciel était initialement modélisé par Abran et Jacquet dans un objectif d'étudier et de positionner les différentes approches de validation proposées en génie logiciel. Ce processus constitue un cadre fondamental, car il identifie soigneusement les différentes étapes allant de la conception d'une méthode de mesure jusqu'à l'exploitation des résultats obtenus (Figure 2). Cependant, il est à enrichir par les concepts du VIM (Figure 6). Le document du VIM représente les principaux concepts de mesure utilisés pour le développement et l'utilisation des instruments de mesure dans les domaines scientifiques et d'ingénierie. Mais, comme beaucoup d'autres normes sur la définition des mesures et de leurs unités, le VIM ne traite pas l'utilisation des résultats de mesure. Les deux modèles (VIM et processus de mesure) sont complémentaires et forment un cadre de référence précieux pour la mesure.

Deux catégories des concepts de la métrologie VIM (grandeurs et unités, et étalons) sont reliées à l'aspect design d'une méthode de mesure, tandis que les autres catégories (mesurages, instruments de mesure et leurs caractéristiques) portent non pas sur le design en soi, mais sur l'application de la méthode par un instrument de mesure, et sur les résultats obtenus par l'intermédiaire de cet instrument. Comme nous l'avons mentionné, le modèle du processus de mesure de Abran *et al.* inclut d'autres concepts que les grandeurs et les unités et les étalons dans l'étape 1. Le VIM traite en détail les instruments de mesure et leurs caractéristiques dans l'étape 2.

Le processus de développement d'une nouvelle méthode de mesure ou d'amélioration d'une mesure existante se compose ainsi de plusieurs étapes. Dans une même étape, la décomposition en sous-étapes peut s'avérer nécessaire pour comprendre les liens entre les concepts. De ce fait, on aura une vision globale et une détaillée.

La plupart des concepts du VIM sont issus des sciences physiques et de la chimie. Plusieurs de ces concepts ont été appliqués avec succès dans les autres domaines scientifiques et de technologie [28].

Historiquement, plusieurs « grandeurs » telles que (chaud, froid) ont été évaluées par des comparaisons qualitatives. Après l'invention et la convention d'une grandeur formellement définie (i.e. la température), et aussi après le développement des unités, des échelles et des étalons, la mesure de la grandeur température est devenue beaucoup plus précise. Comparativement, le processus d'évolution du génie logiciel est au début par rapport aux autres domaines plus matures des sciences physique et de l'ingénierie.

2.8 Sommaire et observations

La mise en place d'un cadre de « vérification » d'une mesure a fait l'objet de plusieurs recherches dans le domaine du génie logiciel. Parmi les cadres proposés, très peu ont été décrits d'une perspective de la métrologie. La plupart des propositions existantes de

mesure de logiciels utilisent uniquement les concepts de la théorie de mesure pour les représentations empiriques et l'assignation numérique. Or, ces représentations ne sont pas suffisantes pour permettre un traitement convenable de mesure. Dans ce chapitre, nous avons présenté un ensemble d'outils analytiques de vérification d'une mesure provenant du domaine du génie et du génie logiciel. Ainsi, nous avons décrit les avantages et les inconvénients de chaque outil.

La proposition d'un cadre intégré pourra aider à structurer le design des mesures en génie logiciel et à analyser celles proposées par les chercheurs. Une fois que les concepts de la métrologie seront mieux connus par le domaine dans la pratique, par les chercheurs et par le comité ISO du génie logiciel, ces concepts de la métrologie pourront donc être utilisés pour la conception de nouveaux instruments de mesure de logiciels.

CHAPITRE 3

MÉTHODOLOGIE DE RECHERCHE

3.1 Introduction

Dans ce chapitre, nous précisons maintenant la méthodologie que nous avons conçue pour notre projet de recherche. Rappelons que notre objectif principal consiste à établir un cadre intégré de vérification des méthodes de mesure en génie logiciel. Nous présentons d'abord une vue globale sur la démarche effectuée pour atteindre nos objectifs. Puis nous présentons une vue détaillée sur les étapes de développement de ce cadre. Nous présentons particulièrement le développement et l'utilisation du cadre de référence de vérification d'une mesure qui a contribué tant à l'analyse des normes ISO sur la mesure de logiciels qu'à la définition des critères pertinents de vérification et à la proposition du cadre intégré de mesure.

3.2 Méthodologie : Démarche globale

Pour établir un cadre intégré de vérification d'une méthode de mesure, la méthodologie de recherche comprendra des volets théoriques, pratiques et intégrateurs, comme le montre la Figure 4. Voici maintenant une vue globale de la démarche proposée.

3.2.1 Volet théorique

Le volet théorique consiste à :

- identifier les principaux concepts de mesure proposés dans le domaine des sciences, du génie en général (exemples : VIM, classification des connaissances de l'ingénierie de Vincenti) et du génie logiciel en particulier (exemple : modèle du processus de mesure de Abran et Jacquet, le domaine de connaissance en

mesure de Schneidewind, cadre de validation de mesure proposé par Kitchenham *et al.*);

- modéliser les concepts du document ISO sur le vocabulaire de la métrologie (VIM'93), afin de faciliter leur compréhension et leur utilisation;
- développer un cadre de référence de vérification de mesure en se basant sur le modèle du VIM et les éléments du processus de mesure (Abran *et al.*) pour intégrer les concepts du VIM au processus de mesure de logiciels et obtenir une envergure plus large de vérification.

3.2.2 Volet pratique

Le volet pratique consiste à :

- illustrer l'utilisation pratique du modèle du VIM pour analyser tant le design de la méthode COSMIC-FFP (ISO 19761) que le design du prototype COSMIC/RUP;
- illustrer l'utilisation pratique de ce cadre de référence de mesure pour analyser les normes internationales ISO traitant du domaine de connaissance de la mesure en génie logiciel, ainsi que des modèles les plus cités dans la littérature de gestion des projets logiciels, tels que :
 - la description des mesures de la taille fonctionnelle du logiciel dans ISO14143, les mesures de la qualité d'un produit logiciel dans ISO 9126;
 - les éléments du processus de mesure du logiciel dans ISO 15939;
 - le modèle d'estimation des coûts du logiciel COCOMO II;
 - le cadre de vérification de mesure de logiciels proposé par Kitchenham *et al.* afin d'identifier les forces et les faiblesses des différents types de mesure du logiciel.
- proposer des recommandations d'amélioration des mesures proposées et des cadres normatifs;
- illustrer l'utilisation pratique du cadre de référence de mesure avec le corpus de connaissance en mesure de Schneidewind et la classification de Vincenti des

catégories de connaissance de l'ingénierie pour analyser le thème de la mesure identifié dans le guide du corpus de connaissances du génie logiciel (SWEBOK : ISO 19759) pour identifier la couverture des concepts et proposer certaines suggestions d'amélioration.

3.2.3 Volet intégrateur

Le volet intégrateur consiste à produire des documents qui pourraient éventuellement être soumis comme rapports techniques d'amélioration des cadres normatifs dans le domaine de connaissance de la mesure en génie logiciel, lesquels rapports sont présentés dans la liste de publications à l'Annexe A. Les rapports techniques sont normalement établis par le *Joint Technical Committee*, ISO/IEC JTC 1 dans des circonstances exceptionnelles dans le domaine de la technologie de l'information. Les rapports techniques, tels que mentionnés dans le préface de ISO TR 14143-3 [64], sont de trois types :

- type 1, quand le support requis ne peut être obtenu pour la publication de normes internationales, malgré des efforts répétés;
- type 2, quand le sujet est encore sous le développement technique ou pour n'importe quelle autre raison dans le futur mais qu'il n'y a pas de possibilité immédiate d'un consensus sur une norme internationale;
- type 3, quand le comité technique a collecté des données de différentes sortes plutôt que pour la raison pour laquelle est normalement publiée une norme internationale (« état de l'art », par exemple).

Les rapports techniques de type 1 et 2 sont sujets à des révisions dans les trois ans de la publication, pour décider s'ils peuvent être transformés en normes internationales. Les rapports techniques de type 3 ne doivent être nécessairement révisés que lorsque les données qu'ils fournissent sont considérées comme non valides ou inutiles à la longue.

Les propositions de rapports techniques que nous suggérons s'insèrent dans le type 2, et dresse d'une part les critères pertinents de vérification des mesures à rencontrer pour évaluer, de la perspective de la métrologie, la qualité des mesures proposées à l'industrie, et pour améliorer les normes actuelles (et bâtir les futures normes) de mesure du logiciel, et d'autre part la proposition du cadre intégré de vérification de mesure en se basant sur les leçons apprises des analyses et sur les éléments du cadre de référence de mesure, ainsi que sur les éléments du cadre de vérification de mesure de Kitchenham *et al.*[82].

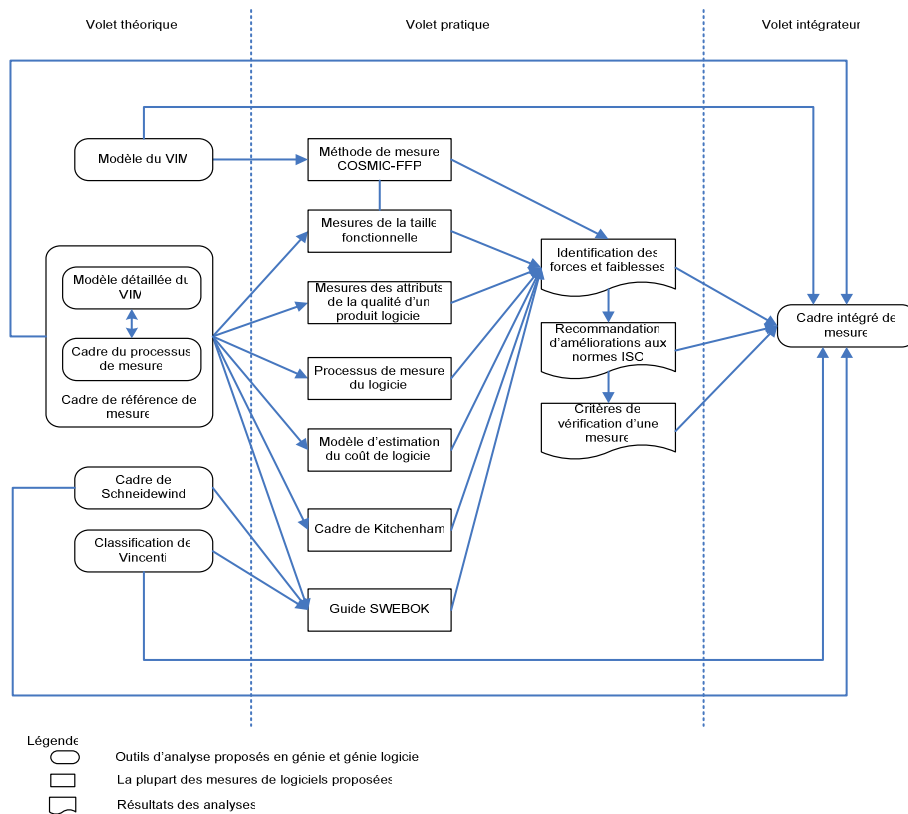


Figure 4 Démarche globale de recherche

3.3 Démarche détaillée

Cette section présente maintenant la méthodologie à un niveau plus détaillé.

3.3.1 Volet théorique

Le volet théorique consiste à introduire dans le domaine de la mesure du logiciel les concepts classiques de métrologie. Ce volet comprend les étapes suivantes :

- identification des principaux concepts de mesure provenant de la littérature des sciences, du génie et du génie logiciel (Chapitre 2);
- sélection du document ISO : VIM, processus de mesure de Abran *et al.*, cadre de vérification de Kitchenham *et al.*, classification de Vincenti, corpus de connaissance de mesure de Schneidewind;
- analyse et modélisation des concepts du VIM (Modèle du VIM) (Chapitre 4) : Nous allons utiliser le document ISO de la métrologie, reconnu par le VIM, comme une base pour identifier l'ensemble du domaine de connaissance de la mesure. Les concepts de mesure dans le document du VIM sont décrits sous un format textuel avec un seul niveau de classification. Pour faciliter le positionnement des concepts interreliés de mesure à d'autres concepts, et pour spécifier les relations et les liens logiques entre les concepts, nous allons élaborer des modèles de concepts du VIM basés sur l'analyse de leurs définitions textuelles et de leurs relations comme définies par leur sémantique. Nous commencerons par établir un modèle de haut niveau présentant les relations entre les principaux concepts du VIM. Ce modèle définira clairement les concepts de métrologie en termes d'intrant, d'extrant et du processus même de mesurage. Nous élaborerons d'autres modèles à un niveau plus détaillé pour examiner les concepts de métrologie séparément et selon différents niveaux d'abstraction. Ces modèles seront conçus pour faciliter d'une part la compréhension des concepts inter-reliés de la métrologie –VIM, et d'autre part, pour permettre leur utilisation en tant qu'outil analytique. Ces modèles sont présentés soit sous forme de processus, soit sous forme de tableau, lorsque les liens entre les concepts sont plutôt énumératifs. Cette première modélisation se réfère à la topologie des concepts dans le VIM. Elle nous permet d'obtenir un modèle de haut niveau et d'autres modèles détaillés du VIM avec les différents niveaux d'abstraction des

concepts et des relations qui existent entre les concepts et les sous-concepts de mesure, et d'illustrer comment les sous-concepts de mesure sont inclus dans le modèle de haut niveau. Cette modélisation a pour objectif de rendre les principaux concepts de mesure plus visibles et plus compréhensibles, et de là d'en faciliter leur application tant dans le design d'une méthode de mesure que dans le design des instruments de mesure reliés à l'application pratique de ces méthodes;

- analyse détaillée et modélisation des concepts du VIM (Modèle détaillé du VIM) (Chapitre 4) : Une autre modélisation plus détaillée est effectuée en utilisant certaines explications additionnelles des termes identifiés dans les notes et dans les exemples des définitions des concepts de mesure du VIM et d'autres références provenant de la littérature sur la mesure en général. Cette modélisation raffinée a pour objectif de nous assurer de la cohérence et de la consistance des représentations des concepts du VIM. L'objectif derrière l'élaboration des modèles du VIM est de faciliter la vérification de la couverture des concepts fondamentaux de la métrologie dans le domaine de la mesure en génie logiciel. Ceci permettra d'apporter de nouvelles perspectives d'analyse et d'interprétation de mesure du logiciel, et de contribuer à la progression de la maturité de ces mesures;
- alignement des concepts du VIM avec ceux du processus de mesure par Abran *et al.* (Cadre de référence de vérification d'une mesure) (Chapitre 5). Dans cette étape, nous utilisons le modèle du VIM et le modèle du processus de mesure de Abran *et al.* pour établir le pont entre les concepts de mesure provenant du domaine de connaissance de la métrologie et ceux qui proviennent d'une analyse structurée de la littérature spécialisée en génie logiciel. L'alignement de ces concepts de mesure produira un cadre initial de référence de vérification d'une mesure identifiant les différentes étapes de la mise au point d'une méthode de mesure jusqu'à l'exploitation des résultats qu'elle peut fournir. Ce cadre permettra de positionner les différentes mesures proposées dans la littérature du

génie logiciel. De plus, il permettra de vérifier si l'ensemble des parties du processus de mesure a été complètement traité.

3.3.2 Volet pratique

Le volet pratique consiste à améliorer le processus de vérification des méthodes de mesure en génie logiciel et d'analyser les normes de mesure. Ce volet comprend les étapes suivantes :

- après avoir identifié le cadre de référence de mesure, nous allons vérifier comment les mesures de logiciel rencontrent ce cadre et nous allons décrire les limites et les problèmes qui doivent être résolus;
- analyse structurée des normes ISO traitant la mesure en génie logiciel (identification des forces et des faiblesses des normes ISO 9126, ISO 14143 et ISO 19761 : COSMIC-FFP) (Chapitre 6) : Pour étudier les mesures proposées par ISO, nous analyserons dans un premier temps le contenu des séries ISO 9126-1 à 4. Une analyse de haut niveau sera effectuée pour en identifier les catégories de mesure présentées dans les documents ISO 9126-2 à 4, en particulier pour identifier quelle est la méthode ISO de mesure de la qualité d'un produit logiciel. Quelles sont les mesures reconnues par la norme ISO? Et comment les appliquer? Par la suite une analyse détaillée sera conduite pour vérifier si les mesures dans les séries 9126-2 à 4 rencontrent l'étape du design d'une méthode de mesure du cadre de référence de mesure;
- analyse de la norme ISO 14143 en utilisant le cadre de référence de mesure pour vérifier si la définition de mesure de la taille fonctionnelle du logiciel rencontre les étapes du cadre initial de mesure;
- analyse de la norme ISO 19761, qui est reconnue par la méthode de mesure de la taille fonctionnelle du logiciel COSMIC-FFP, en utilisant principalement le modèle du VIM pour vérifier non seulement la couverture du design de cette méthode par rapport aux catégories grandeurs et unités et étalon du VIM, mais

aussi son application dans un environnement choisi RUP/ Rational Rose selon les catégories mesurage, résultat de mesure, instruments de mesure et leurs caractéristiques. Cette analyse permettra ainsi d'explorer si une telle méthode de mesure couvre en elle-même la majorité des concepts du domaine du VIM;

- analyse structurée de la norme ISO 15939 et le modèle d'estimation COCOMO II (identification des forces et des faiblesses des normes ISO 15939 et COCOMO) (Chapitre 7) : Analyse de la norme ISO 15939 et du modèle d'estimation des coûts de logiciel COCOMO en utilisant le cadre de référence de mesure pour ressortir leurs forces et leurs faiblesses. Cette analyse a pour objectif d'identifier les insuffisances des normes utilisées à date pour le design et pour l'application des mesures de logiciels.

3.3.3 Volet intégrateur

Le volet intégrateur consiste à développer une proposition d'un cadre intégré avec des critères recommandés de vérification. Ce volet comprend les étapes suivantes :

- analyse structurée du cadre de vérification proposée par Kitchenham *et al.*[82] (identification des forces et des faiblesses du cadre de Kitchenham *et al.*) (Chapitre 8) : Dans cette étape, nous utiliserons le cadre de référence de vérification de mesure pour analyser le cadre de vérification de mesure proposé par Kitchenham *et al.* Cette analyse a pour objectif d'identifier si les concepts du cadre proposé par Kitchenham *et al.* rencontrent les concepts du cadre de référence, c'est-à-dire de vérifier s'il couvre complètement l'ensemble détaillé des concepts de métrologie et du processus de mesure. Ceci permettra d'identifier la liste des concepts de mesure discutés dans Kitchenham *et al.* d'une part, et d'apporter certaines clarifications et améliorations d'autre part;
- analyse structurée de ISO TR 19759 : SWEBOK (identification des forces et des faiblesses de SWEBOK) (Chapitre 9) : Dans cette étape, nous analyserons le thème de la mesure dans le guide du corpus des connaissances en génie logiciel (SWEBOK : ISO 19759). Ainsi, nous commencerons par comparer la taxonomie

initiale pour le domaine de connaissance (KA) de la mesure de logiciels proposé par Buglione *et al.* [27] par rapport au BOK de la mesure de Schneidewind [101] pour vérifier leurs structures. Par la suite, nous procéderons à la classification des concepts du BOK de la mesure en utilisant la classification de Vincenti des connaissances de l'ingénierie [109] pour identifier les forces et les faiblesses de la mesure de logiciels en tant que concept et qu'outil fondamental de l'ingénierie, et pour identifier les améliorations requises d'une perspective de l'ingénierie. Finalement, nous proposerons la mise en correspondance des mesures dans les dix chapitres de SWEBOK par rapport au cadre de référence pour vérifier les fondements de mesure d'une perspective de métrologie et pour apporter des suggestions d'amélioration. Cette analyse permettra également d'identifier à quel niveau les recherches subséquentes de mesure sont requises;

- mise en correspondance entre les cadres proposés de mesure (cadre intégré de vérification d'une mesure) (Chapitre 10) : la combinaison des différents cadres de vérification (VIM, Abran *et al.*, Kitchenham *et al.*) permettra de présenter de manière synthétique le cadre intégré de vérification d'une mesure;
- identification des critères de vérification d'une mesure (recommandation d'utilisation du cadre intégré) (Chapitre 10) : finalement, les résultats des analyses détaillées du VIM, du cadre de vérification de mesure de Kitchenham *et al.*, l'utilisation du cadre de référence de mesure et à partir des leçons apprises des analyses des documents ISO, nous identifierons de nouveaux critères de vérification que devraient éventuellement respecter toute nouvelle proposition de mesure de logiciels; un cadre intégré et unifié de vérification d'une mesure du logiciel.

Nous proposons également des rapports techniques présentés à l'annexe A pour l'amélioration des cadres normatifs dans le domaine de connaissance de la mesure en génie logiciel. La Figure 5 illustre les étapes de notre méthodologie détaillée de recherche.

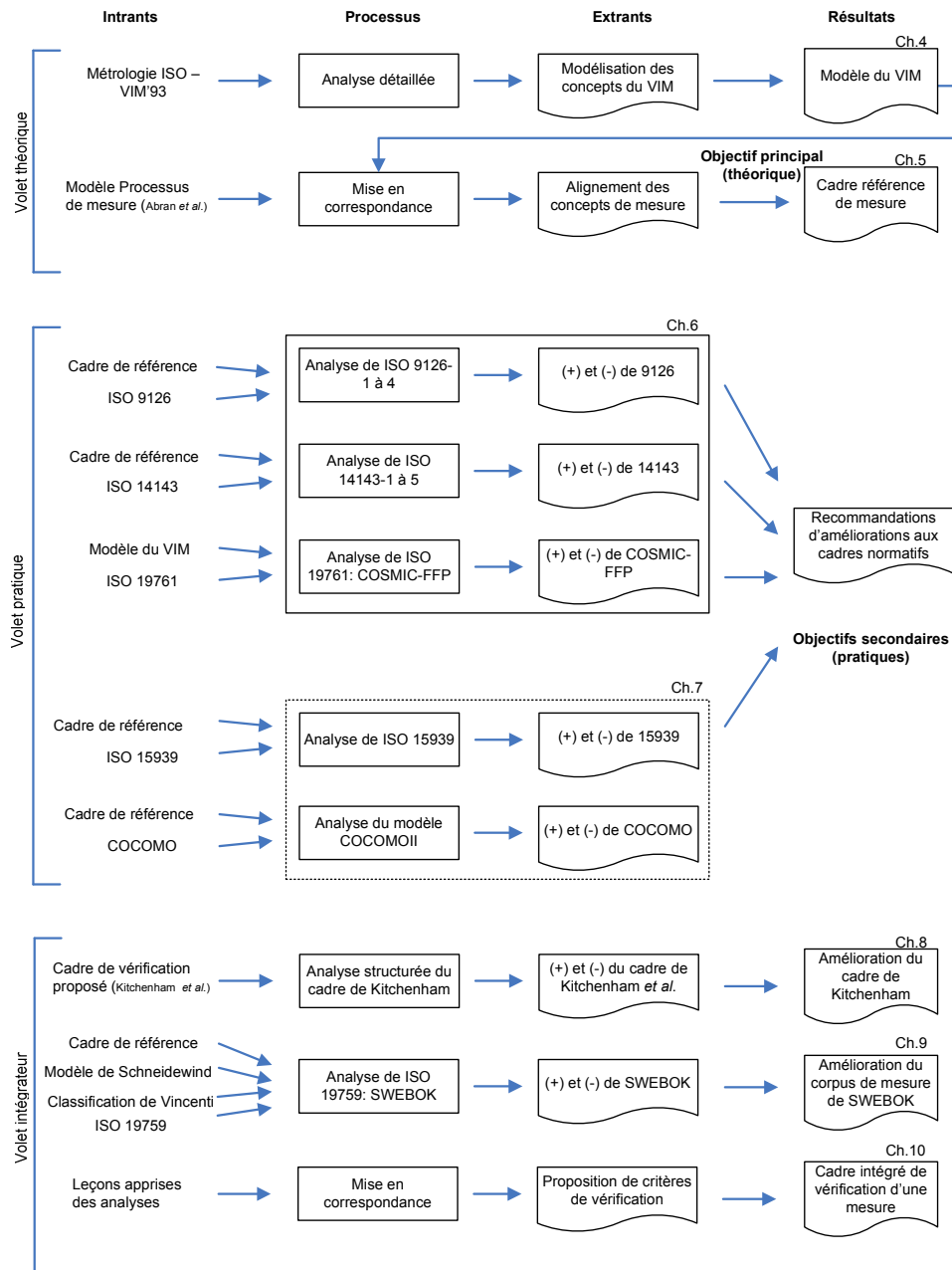


Figure 5 Méthodologie détaillée de recherche

Cette méthodologie nous permettra donc d'élaborer un cadre intégré de vérification des méthodes de mesure en génie logiciel. À plus long terme, ce cadre pourrait influencer le développement d'un modèle d'une méta-norme couvrant tous les critères que devrait éventuellement rencontrer toute nouvelle proposition de mesure du logiciel. Ce cadre présentera les critères de vérification requis pour les méthodes de mesure de logiciels. Quelques résultats plus spécifiques incluront un ensemble de rapports (Annexe A) proposant des améliorations aux normes internationales de mesure du logiciel.

3.4 Vérification

Étant donné que nous proposons un cadre intégré de vérification des méthodes de mesure, il sera nécessaire d'entreprendre des étapes de vérification telles que proposées au Tableau IX. La première vérification est d'ordre logique. Il faut s'assurer qu'il n'y a pas de contradictions logiques. La deuxième vérification consiste à se demander si le cadre proposé peut être intégré au sein des connaissances déjà admises, et s'il y a une compatibilité. La troisième vérification est expérimentale et consiste à contrôler des faits constatés au cours des observations selon des instruments appropriés. Le fait de trouver des vérifications expérimentales décisives demande parfois beaucoup de temps.

Tableau IX

Étape à suivre (inspiré de [44])

Étapes à suivre		Objectif
Vérification logique	Les critères du cadre proposé devraient être exemptés de toute contradiction logique	Veiller à la cohérence du cadre
Vérification de compatibilité	Pas de contradiction entre ces critères et l'un ou l'autre élément des connaissances déjà admises	Veiller à la cohésion avec l'ensemble des connaissances déjà admises

Tableau IX (Suite)

Étapes à suivre		Objectif
Vérification expérimentale	<p>Identification des faits, les plus nombreux et variés, qui sont logiquement déductibles du cadre, des connaissances déjà admises et des conditions initiales d'expérimentation</p> <p>Si on trouve de tels faits, alors le cadre et les connaissances déjà admises sont globalement confirmés. Sinon, au moins un parmi le cadre et les connaissances déjà admises doit être abandonné ou modifié ou bien il faut revoir le protocole expérimental</p>	Vérifier la conformité du cadre avec les faits connus

Ainsi, notre hypothèse est que le cadre proposé contient l'ensemble des éléments qui jouent un rôle dans la vérification d'une mesure. On s'intéresse à des connaissances déjà établies, à divers outils d'analyse proposés et à des analyses pour confirmer ou rejeter l'hypothèse. Autrement dit, on propose au début un cadre de référence de vérification. On suppose que ce cadre identifie tous les éléments potentiels de vérification, puis on l'utilise ensuite pour faire des analyses de normes. Selon les résultats des analyses, il y a deux cas : ou bien il faut revoir le cadre, ou bien on confirme le cadre. Une fois confirmé, et à partir des leçons apprises des analyses, on peut l'enrichir.

CHAPITRE 4

MODÉLISATION DES CONCEPTS DE MESURE DANS VIM'93

4.1 Introduction

Dans les domaines scientifiques, incluant l'ingénierie, ainsi que dans les autres domaines tels que l'administration des affaires et les sciences sociales, la mesure représente l'un des outils analytiques de ces domaines de connaissance. Pour la mesure, ces domaines se réfèrent en général au sous-domaine de connaissance relié à la métrologie.

Le document ISO qui constitue un consensus officiel national et international sur la terminologie commune de la métrologie est connu sous l'abréviation VIM (Vocabulaire international des termes fondamentaux et généraux de métrologie) [57].

Le VIM obéit à quelques concepts de présentation traditionnelle du vocabulaire, avec des termes définis individuellement sous forme de texte utilisant un seul niveau de classification. Ce mode de représentation constitue un défi en termes d'assemblage de l'ensemble des termes inter-reliés de mesure ainsi que de comprendre la liste de ces termes ayant leurs propres significations distinctes et précises. Pour améliorer la présentation et faciliter la compréhension de cet ensemble complexe des concepts inter-reliés de mesure qui se réfère à la métrologie, nous utilisons les techniques d'ingénierie de modélisation de processus de production de logiciels.

Nous proposons dans cette thèse un modèle initial de l'ensemble des 120 définitions documentées dans le VIM ainsi que les concepts additionnels de mesure se trouvant dans ces définitions mais qui ne sont pas explicitement définies. En particulier, cette modélisation illustre les différents niveaux d'abstraction des concepts ainsi que les relations qui existent entre les concepts et les sous-concepts. Cette représentation se réfère à la topologie des concepts dans le VIM.

Comme décrit dans la section 3.2, le document ISO de la métrologie VIM'93 ne met à la disposition des lecteurs aucun modèle pour faciliter le positionnement des concepts inter-reliés à d'autres concepts en mettant en relief pour les lecteurs les relations entre les termes et les liens logiques entre ceux-ci. Notre méthodologie nous a permis de construire des modèles des termes de VIM basés sur l'analyse de leurs définitions textuelles et de leurs relations définies par leur sémantique. Le modèle de haut niveau proposé pour l'ensemble des six catégories des termes du VIM est présenté dans la Figure 6.

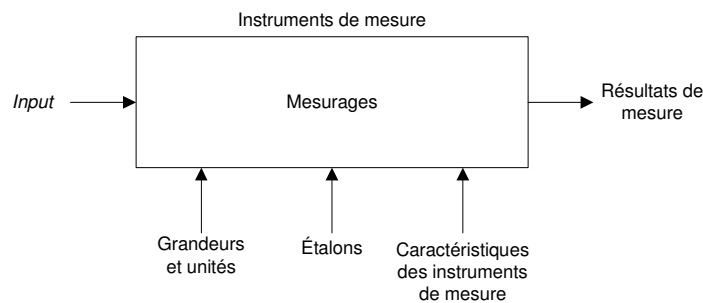


Figure 6 Modèle des processus des catégories des termes VIM'93

Pour représenter les relations entre les éléments des catégories de la Figure 6, la représentation usuelle d'un processus de production a été sélectionnée : exemple, input, output et les variables de contrôle ainsi que le processus lui-même. Dans la Figure 6, l'output est représenté par les « résultats de mesure » et le processus par le « mesurage » dans le sens des opérations de mesure, tandis que les variables de contrôle sont les « étalons » et les « grandeurs et unités ». De plus, les opérations de mesure elles-mêmes peuvent être influencées par les « caractéristiques des instruments de mesure ».

L'objectif principal de toute mesure est d'établir une description quantitative des choses (phénomène, corps, substance,...) pour obtenir une valeur. Ceci est basé sur les valeurs ou les tendances fournies par les dispositifs de mesure qui doivent avoir certaines caractéristiques. Pour interpréter l'information obtenue, des étalons de mesure sont

nécessaires. Toutes ces considérations sont importantes et seront traitées en détail dans les sections suivantes.

Il est à noter que dans toutes les figures et les tableaux de cette modélisation, un terme pris directement du VIM apparaîtra en lettres à caractère régulier alors que les concepts qui ne sont pas listés spécifiquement apparaîtront en *italique*. Par exemple : dans la Figure 6, nous avons ajouté le terme *input* qui n'est pas inclus dans aucune des six catégories du VIM. Par ailleurs, dans le VIM, un terme qui a un qualificatif additionnel apparaît entre parenthèses pour faciliter la compréhension et prévoir les illusions des interprétations spécifiques données à ce terme du document ISO; par exemple grandeur (mesurable) du Tableau X. Finalement, le VIM propose quelquefois deux expressions pour le même ensemble de concepts. Ces expressions sont incluses dans le VIM et sont rattachées par un tiret oblique (/), ce qui veut dire que les deux termes peuvent être utilisés alternativement, exemple : grandeur de dimension un/grandeur sans dimension du Tableau X. Quand deux termes qui figurent dans une figure ne sont pas liés par un tiret oblique, ils représentent donc deux termes distincts du VIM, par exemple : les termes « mesurande » et « signal de mesure » de la Figure 9 représentent deux termes distincts.

Ce modèle, et l'ensemble des sous-modèles présentés ci-après, correspondent à notre compréhension courante de la topologie intégrée dans le vocabulaire de ce domaine spécialisé du corpus de connaissance relatif à la métrologie.

4.2 PARTIE I : Modèle initial du VIM'93

Tout d'abord, un modèle de haut niveau de l'ensemble des concepts du VIM est présenté dans la partie I, suivi par d'autres modèles de niveaux plus détaillés de chacune des six catégories des termes du VIM. Par la suite, dans la partie II, d'autres détails de chacun des termes de chaque catégorie du VIM sont présentés en tenant en considération non seulement les définitions des 120 termes du VIM explicitement définis mais aussi

d'autres concepts de mesure incorporés dans les notes des définitions et d'autres références utilisées soit pour clarifier ces concepts soit pour fournir des exemples additionnels. Finalement, nous présentons certaines observations et conclusions.

4.2.1 Grandeurs et unités

La Figure 7 présente la modélisation de la première catégorie des concepts de mesure du VIM qui est « grandeurs et unités ». Cette catégorie a été modélisée en identifiant quatre concepts fondamentaux parmi l'ensemble des 22 termes : système de grandeurs, dimension d'une grandeur, unité de mesure et valeur d'une grandeur. La topologie détaillée de ce modèle est présentée au Tableau X qui positionne l'ensemble des termes définis dans cette catégorie. Dans un système de grandeurs, une grandeur est généralement exprimée par une dimension, une valeur et une unité.

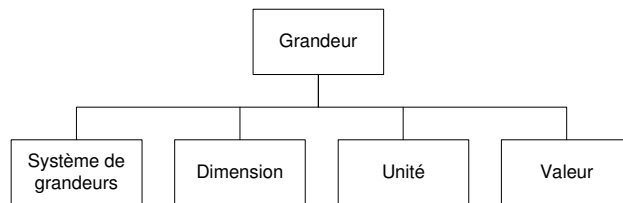


Figure 7 Topologie de haut niveau de la catégorie « grandeurs et unités »

Tableau X

Topologie détaillée de l'ensemble des concepts « grandeurs et unités »

Grandeur (mesurable)			
Système de grandeur	Dimension d'une grandeur	Unité de mesure	Valeur (d'une grandeur)
Grandeur de base	Grandeur de dimension un/	Symbole d'une unité	Valeur vraie
Grandeur dérivée	Grandeur sans dimension	Système d'unités	Valeur conventionnellement vraie
		Unité dérivée cohérente	Valeur numérique
		Système cohérent d'unités	Échelle de repérage
		Système international d'unités/ SI	
		Unité de base	
		Unité dérivée	
		Unité hors système	
		Multiple d'une unité	
		Sous-multiple d'une unité	

4.2.2 Mesurages

Le second ensemble des concepts est dans la catégorie « mesurage ». Cette catégorie regroupe neuf termes de mesure et est considérée comme étant le processus permettant d'obtenir expérimentalement un résultat de mesurage. Tout d'abord, la Figure 8 présente la topologie de haut niveau de cette catégorie. La hiérarchie des niveaux est définie du plus général vers le plus spécifique. En effet, le terme « métrologie » inclut tous les aspects de mesurage (théorique et pratique) qui se réfèrent collectivement à la science de la mesure dans la littérature de la métrologie. Le principe de mesure représente la base scientifique d'un mesurage. À partir de ce principe de mesure, une « méthode de mesure » décrite dans le sens général comme étant une succession logique des opérations mises en œuvre lors de l'exécution de mesurage est ainsi « instanciée » d'une manière spécifique par le « mode opératoire » comme étant un ensemble détaillé d'opérations. Le mode opératoire représente ainsi la description détaillée d'un mesurage en se basant sur un ou sur plusieurs principes de mesure et sur une méthode de mesure

donnée. À la Figure 8, nous nous référons à cette topologie, à cette hiérarchie des concepts de mesure comme étant les fondements de la mesure.

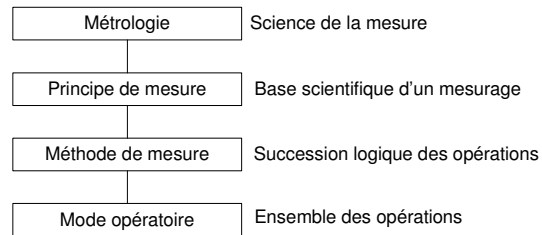


Figure 8 Fondements de la mesure

Par ailleurs, la technique de modélisation pour les processus est utilisée pour présenter d'autres concepts dans cette catégorie. Le mode opératoire est représenté par le processus de mesure dans la Figure 9. Son input est le mesurande et son output est le résultat de mesurage. Les deux variables de contrôle sont explicitement énoncées dans cette catégorie, c'est-à-dire la méthode de mesure en général et la grandeur d'influence qui a un effet sur le résultat de mesure durant le processus de mesurage : par exemple, la température d'un micromètre lors de la mesure d'une longueur d'un objet particulier peut causer une déviation de mesure. De la même manière, un opérateur de mesure est une variable de contrôle dans le processus de mesure; le terme « opérateur » n'est pas inclus explicitement dans l'ensemble des définitions du VIM mais paraît dans la note du concept « mode opératoire » de mesure.

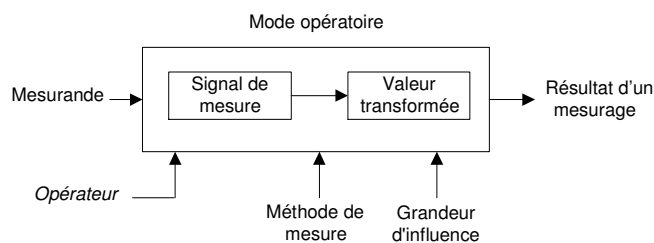


Figure 9 Processus de mesure – Topologie détaillée des sous-concepts

Ainsi, pour effectuer un mesurage, un opérateur doit concevoir et suivre un « mode opératoire » qui consiste en un ensemble d'opérations, décrites d'une manière spécifique, et mises en œuvre lors de l'exécution de mesurages particuliers selon une méthode donnée. Le VIM spécifie également qu'une instanciation d'un mode opératoire traite un « signal de mesure » et produit une valeur transformée, valeur de l'input qui représente un mesurande donné.

4.2.3 Résultats de mesure

La troisième catégorie des concepts du VIM est les « résultats de mesure ». Dans cette catégorie, les 16 termes ont été regroupés en trois sous-ensembles selon les types de résultats de mesure, les modes de vérification des résultats de mesure et l'information de l'incertitude associée à ces résultats, comme présentés dans le Tableau XI.

Tableau XI

Topologie détaillée de la catégorie « résultat d'un mesurage »

Résultat d'un mesurage		
<i>Types des résultats de mesure</i>	<i>Modes de vérification des résultats de mesure</i>	Incertitude de mesure
Indication (d'un instrument de mesure) Résultat brut Résultat corrigé	Exactitude de mesure Répétabilité (des résultats de mesurage) Reproductibilité (des résultats de mesurage)	Écart-type expérimental Erreur (de mesure) Écart Erreur relative Erreur aléatoire Erreur systématique Correction Facteur de correction

4.2.4 Instruments de mesure

Le quatrième ensemble de concepts de mesure du VIM est la catégorie « instruments de mesure » regroupant 31 termes. La suite d'éléments d'un système de mesure à partir de l'input jusqu'à l'output constitue la « chaîne de mesure » : d'ailleurs, le concept « chaîne de mesure » est représenté comme un modèle d'un processus dans la Figure 10 avec

l'input (le stimulus) comme un signal d'entrée et l'output (la réponse) comme le signal de sortie, et le système de mesure représentant lui-même le processus, c'est-à-dire l'ensemble complet des instruments de mesure avec d'autres équipements assemblés pour réaliser des mesurages. La topologie de haut niveau d'une « chaîne de mesure » (Figure 10) est ensuite détaillée dans la Figure 11 et la Figure 12.

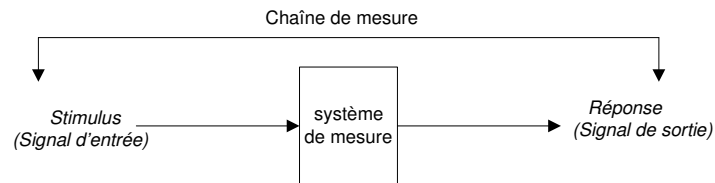


Figure 10 Chaîne de mesure

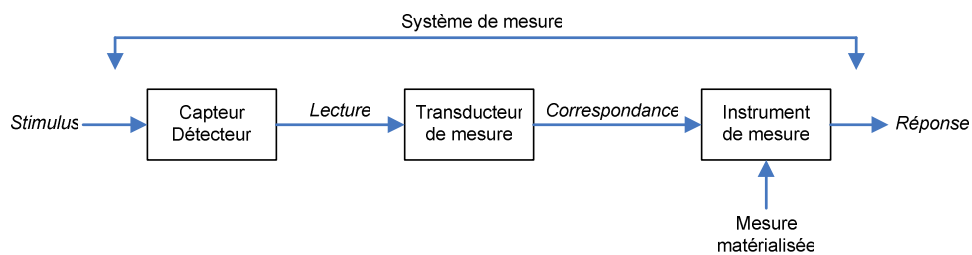


Figure 11 Système de mesure

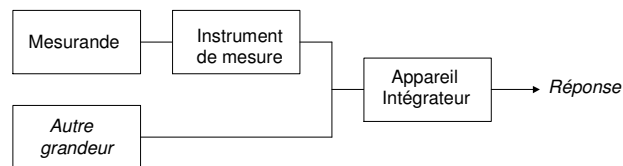


Figure 12 Appareil intégrateur

La Figure 11 présente une vue plus détaillée d'un système de mesure : suite à un stimulus ou à un signal d'entrée, le capteur (ou le détecteur) va détecter la présence d'un signal. Suite à une lecture, le transducteur de mesure fournit une grandeur ayant une

relation bien déterminée avec la grandeur du signal d'entrée. La grandeur obtenue par le transducteur correspond ainsi à la grandeur destinée à être mesurée par le moyen de l'instrument de mesure, qui va indiquer à son tour une valeur correspondante à la réponse. Certains instruments de mesure ne peuvent être utilisés qu'avec l'association d'un ou plusieurs dispositifs annexes tels que par exemple les mesures matérialisées (exemple : pour mesurer le poids d'un objet, il faut associer des masses marquées avec l'instrument balance). S'il y a une autre grandeur (Figure 12), un appareil intégrateur va déterminer la valeur du mesurande en intégrant une grandeur avec d'autres destinées à être mesurées. L'output va être indiqué par un dispositif d'affichage et peut être aussi enregistré dans certains dispositifs enregistreurs (Figure 13). Dans le cas où il y a plus qu'un système de mesure, un appareil de mesure totalisateur détermine la valeur du mesurande par l'addition des valeurs partielles du mesurande obtenues simultanément ou consécutivement à partir d'une ou plusieurs sources (Figure 14).

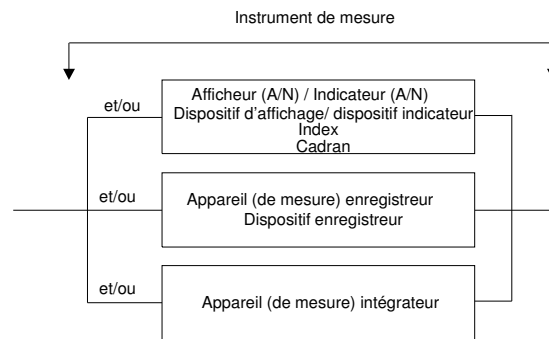


Figure 13 Détail d'un instrument de mesure

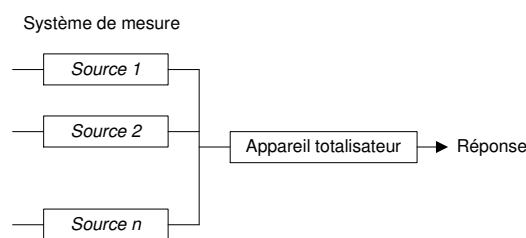


Figure 14 Modèle d'un appareil de mesure totalisateur

Dans cette catégorie de termes sur les instruments de mesure, les dix termes suivants ont été regroupés dans un tableau structuré de la topologie des échelles (Tableau XII). Ce tableau énumère la collection de quatre types d'échelle, c'est-à-dire linéaire, non-linéaire, à zéro décalé et dilatée, ainsi que les propriétés d'échelle incluant la longueur, l'étendue des indications, la division, la longueur d'une division, l'échelon ou encore la valeur d'une division et la « chiffraison » d'une échelle. Les termes que nous avons ajoutés (en italique) sont les « types d'échelle » et les « propriétés d'une échelle ».

Tableau XII

Topologie détaillée des échelles d'un appareil de mesure

Échelle (d'un appareil de mesure)	
<i>Type d'échelle</i>	<i>Propriétés d'une échelle</i>
Échelle linéaire	Longueur d'échelle
Échelle non-linéaire	Étendue des indications
Échelle à zéro décalé	Division
Échelle dilatée	Longueur d'une division
	Échelon/ valeur d'une division
	« Chiffraison » d'une échelle

Finalement, la catégorie des termes des « instruments de mesure » contient également l'ensemble des concepts sur les types d'opérations requis pour l'utilisation d'un instrument particulier mis à la disposition de l'utilisateur : calibrage, ajustage et réglage (Tableau XIII).

Tableau XIII

Opérations sur les instruments de mesure

Opérations sur un instrument de mesure
calibrage
ajustage
réglage

4.2.5 Caractéristiques des instruments de mesure

La cinquième catégorie des concepts du VIM est les « caractéristiques des instruments de mesure ». Dans cette catégorie, les 28 termes ont été regroupés en deux sous-ensembles selon les caractéristiques d'utilisation et de contrôle, comme présentés dans le Tableau XIV. Ces termes ont été ajoutés (en italique). Les caractéristiques d'utilisation regroupent les caractéristiques d'opération, d'affichage et dynamiques. Les caractéristiques de contrôle regroupent les caractéristiques qualitatives et quantitatives.

Tableau XIV

Topologie détaillée des « caractéristiques des instruments de mesure »

Caractéristiques des instruments de mesure	
<i>Caractéristiques d'utilisation</i>	
<i>Caractéristiques d'opération (quantitatives)</i>	Conditions assignées de fonctionnement (valeur assignée) Conditions limites Conditions de référence (valeur de référence/ étendue de référence)
<i>Caractéristiques d'affichage (reliées à l'indication)</i>	Calibre Intervalle de mesure (étendue) Étendue de mesure Valeur nominale Constante (d'un instrument) Résolution (d'un dispositif afficheur)
<i>Caractéristiques dynamiques (reliées au temps)</i>	Caractéristique de transfert Constance Discrétion Dérive Temps de réponse Zone morte

Tableau XIV (Suite)

Caractéristiques des instruments de mesure	
<i>Caractéristiques de contrôle</i>	
<i>Caractéristiques qualitatives (statiques)</i>	Sensibilité (seuil de) mobilité Exactitude d'un instrument de mesure Justesse (d'un instrument de mesure) Fidélité (d'un instrument de mesure)
<i>Caractéristiques quantitatives de contrôle et d'estimation (reliées à l'erreur)</i>	Classe d'exactitude Erreur (d'indication) d'un instrument de mesure Erreurs maximales tolérées/ limites d'erreur tolérées Erreur au point de contrôle Erreur à zéro Erreur intrinsèque Erreur de justesse Erreur réduite conventionnelle

4.2.6 Étalons

Finalement, le sixième ensemble des 14 termes du VIM est la catégorie « étalon ». Les étalons sont requis pour établir une unité d'une ou de plusieurs valeurs d'une grandeur pour être utilisés comme référence. Les 14 termes relatifs aux étalons sont présentés dans le Tableau XV et sont regroupés en deux sous-catégories : le type d'étalons et les concepts relatifs à la conservation d'un étalon. Le terme que nous avons ajouté (en italique) est le type d'étalon.

Tableau XV

Topologie détaillée d'un étalon

Étalon	
<i>Type d'étalon</i>	Conservation d'un étalon
Étalon international	Traçabilité
Étalon national	Étalonnage
Étalon primaire	Matériau de référence (MR)
Étalon secondaire	Matériau de référence certifié (MRC)
Étalon de référence	
Étalon de travail	
Étalon de transfert	
Étalon voyageur	

Généralement, on retrouve l'étalon de référence, dont dérivent les mesurages et, qui sert comme statut légal (national ou international) ou largement reconnu (primaire ou secondaire). L'étalon de transfert sert comme intermédiaire pour comparer des étalons entre divers utilisateurs. Pour la conservation d'un étalon, les matériaux de référence et les matériaux de référence certifiés sont requis avec la traçabilité et l'étalonnage à chaque fois que c'est nécessaire.

Finalement, quinze termes ont été ajoutés, soit pour illustrer la classification des concepts, soit pour tenir compte des explications additionnelles dans les notes et les exemples de concepts de mesure ainsi que pour clarifier la présentation des modèles.

4.3 PARTIE II : Modèle détaillé du VIM'93

Dans cette deuxième partie, l'objectif est de préciser le modèle initial des concepts de mesure dans le VIM et de fournir un modèle détaillé représentant un outil de vérification de la qualité des mesures en termes de concepts de métrologie. Nous croyons que cet outil est un guide fondamental qui saura mieux répondre aux besoins des industriels. Il s'agit d'un document de synthèse qui peut être adapté particulièrement à la mesure du logiciel.

Dans la partie I, les 120 termes relatifs aux concepts de mesure des six catégories définies dans le VIM ont été présentés, soit en utilisant des modèles de processus quand c'est applicable soit en les groupant en sous-catégories des termes dans différents tableaux quand c'est approprié. Dans la partie I, seulement les 120 termes explicitement définis ont été modélisés et classifiés. Cependant, le VIM contient également d'autres concepts de mesure au niveau des définitions ainsi que d'autres références utilisées pour clarifier ces concepts ou pour fournir des exemples additionnels. Quand présentés, ces termes additionnels permettent de comprendre la terminologie spécialisée du VIM incluant les relations entre plusieurs termes.

Ces termes seront présentés dans des sections séparées pour chacune des six catégories de termes. Selon l'utilité, des exemples et des clarifications de la littérature de la métrologie peuvent être inclus.

4.3.1 Grandeurs et unités

4.3.1.1 Concepts clés

Selon le VIM, le terme grandeur (mesurable) est défini comme étant un « attribut d'un phénomène, d'un corps ou d'une substance susceptible d'être distingué qualitativement et déterminé quantitativement ». Il découle de cette définition qu'une grandeur (dans le sens général) s'applique généralement à tout ce que l'on peut mesurer ou quantifier, telles que la température, le temps, la longueur, la masse, etc. Quand on délimite le mesurande (le mesuré) en portions distinctes et égales, on parle d'une grandeur particulière (exemple : longueur d'une tige donnée). Pour ce faire, le mesureur utilise une unité de mesure (grandeur particulière). Cette unité est normalement choisie pour des raisons de commodité eut égard aux circonstances dans lesquelles la mesure est faite. Il en est de même pour l'utilisation de l'étalon de mesure. Par exemple, un étalon de mètre est certainement commode pour mesurer la longueur d'une tige donnée ou encore les dimensions d'une maison donnée, mais il serait inopérant pour évaluer les dimensions d'une particule atomique. D'où la nécessité de choisir une unité, une grandeur particulière à laquelle on compare les autres grandeurs de même nature pour les exprimer quantitativement par rapport à cette grandeur. Les grandeurs de même nature sont des grandeurs qui peuvent être classées les unes par rapport aux autres en ordre croissant ou décroissant, et qui peuvent être groupées en catégories de grandeurs³

Figure 15.

³ Ces éléments ne sont pas définis dans le VIM, mais illustrés par des exemples

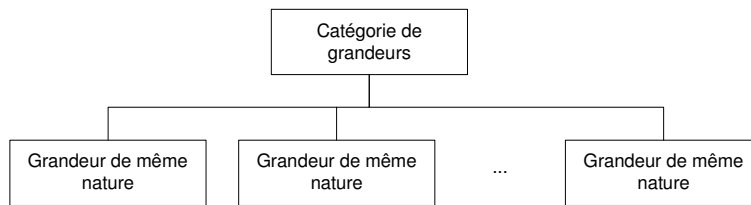


Figure 15 Catégories de grandeurs

Une grandeur possède ainsi les caractéristiques suivantes, comme listées dans le Tableau XVI : par sa nature, par sa grandeur mesurable ou repérable, par son niveau qui peut être général ou particulier (unité), et par son nom et son symbole selon les normes ISO 31 [56] ou le système international d'unité (SI).

Tableau XVI

Caractéristiques d'une grandeur

Niveau	Nature	Nom	Symbole
Général	Mesurable	Nom de la grandeur	Basé sur l'ISO31
Particulier	Repérable		Basé sur le SI, etc.

Une mesure a pour but d'attribuer une valeur à une grandeur. L'opération de mesure (ou encore le mesurage) diffère selon la nature de grandeur utilisée. À cet effet, il est utile de distinguer les grandeurs mesurables des grandeurs dites repérables. Les grandeurs pour lesquelles on peut définir l'identité et la relation d'ordre en se fixant une échelle de repérage sont dites grandeurs repérables [30]. Ces grandeurs jouent un rôle important dans les domaines scientifiques et techniques là où on a besoin de repérer des objets pour être mesurables tels que les domaines de la biologie, la pharmacologie ou la résistance des matériaux. Les grandeurs mesurables sont des grandeurs pour lesquelles on peut définir non seulement l'identité, exemple $Q_1 = Q_2$ et la relation d'ordre entre les grandeurs $Q_1 \langle Q_2$ ou $Q_1 \rangle Q_2$ mais aussi $\frac{Q_1}{Q_2} = n$, où Q_1, Q_2 sont des grandeurs et n est un nombre réel. Les grandeurs mesurables se prêtent à une description simple entre les grandeurs et les phénomènes physiques en utilisant des modèles algébriques. La

première utilisation de l'algèbre consiste à exprimer une valeur d'une grandeur mesurable comme une unité de mesure multipliée par un nombre, n exemple : ($Q_1 = n.Q_2$). La seconde utilisation de l'algèbre consiste à exprimer les relations entre différentes grandeurs sous forme algébrique (exemple : une aire s'exprime par le produit de deux longueurs, etc.).

Nous reviendrons sur ces concepts plus loin, notamment à propos de l'unité et de l'étalon de mesure. Nous notons que l'étalon de mesure peut être un objet ou un processus naturel destiné à conserver ou à reproduire une unité de mesure. Processus naturel, dans le sens d'une unité hors système (exemple : un jour est le temps écoulé d'une aube à une autre, est un étalon naturel de mesure du temps). Par contre, le mètre, le kilogramme, la minute, de même que leurs multiples (multiple d'une unité) et leurs subdivisions (sous-multiple d'une unité) sont des étalons de mesure « artificiels » qui ont une dimension standard garantie par des mécanismes appropriés, et ils sont normalement plus stables que les étalons naturels.

4.3.1.2 Mesure d'une grandeur

Dans ce contexte, la mesure d'une grandeur consiste à « établir le rapport d'une grandeur avec [sic] une autre de même espèce choisie comme unité » [91] ou encore « de lui attribuer un nombre qui fixe, de façon exacte et indiscutable, son état et qui fournisse ainsi, par simple rapprochement de nombres, le résultat de sa comparaison avec une grandeur quelconque de la même espèce, mesurée de même, en tout autre temps comme en tous autres lieux et circonstances » [91]. En d'autres termes, l'unité de mesure est une grandeur bien définie prise comme terme de comparaison avec des grandeurs de même nature. Cela veut dire que les nombres qui résultent de ces comparaisons donnent les mesures de ces grandeurs.

4.3.1.3 Grandeur de base

Selon le VIM, une grandeur de base est définie comme « l'une des grandeurs qui, dans un système de grandeurs, sont admises par convention comme étant fonctionnellement indépendantes les unes des autres ». Les caractéristiques d'une grandeur de base sont présentées sous forme d'un tableau incluant le symbole, la dimension, l'unité de base et la valeur numérique (Tableau XVII). Il est à noter que la dimension d'une grandeur est une « expression représentant une grandeur d'un système de grandeurs comme le produit de puissances de facteurs qui représentent les grandeurs de base de ce système ». Par exemple : supposons qu'on a un système de grandeurs comprenant la longueur (L), la masse (M) et le temps (T) comme étant des grandeurs de base. L, M et T sont dits facteurs qui représentent des grandeurs de base et sont appelés « dimension ». Pour une grandeur dérivée telle que la force, sa dimension est le produit de puissance de facteurs, qui est égale à LMT^{-2} .

Tableau XVII

Grandeur de base

Concepts liés à une grandeur de base			
Symbole	Dimension	Unité de base	Valeur numérique

4.3.1.4 Grandeur dérivée

Une grandeur dérivée est une « grandeur définie, dans un système de grandeurs, comme fonction des grandeurs de base de ce système ». Cela veut dire qu'elles se définissent à partir des grandeurs de base (exemple, la vitesse est une grandeur dérivée définie comme le quotient des grandeurs de base : longueur par temps). Les caractéristiques d'une grandeur dérivée sont la dimension de la grandeur, l'unité dérivée d'une grandeur dérivée dans un système donné, et la valeur numérique (Tableau XVIII).

Tableau XVIII

Grandeur dérivée

Concepts liés à une grandeur dérivée			
Symbole	Dimension	Unité dérivée	Valeur numérique

Certaines grandeurs dérivées sont définies par le rapport de deux grandeurs de même nature (exemple : indice de réfraction en physique qui est le rapport de la vitesse de la lumière dans un milieu à la vitesse de la lumière dans un autre milieu); la dimension de ces grandeurs peut être exprimée par le nombre un et leur unité est nécessairement une unité dérivée cohérente avec les autres unités d'un système d'unités (exemple le système international d'unités SI). Leur unité est exprimée par le nombre un, puisqu'elle résulte du rapport des deux unités SI identiques. Cela veut dire que l'unité SI de toutes les grandeurs dont la dimension est un produit de dimension égale à un est le nombre un. D'autre part, pour les grandeurs ayant pour unité le nombre un, on retrouve des nombres qui servent à indiquer un comptage (exemple le nombre de molécules). Généralement, l'unité un n'est pas mentionnée explicitement. Les grandeurs sans dimension sont un nombre sans unité mais qui sont significatifs. Cependant, dans certains cas on attribue à l'unité un nom spécial pour éviter la confusion avec certaines unités dérivées composées (exemple : angle plan est exprimé en radian, qui est $m.m^{-1} = 1$). On en déduit qu'à partir des grandeurs de base, on peut avoir une grandeur de dimension un ou sans dimension et une valeur sous forme d'un nombre, comme le montre la figure suivante (Figure 16).

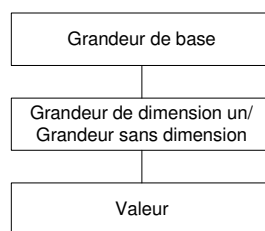


Figure 16 Grandeur de dimension un/ Grandeur sans dimension

Selon le VIM, les unités de grandeurs qui ont la même dimension peuvent avoir le même nom et le même symbole, même si ces grandeurs ne sont pas de même nature. Nous supposons l'exemple suivant en nous basant sur les noms des unités du SI : l'unité Ampère (A) peut correspondre aussi bien à la grandeur de base courant électrique qu'à la grandeur dérivée force magnétique. Ainsi, une même unité peut correspondre à plusieurs grandeurs différentes. À cet effet, le nom de l'unité à lui seul ne fait pas connaître la grandeur mesurée. Pour cela, un instrument de mesure doit donner des indications tant pour l'unité que pour la grandeur mesurée. D'autre part, en pratique, pour des raisons de commodité et pour éviter toute confusion entre les grandeurs ayant la même dimension, certaines unités sont exprimées par des noms spéciaux ou par des combinaisons d'unités telle que l'unité SI de fréquence qui est appelée Hz plutôt que s^{-1} . D'autre part, Figliola mentionne dans son livre [43] qu'une dimension définit une variable physique pour décrire certains aspects d'un système physique et que la valeur de base associée avec toute dimension est donnée par l'unité. Ainsi, une unité définit une mesure d'une dimension. Par exemple, des termes tels que la masse, la longueur et le temps décrivent des dimensions de base pour lesquelles on associe les unités kilogramme, mètre et seconde.

4.3.1.5 Système de grandeurs

Par définition, un système de grandeurs est un « ensemble de grandeurs, dans le sens général, dans lesquelles il existe des relations définies » [57]. Dans un système de grandeur, on retrouve des grandeurs admises par convention qui peuvent être fonctionnellement indépendantes telles que les grandeurs de base ou dépendantes des grandeurs de base et qui constitue une fonction mathématique telles que les grandeurs dérivées. Ainsi, il existe différentes catégories de grandeurs qui peuvent inclure un groupement des grandeurs de même nature (Figure 17).

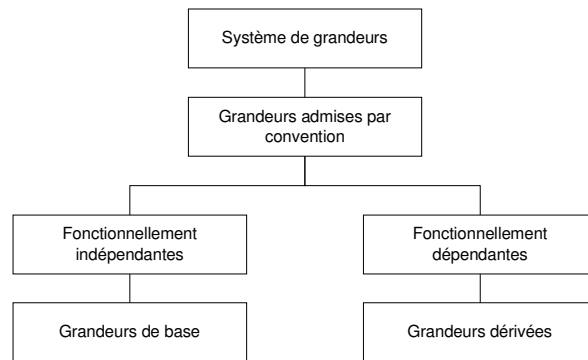


Figure 17 Système de grandeurs

4.3.1.6 Système d'unités de mesure

Un système d'unités est « l'ensemble des unités de base et des unités dérivées, définies suivant les règles données, pour un système de grandeurs donné » [57]. Un système cohérent d'unités inclut toutes les unités cohérentes de mesure pour un système donné. Par exemple : le système international d'unités (SI) est un système cohérent d'unité adopté par la CGPM (Conférence générale des poids et des mesures). Pour un système cohérent d'unités (tel que le SI) et pour chaque grandeur de base correspond une seule unité de base (Figure 18). Selon le VIM, une unité (de mesure) (dérivée) cohérente est une unité dérivée qui s'exprime par le produit de puissance des unités de base avec un facteur de proportionnalité égal à un. Une unité peut être cohérente dans un système et non dans un autre.

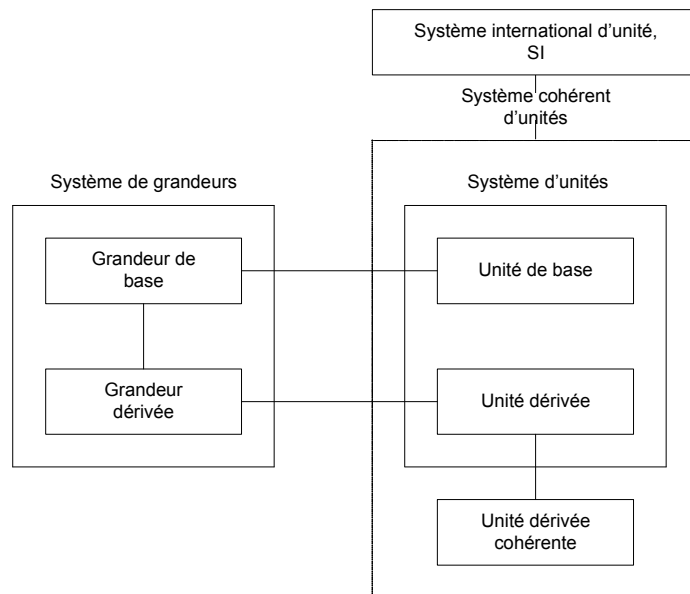


Figure 18 Système de grandeurs, système d'unités et système cohérent d'unités

Dans leur livre *Mesures physiques*, Pérard *et al.* [91] mentionnent que l'unité d'une grandeur peut être une grandeur quelconque de la même nature. Cependant, comme il existe entre les différentes grandeurs de même nature des rapports de proportionnalité, il y a souvent des simplifications de certains de ces rapports pour les ramener à l'unité. Exemple : les rapports des surfaces S et S' de deux rectangles sont égaux aux rapports des produits a et b , a' et b' des dimensions linéaires de leurs côtés.

$$\frac{S}{a.b} = \frac{S'}{a'.b'} = h; \text{ ou encore } S = h.a.b; \quad S' = h.a'.b'. \text{ Mais pour des raisons de}$$

simplification, on prend $h=1$. Il en résulte pour un carré de côté a , $\frac{S}{a^2} = 1$; si $a = 1$, alors

$S = 1$; cela revient à prendre comme unité de surface le carré ayant pour côté l'unité de longueur. C'est ainsi que la surface d'un rectangle est égale au produit de ses côtés. Selon Pérard [91], un système d'unité est dit absolu ou cohérent, si tous les facteurs, tels que h , sont égaux à l'unité. Les lois physiques, qui s'expriment par des formules telles

que $S = h.a.b$ lient entre elles la presque totalité des diverses grandeurs, et par la suite leurs unités. Certaines unités sont prises comme de base, les autres comme dérivées. Le choix des unités de base et même de leur nombre est principalement arbitraire. Mais, la précision d'une unité dérivée est, au plus, égale à la plus petite des précisions qu'autorisent séparément celles des unités de base d'où elle procède. Ainsi, une unité de base doit être susceptible de fournir un étalon de la plus haute précision.

4.3.1.7 Unités de mesure

Comme nous l'avons dit plus haut, l'unité de mesure hors système représente un étalon de mesure qui peut être naturel (exemple : les unités de temps jour, heure sont des unités hors système pour le SI) ou artificiel (exemple : l'unité d'énergie électronvolt est une unité hors système pour le SI). Les multiples d'une unité, leurs subdivisions et les sous-multiples d'une unité sont des étalons de mesure qui ont une dimension standard garantie par des mécanismes appropriés (Figure 19). Généralement, une unité de mesure représente une grandeur particulière ayant un nom et un symbole de l'unité (Figure 20).

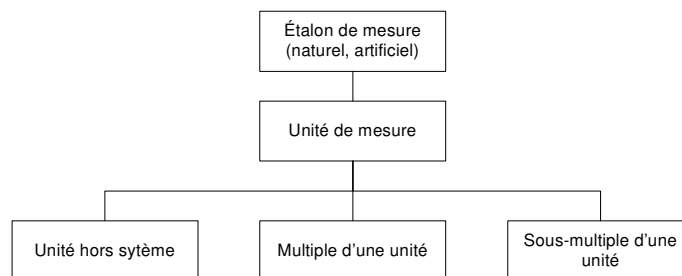


Figure 19 Unité de mesure hors système, multiple d'une unité et sous-multiple d'une unité

Grandeur particulière	Unité de mesure	
	Nom	Symbole d'une unité

Figure 20 Unité de mesure

L'étalon de mesure permet de définir une unité de mesure. Et, donc une grandeur particulière (exemple : le mètre) pour servir de référence (Figure 21). Cela signifie que, par convention et pour des raisons de l'opération de mesure, l'étalon est maintenu indivisé, bien qu'il soit en principe divisible. Il doit être stable, de telle sorte que les modifications du lieu n'entraînent pas des variations quantitatives à l'étalon en question. Il doit être aussi homogène par rapport au mesurande pour avoir une plus grande précision des résultats de mesurage; cela veut dire qu'il vaut mieux mesurer une longueur par une unité de longueur, un poids par une unité de poids, etc. Cependant, cela n'est pas toujours possible pour les mesures indirectes ou les mesures de ce qui est qualitatif et qu'il faut quantifier.

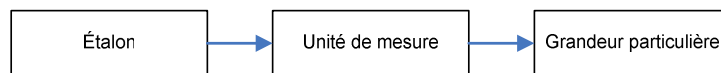


Figure 21 Lien entre grandeur, unité et étalon

4.3.1.8 Valeur d'une grandeur

Une valeur est une « expression quantitative d'une grandeur particulière, généralement sous la forme d'une unité de mesure multipliée par un nombre » [57]. Nous supposons qu'une mesure ait une valeur méthodologique, dans le sens qu'elle constitue une méthode (succession logique d'opérations) nous permettant d'obtenir une valeur d'une grandeur donnée, représentée par le résultat du mesurage et associée en principe à l'incertitude de mesure. Cependant, dans la description des notes (1.19, page 16) du VIM'93 [57], il est pris pour acquis qu'un mesurande peut être décrit par une seule valeur vraie, obtenue par un mesurage parfait. Mais dans la réalité, les instruments et les mesurages ne fournissent pas cette valeur à cause des erreurs systématiques et de l'incertitude de mesure. Par contre, le mesurage dérivé par l'utilisation d'un étalon de référence dans un lieu donné ou une organisation donnée permet d'obtenir une valeur conventionnellement vraie, représentant une valeur avec une incertitude appropriée pour un usage donné. Cette valeur est quelquefois appelée valeur assignée, meilleure

estimation de la valeur, valeur convenue ou valeur de référence selon les notes (1.20, page 17) de [57]. De plus, l'incertitude est déterminée par la dispersion des valeurs obtenues par un grand nombre de résultats de mesure (Tableau XIX).

Une valeur d'une grandeur peut être aussi obtenue suite à un comptage (exemple valeur numérique) en se basant par exemple sur une échelle de repérage (Figure 22). D'autre part, certaines grandeurs, pour lesquelles on ne sait pas définir leur rapport à une unité, peuvent être exprimées par référence à une échelle de repérage et/ou à un procédé de mesure spécifié (Figure 22 et Tableau XX).

Tableau XIX

Valeur d'une grandeur

Grandeur particulière	Mesurage	Résultat de mesure	Incertitude de mesure
Unité de mesure	Mesurage parfait	Valeur vraie	Indéterminée
Étalon de référence	Mesurage dérivé par l'étalon de référence en un lieu donné	Valeur conventionnellement vraie : valeur assignée meilleure estimation de la valeur valeur convenue valeur de référence	Dispersion des valeurs obtenues par un grand nombre de résultats de mesure

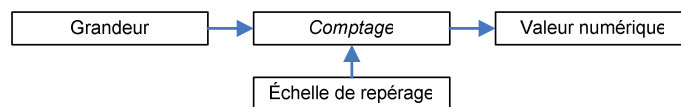


Figure 22 Valeur numérique obtenue par comptage

Tableau XX

Échelle de repérage

Grandeur	Référence	Catégories de grandeurs
Grandeurs particulières	Échelle de repérage	Grandeurs de même nature

4.3.1.9 Observations : comptage et mesurage

Il est important de comprendre les différences entre le comptage et le mesurage. Le comptage vise à répondre à une question qui commence par le mot « combien », donc à faire l'évaluation quantitative d'une grandeur. En d'autres termes, c'est établir une correspondance biunivoque (correspondance un à un) entre le mesurande et une valeur numérique (ensemble des nombres entiers, rationnels, irrationnels, réels ou complexes). Cela veut dire : exprimer le mesurande à l'aide d'un élément différent de l'ensemble des nombres qui est utilisé, en respectant la relation d'ordre qu'on y trouve (en utilisant une échelle de type ordinal). Exemple, compter le nombre d'étudiants dans une classe, c'est mettre celui-ci en correspondance biunivoque avec la série des nombres entiers positifs, le premier élément de cette série étant le nombre un. Il en est de même quand on mesure une longueur, une durée, un volume, etc. Sauf dans ces cas, on utilise une unité de mesure qui est choisie pour rendre facile le mesurage (l'opération de mesure).

Mesurer, c'est établir une valeur d'une grandeur donnée. Elle consiste à comparer une grandeur physique inconnue avec une grandeur de même nature prise comme référence, à l'aide d'un instrument de mesure. Le résultat de cette comparaison est exprimé en une valeur (qui peut être proposée par un système de mesure) associée à une unité qui rappelle la nature de référence et à l'incertitude qui dépend de la qualité de l'expérience, de la connaissance de la référence et des conditions d'utilisation. Selon Himbert, mesurer consiste à comparer une grandeur physique inconnue avec une référence dont la traçabilité est établie dans le cadre du SI, et d'évaluer l'incertitude associée, par une approche statistique de la mesure permettant d'apprécier la justesse du résultat, la fidélité des instruments : c'est la métrologie statistique [48]. La question qui se pose est comment établir la relation entre la valeur réelle d'une grandeur physique et celle qui est réellement mesurée?

4.3.1.10 Sommaire

Pour clarifier l'ensemble des termes associés avec les grandeurs et unités comme définies explicitement dans le VIM, des termes additionnels ont été introduits de même que leurs relations modélisées dans différents tableaux et figures. Ces termes additionnels ont été incorporés soit à partir des définitions du VIM soit à partir de la littérature de métrologie. Pour un objectif de traçabilité, ces termes sont énumérés dans le Tableau XXI associés avec les références consultées.

Tableau XXI

Termes additionnels dans la catégorie « grandeurs et unités »

Termes additionnels	Référence du VIM des termes additionnels	Autres références
Grandeur dans un sens général Grandeur particulière Grandeurs de même nature Catégories de grandeurs Symboles de grandeurs	Dans la définition « grandeur (mesurable) »	
Grandeur repérable		Cloarec [31] (AFNOR)
Étalon de mesure naturel	Dans la définition « unité de mesure hors système »	Gagnon <i>et al.</i> [44]
Étalon de mesure artificiel		
Mesure d'une grandeur		Pérard <i>et al.</i> [91]
Valeur assignée Meilleure estimation Valeur convenue Valeur de référence	Dans la définition « valeur conventionnelle vraie d'une grandeur »	
Métrologie statistique		Himbert [48]

4.3.2 Mesurages

Dans le VIM, le concept « mesurage » représente un ensemble d'opérations ayant pour but de déterminer une valeur d'une grandeur. Il est à noter que certains éléments ne figurent pas directement dans le VIM mais ces éléments nous semblent nécessaires à la cohérence et à la précision de certains concepts et servent de liens entre d'autres concepts tels que le mode de vérification des résultats de mesure.

Pour effectuer un mesurage, un opérateur doit concevoir et suivre un mode opératoire, qui consiste en un ensemble d'opérations, décrites pour la mise en œuvre de mesurages particuliers selon une méthode donnée. Le document qui donne les informations nécessaires à la réalisation de mesurage est parfois appelé lui-même le mode opératoire, permettant à l'opérateur de réaliser un mesurage sans avoir recours à des informations additionnelles. Pour accomplir un mesurage, une méthode de mesure est requise et doit être « instanciée » dans un contexte particulier. Dans le VIM, une telle méthode de mesure peut être obtenue selon la méthode de substitution, la méthode différentielle ou la méthode de zéro. Le résultat obtenu peut être influencé par une grandeur d'influence, comme décrite dans la Figure 9 (grandeur qui n'est pas le mesurande mais qui a un effet sur le résultat de mesure, telle que la température d'un micromètre lors de la mesure d'une grandeur) (Figure 23).

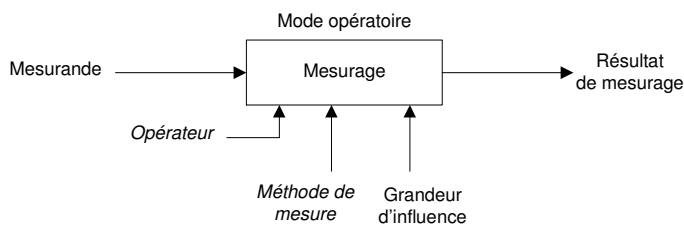


Figure 23 Topologie détaillée de « mesurage »

Dans un mesurage, l'instanciation d'une mesure utilise un signal de mesure et produit une valeur transformée du mesurande, représentant un mesurande donné en utilisant un dispositif de mesure, exemple : signal électrique de la sortie d'un transducteur de pression (Figure 24).

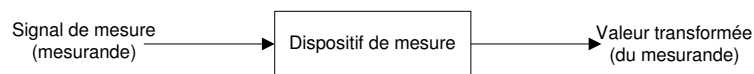


Figure 24 Phénomène de transformation d'un signal de mesure

Il est à noter que le dispositif de mesure de la Figure 24 a été décomposé en deux sous-processus dans le VIM, c'est-à-dire à travers les opérations du capteur/détecteur et le

transducteur (Figure 11). Par ailleurs, dans son livre *Theory and Design for Mechanical Measurements*, Figliola [43] considère le mesurande qui représente le signal d'input à un système de mesure comme étant une fonction $F(t)$ et le signal d'output comme une fonction $Y(t)$. Il mentionne que pour un signal d'input, $F(t)$, le système exécute une opération donnée qui fournit le signal d'output, $Y(t)$. Cependant, dans la pratique, $Y(t)$ doit être utilisé pour déduire $F(t)$, et ce à travers un modèle mathématique général du système. Ainsi, il est nécessaire de comprendre la qualité de l'opération de mesure exécutée par le système pour interpréter correctement le signal de l'input. Par conséquent, un modèle général mathématique devrait être proposé pour un système de mesure. En représentant le signal d'entrée comme une fonction de l'input au modèle, ça permet d'étudier simplement comment le système de mesure se comporte en résolvant l'équation du modèle, qui permet d'accomplir essentiellement l'équivalence analytique d'étalonnage d'un système. Ceci fournit l'information requise pour déterminer le signal de l'input auquel un système de mesure est plus adapté.

4.3.3 Résultats de mesure

La catégorie des « résultats de mesure » a été présentée dans la section 4.2.3 dans le Tableau XI. Un résultat de mesure doit être associé à l'évaluation de son incertitude. Selon Placko [93], il faut s'assurer d'une part du raccordement de la mesure à des étalons de référence et d'autre part du calcul de l'incertitude. Un résultat de mesure peut être modifié sous l'effet des 5M (moyen, milieu, méthode, matière et main-d'œuvre) : par l'instrument de mesure (moyen), l'environnement dans lequel la mesure est faite (milieu), la méthode utilisée, les propriétés mécaniques (matière) et le mesureur (main-d'œuvre). Par ailleurs, pour le mode quantitatif de vérification des résultats de mesure, le terme utilisé en métrologie est celui d'exactitude plutôt que de précision. L'exactitude de mesure est définie dans le VIM par « l'étroitesse de l'accord entre le résultat d'un mesurage et la valeur vraie du mesurande (valeur conventionnellement vraie dans la pratique) ». Les mesures très exactes sont également précises, mais les mesures très

précises ne sont pas nécessairement exactes. Par exemple, un mètre mal calibré ou une balance défectueuse peuvent donner des valeurs précises mais erronées (inexactes). Par contre, la répétabilité est définie par « l'étroitesse de l'accord entre les résultats des mesurages successifs du même mesurande, mesurages effectués dans la totalité des mêmes conditions de mesure ». Elle nous permet de déterminer l'erreur de l'input (mesurande) en fonction de l'output (résultat de mesure) dans les mêmes conditions de mesure. En d'autres termes, la répétabilité est basée sur l'écart-type expérimental (mesure de la variation de l'output pour un input donné). Par ailleurs, la reproductibilité est définie par « l'étroitesse de l'accord entre les résultats des mesurages du même mesurande, mesurages effectués en faisant varier les conditions de mesure ». Le terme reproductibilité se réfère aux résultats obtenus par des tests de réplication selon des conditions séparées (Tableau XXII).

Tableau XXII

Modes de vérification des résultats de mesure et leurs conditions

<i>Mode de vérification des résultats de mesure</i>	<i>Conditions</i>
Exactitude	Accord entre une valeur du mesurande et une valeur (conventionnellement) vraie du mesurande
Répétabilité	Même conditions de répétabilité : même mode opératoire même observateur même instrument de mesure utilisé dans les mêmes conditions même lieu Répétition durant une courte période de temps
Reproductibilité	Conditions à faire varier : principe de mesure méthode de mesure observateur instrument de mesure étalon de référence lieu conditions d'utilisation Temps

En conclusion, le mesurage qui constitue un ensemble d'opérations nous permet d'obtenir des résultats de mesure, qui doivent être associés à l'incertitude de mesure. En effet, en absence d'incertitude, il n'est plus pertinent de comparer entre eux les résultats trouvés ou de comparer un résultat à des valeurs de référence spécifiées par exemple dans une norme ou dans un texte réglementaire.

4.3.4 Instruments de mesure

4.3.4.1 Types d'instruments de mesure

À partir d'un ou de plusieurs dispositifs de mesure, nous formons un instrument de mesure/appareil de mesure. Il y a différents types d'appareils de mesure : appareil afficheur/appareil indicateur, enregistreur, totalisateur, intégrateur, à affichage analogique, à affichage numérique. Chacun a ses propres fonctionnalités (Tableau XXIII), par exemple, un appareil de mesure, une mesure matérialisée ou un transducteur de mesure, représentent chacun un dispositif (Figure 25). Cependant, un dispositif de mesure peut représenter une partie d'un appareil de mesure⁴, par exemple un dispositif d'affichage/dispositif indicateur, le dispositif enregistreur et le capteur/détecteur.

Tableau XXIII

Types d'appareils de mesure

Types d'appareils de mesure
Appareil afficheur/ appareil indicateur
Appareil enregistreur
Appareil totalisateur
Appareil intégrateur
Appareil de mesure analogique
Appareil de mesure numérique

⁴ Un dispositif de mesure peut ne pas fournir une valeur de la grandeur associée, par exemple le détecteur qui indique la présence d'un phénomène

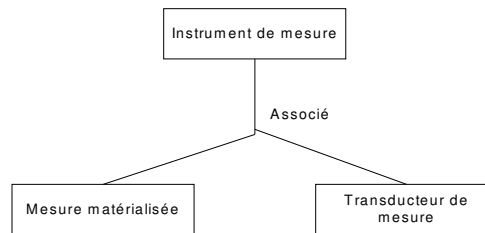


Figure 25 Instrument de mesure et dispositifs associés

4.3.4.2 Échelle d'un instrument de mesure

Un dispositif indicateur/afficheur d'un appareil de mesure est composé de l'index, de l'échelle et du cadran (Tableau XXIV). Une échelle doit appartenir à un type d'échelle avec certaines propriétés (caractéristiques). Parmi les types d'échelle, il y a l'échelle linéaire, non-linéaire, à zéro décalé et dilatée. Les propriétés d'une échelle d'un appareil de mesure sont : longueur d'échelle, étendue des indications, division, longueur d'une division, échelon (valeur d'une division) et « chiffraison » d'échelle (Tableau XII).

Tableau XXIV

Détails d'un appareil de mesure afficheur/ indicateur

<i>Dispositif de mesure : Partie d'un appareil de mesure afficheur/ indicateur</i>	Dispositif d'affichage/ dispositif indicateur Dispositif enregistreur Capteur/ Détecteur
<i>Partie d'un dispositif indicateur</i>	Cadran Échelle Index

4.3.4.3 Système de mesure

Comme mentionné dans la section 4.3.4.1, à partir d'éléments, nous formons un dispositif de mesure. Ce dernier peut représenter un appareil de mesure ou un instrument de mesure. Pour être utilisé, un instrument de mesure peut être associé à plusieurs dispositifs annexes. Il est destiné à faire des mesurages et donc à produire un certain résultat. La suite d'éléments d'un appareil de mesure ou d'un système de mesure forme

une chaîne de mesure. Un système de mesure constitue l'ensemble complet d'instruments de mesure et d'autres équipements assemblés pour exécuter des mesurages spécifiés. Une fois qu'il est installé, le système est appelé installation de mesure (Figure 26).

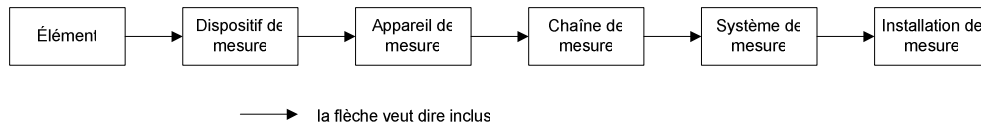


Figure 26 Modèle général - Succession des concepts de la catégorie « instrument de mesure » selon un ordre approximatif de complexité

Par convention et pour les besoins de l'opération de mesure, un instrument de mesure doit répondre aux conditions de calibrage, d'ajustage et de réglage (Tableau XIII). Selon Figliola *et al.*, une mesure consiste principalement à attribuer une valeur spécifique à une variable physique [43]. Cette variable physique devient la variable mesurée. Un système de mesure représente l'outil utilisé pour la quantification de la variable physique. Toujours selon Figliola *et al.*, ce système de mesure consiste en quatre étapes principales : (1) étape de capteur-transducteur, (2) étape de conditionnement du signal, (3) étape de l'output, et (4) étape de feed-back –contrôle (Figure 27).

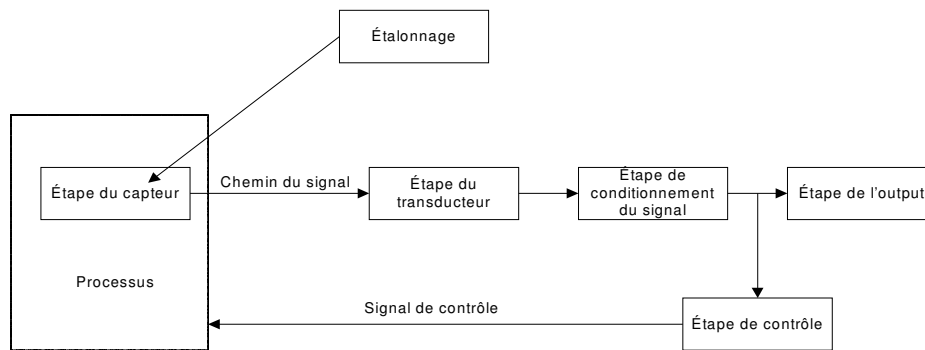


Figure 27 Composants d'un système de mesure général [43]

(1) Étape de capteur-transducteur : le capteur utilise un phénomène donné pour détecter la variable à mesurer, représentant l'input du système de mesure. Le transducteur convertit à son tour cette information détectée dans une forme d'un signal détectable qui peut être électrique, mécanique, optique ou autre, c'est-à-dire, dans une forme qui peut être facilement quantifiée. L'exemple donné dans [43] est celui d'un thermomètre. Le mercure dans l'ampoule du thermomètre agit comme un capteur. En forçant l'expansion du mercure, ce système de mesure transforme l'information thermique en déplacement mécanique. Ainsi le design interne de l'ampoule agit comme un transducteur. Il est à noter que le terme « transducteur » peut être utilisé en référence à un dispositif de *package* contenant le capteur, le transducteur et d'autres éléments de conditionnements du signal. Le contexte dans lequel le terme est utilisé évite toujours toute ambiguïté.

(2) Étape de conditionnement du signal : cette étape intermédiaire est optionnelle. Elle peut être utilisée pour modifier le signal du transducteur dans une grandeur désirée.

(3) Étape de l'output : cette étape fournit une indication de la valeur du mesurage. Exemple : l'échelle du thermomètre sert d'output pour ce système de mesure.

(4) Étape de feed-back –contrôle : cette étape contient un contrôleur qui interprète le signal mesuré et prend la décision selon le contrôle du processus. Cette décision résulte dans le changement d'un paramètre du processus qui affecte la grandeur de la variable détectée.

En résumé, la conception d'un instrument de mesure nécessite la sélection d'une méthode de mesure appropriée. Une fois que l'instrument de mesure est mis en place, il est nécessaire qu'il soit calibré, ajusté et réglé. En d'autres termes, dans un processus de vérification, il est nécessaire d'identifier les grandeurs à mesurer et les paramètres de contrôle (objectif de mesure); de développer la fonction de mesure (relation entre l'output et l'input) permettant de rencontrer l'objectif de mesure (procédure à suivre);

d'identifier l'instrument de mesure ou la méthode la plus appropriée en se basant sur les nouvelles normes de mesure, nous permettant la vérification de l'opération; de déterminer l'incertitude de mesure et le niveau d'exactitude; l'étalonnage permettra de réduire l'erreur de mesure; le processus de mesure doit tenir compte des caractéristiques de l'instrument de mesure (conditions d'utilisation, etc.); l'étalonnage nécessite la présence d'étalons de mesure.

4.3.5 Caractéristiques des instruments de mesure

L'objectif derrière un instrument de mesure est de fournir une indication exacte de la valeur du mesurande. Mais, en pratique aucun instrument ne permet de mesurer exactement la valeur de la grandeur à mesurer. La qualité de l'information qui résulte d'un mesurage dépend des caractéristiques de l'instrument, de l'erreur résultante lors du mesurage et aussi de l'interprétation des résultats mesurés. Il est important de comprendre les caractéristiques des instruments de mesure affectant le signal d'entrée et de sortie. Dans le VIM, le terme « instrument de mesure » est utilisé dans le sens général. Mais, les principaux concepts des « caractéristiques des instruments de mesure » sont applicables aussi bien à un appareil de mesure, à un dispositif de mesure, à un transducteur de mesure ou à un système de mesure, et à une mesure matérialisée ou à un matériau de référence. Nous avons proposé de classer les caractéristiques des instruments de mesure en caractéristiques d'utilisation et de contrôle dans la section 4.2.5 (Tableau XIV).

4.3.6 Étalons

4.3.6.1 Concepts clés

Un étalon est défini dans le VIM par une mesure matérialisée, un appareil de mesure, un matériau de référence ou un système de mesure destiné à définir, à réaliser, à conserver ou à reproduire une unité ou une ou plusieurs valeurs d'une grandeur pour servir de référence. L'étalonnage constitue l'ensemble des opérations établissant, dans des

conditions spécifiques, la relation entre les valeurs de la grandeur indiquées par un appareil de mesure ou un système de mesure, ou les valeurs représentées par une mesure matérialisée ou par un matériau de référence, et les valeurs correspondantes de la grandeur réalisées par des étalons. Lors d'un étalonnage, la valeur connue du signal d'entrée d'un système de mesure est l'étalon. L'étalon primaire sert comme référence pour déterminer l'exactitude du système de mesure. Mais pour des raisons pratiques, on utilise un étalon secondaire, qui est dupliqué à l'étalon primaire et qui peut être utilisé dans un objectif d'étalonnage. Cependant, le degré d'incertitude doit être accepté avec l'utilisation des étalons qui sont dupliqués aux étalons primaires.

4.3.6.2 Hiérarchie des étalons et étalonnage

Selon Figliola *et al.* [43], il existe une hiérarchie des étalons allant de l'étalon primaire jusqu'à l'étalon de travail (Tableau XXV).

Tableau XXV

Hiérarchie des étalons [43]

Hiérarchie des étalons
Étalon primaire (<i>primary standard</i>)
Étalon de transfert (interlaboratory transfer standard)
Étalon de référence (<i>local standard</i>)
Étalon de travail (<i>working instrument</i>)

Le Tableau XXV décrit la hiérarchie des étalons tels qu'ils peuvent être utilisés dans un laboratoire pour étalonner un système de mesure : chaque niveau subséquent de la hiérarchie est dérivé par l'étalonnage dans un niveau préalable élevé. À mesure qu'un étalon se déplace vers le bas de la hiérarchie, le degré d'exactitude par lequel cet étalon se rapproche de l'étalon primaire (par lequel il dérive) se détériore, c'est-à-dire que de nombreux éléments d'erreurs sont introduits dans l'étalon au fur et à mesure qu'il se déplace d'une génération d'étalons à une autre subséquente [43]. L'étalonnage de référence peut se faire par rapport à un étalon de transfert (traçabilité). Ainsi,

l'incertitude dans le rapprochement d'une valeur connue augmente au fur et à mesure qu'un étalon se déplace vers le bas de la hiérarchie. Cependant, puisque l'étalonnage permet de déterminer la relation entre la valeur de l'input et celle de l'output, l'exactitude d'étalonnage dépendra en partie de l'exactitude de l'étalon en question. Ainsi, l'exactitude ne peut être qu'estimée et la confiance dans cette estimation dépendra de la qualité de l'étalon et des techniques utilisées d'étalonnage.

4.3.6.3 Traçabilité d'un étalon

Quelles sont les caractéristiques métrologiques d'un étalon de référence? À partir d'un étalon de référence dérive la mesure, et son utilité dans un contexte nécessitant un état officiel (national ou international) ou une base largement reconnue (primaire ou secondaire) (Figure 28). L'étalon de transfert est utilisé comme intermédiaire pour comparer des étalons. Si l'intermédiaire n'est pas un étalon, on parle de dispositif de transfert.

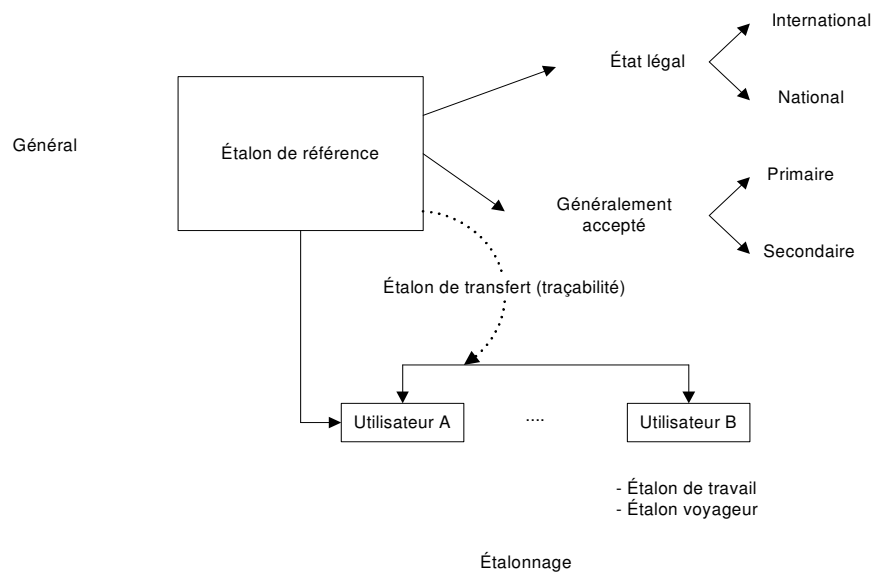


Figure 28 Liaison entre étalons

La traçabilité aux étalons nationaux permet d'assurer une harmonisation des mesures au niveau national et international et autorise ainsi une comparaison des résultats de mesure dans le temps. Le Tableau XXVI présente la classification des caractéristiques d'un étalon en termes de ses propriétés, son mode de conservation et son mode de matérialisation.

Tableau XXVI

Classification des caractéristiques d'un étalon

<i>Propriété d'un étalon</i> [44]	Mode de conservation	Mode de matérialisation
Homogène Bien défini (précis) Stable	Traçabilité Étalonnage Matériau de référence (MR) Matériau de référence certifié (MRC)	État légal Généralement reconnu

4.3.6.4 Commentaires additionnels sur les étalons et l'étalonnage

Quand un système de mesure est étalonné, il est comparé avec un étalon dont la valeur est supposée connue. Cet étalon peut être un objet ayant un attribut physique bien défini pour être utilisé pour des raisons de comparaison, ou il peut être une technique connue bien acceptée pour produire une valeur fiable. Ces étalons représentent le fondement de n'importe quel mesurage. Les étalons primaires sont nécessaires, car la valeur assignée à une unité est réellement arbitraire [43]. Que ce soit le mètre choisi (la longueur du bras d'un roi ou la distance de la lumière dans une trajectoire d'une fraction de seconde), il dépend seulement de comment on veut définir la mesure. Pour éviter toute confusion, les unités sont définies par un consensus et un accord international selon l'utilisation des étalons primaires. Une fois que le consensus est établi, l'étalon primaire forme la définition exacte de l'unité jusqu'à un changement ultérieur par accord. Selon Figliola *et al.* [43], les principales caractéristiques d'un étalon sont :

- validité globale;
- fiabilité continue;

- stabilité;
- avec un minimum de sensibilité aux sources environnementales externes.

Il est à noter aussi que la valeur du signal d'entrée n'est pas connue lors d'un mesurage mais peut être déduite à partir de la valeur du signal de sortie d'un instrument de mesure (cas d'estimation). C'est seulement lors de l'étalonnage que la valeur du signal d'entrée est réellement connue et permet ainsi de développer la relation entre les valeurs du signal d'entrée et de sortie, nécessaire pour interpréter le signal de sortie durant des mesurages ultérieurs [43]. C'est pour cette raison qu'il est nécessaire d'introduire dans la conception d'un instrument de mesure un moyen d'étalonnage reproductible pour simuler le type de signal d'être du stimulus lors des mesurages réels.

4.4 Sommaire et observations

Le document ISO de métrologie (VIM) représente un consensus international sur la terminologie cohérente pour l'ensemble des six catégories des concepts de la métrologie [57]. La métrologie qui constitue le domaine de connaissance de la mesure est la base pour le développement et l'utilisation des instruments de mesure et les processus de mesure dans tous les domaines scientifiques et d'ingénierie. Cependant, la structure de ce document est présentée sous la forme d'un glossaire, sans aucun ordre alphabétique et son système de classification de haut niveau classe les 120 termes de métrologie en six catégories, assignées à un chapitre séparé. Dans chaque chapitre du VIM, les termes sont présentés selon un « ordre approximatif de complexité croissante ». En plus de ces 120 termes explicitement définis, il y a 52 termes additionnels inclus dans la partie « Note » ou « Exemples » de certaines de ces définitions de l'Annexe C où on énumère les 172 définitions explicites et implicites du VIM.

Pour rendre plus facile l'utilisation et l'application des concepts fondamentaux de métrologie, nous avons proposé à la fois un modèle général de l'ensemble reconnu des concepts de métrologie ISO et un modèle plus détaillé pour permettre une

communication plus précise entre chercheurs à propos de concepts aussi variés et complexes que leurs relations.

Par ailleurs, pour faciliter la compréhension des relations à travers les 120 concepts individuels de mesure dans toutes les catégories et les différents niveaux d'abstraction, nous avons présenté dans ce chapitre un modèle de haut niveau de la classification du VIM, suivi par des modèles détaillés illustrant comment les termes individuels sont inclus dans ce modèle de haut niveau. D'ailleurs dans la troisième édition du VIM'2005 [74], il y a une tentative de modéliser les relations entre les concepts de mesure sous forme de diagrammes. Cependant, le contenu de ce document n'a pas encore reçu une autorisation finale d'ISO. Nous présentons dans l'Annexe B, la modélisation des travaux de la norme ISO en cours sur la métrologie.

Dans le contexte du génie logiciel, ce chapitre a présenté notre modélisation de l'ensemble des concepts de mesure documentés dans le VIM. En particulier, cette modélisation a montré qu'il y a différents niveaux d'abstraction des concepts ainsi que des relations entre les concepts et les sous-concepts de mesure. Nous avons référé ce type de représentation comme étant la topologie des concepts dans le VIM. D'autres travaux ont été réalisés pour inclure dans notre modélisation les concepts de mesure implicitement introduits dans la partie Note et Exemples du VIM pour s'assurer de la cohérence et de la consistance de représentation.

Cette représentation raffinée de modélisation va être utilisée dans les prochains chapitres pour examiner lesquels des concepts ont été effectivement discutés dans la littérature de mesure en génie logiciel, en plus de la perspective traditionnelle de la « théorie de mesure ». Par la suite, nous étudierons comment ces concepts de métrologie peuvent aider à comprendre l'état courant de maturité du domaine de la mesure du logiciel et ainsi contribuer à son processus continu d'amélioration.

L'introduction des concepts fondamentaux de métrologie nous permet d'apporter de nouvelles possibilités d'interprétation de mesure dans le domaine du génie logiciel, et même de contribuer éventuellement à l'évolution et à la maturation des mesures de logiciels. En particulier, certains des modèles détaillés peuvent suggérer comment de tels concepts de métrologie peuvent être utilisés pour concevoir de nouveaux instruments de mesure ainsi que la vérification (et la validation) de l'ensemble courant des mesures de logiciels proposés.

CHAPITRE 5

CADRE DE RÉFÉRENCE DE VÉRIFICATION D'UNE MESURE

5.1 Introduction

L'importance accordée aux mesures a contribué à la prolifération de propositions de nouvelles mesures en génie logiciel. Cependant, il y a très peu de discussion sur la compréhension de la mesure en soi et sur les hypothèses empiriques derrière plusieurs de ces nouvelles mesures et il y a peu d'information sur le niveau de vérification des mesures proposées et utilisées. Par conséquent, il est difficile pour les praticiens et les chercheurs d'obtenir une idée claire sur l'état de l'art pour choisir ou définir une mesure de logiciel.

Cette situation pourrait être améliorée en utilisant un cadre de référence permettant de réviser et de vérifier les mesures existantes. Selon notre approche, ce cadre doit être basé d'une part sur une terminologie cohérente de mesure provenant du VIM et d'autre part sur un modèle structuré du processus de mesure en logiciel de Abran et Jacquet. Ce chapitre décrira les quatre étapes composant ce cadre de référence : design d'une méthode de mesure, design de l'instrument de mesure, résultats de mesure et utilisation de ces résultats.

5.2 Alignement des principaux concepts du VIM avec les étapes du processus de mesure

Comme décrit dans le chapitre 2, le modèle du processus de mesure proposé par Abran et Jacquet identifie quatre étapes allant du design d'une méthode de mesure jusqu'à l'exploitation des résultats obtenus. Chacune des étapes est décomposée en d'autres étapes permettant ainsi une vision globale et une détaillée des concepts de mesure. En

effet, le terme « mesure » utilisé en génie logiciel se réfère souvent à plusieurs concepts distincts pour représenter à la fois :

- une méthode de mesure;
- une application de cette méthode;
- un résultat de cette application;
- un processus de mesurage.

Pour des raisons de clarification dans le vocabulaire spécialisé de mesure de logiciels, la signification du terme « mesure » est précisée par l'utilisation d'expressions telles que « méthode de mesure », « application d'une méthode de mesure », « résultat de mesure » et « processus de mesure ». Par exemple, le modèle présenté dans la Figure 29 décrit en détail les différentes étapes d'un processus de mesure du logiciel [6]. De même, dans le VIM, le terme « mesure » n'est jamais utilisé seul.

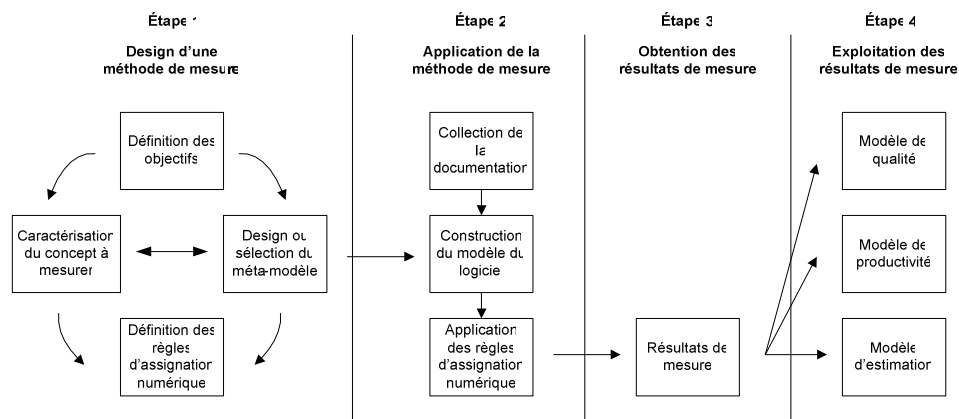


Figure 29 Processus de mesure – modèle détaillé [6]

Notre cadre de référence de mesure sera basé sur les éléments du processus de mesure décrit par Abran *et al.* auxquels sont ajoutés d'autres concepts fondamentaux de la métrologie. Ceci est illustré dans le Tableau XXVII qui établit une première mise en correspondance entre la Figure 2 et la Figure 6.

Tableau XXVII

Alignement des concepts de mesure

Type des connaissances de mesure & Métrologie VIM				
Modèle du processus de mesure (Abran et Jacquet, 1999)	Étape 1 Conception de la méthode de mesure	Étape 2 Application de la méthode de mesure	Étape 3 Obtention des résultats de mesure	Étape 4 Exploitation des résultats de mesure
Modèle du VIM (Abran et Sellami, 2002)	Grandeurs et unités Étalons	Mesurage Instruments de mesure Caractéristiques des instruments de mesure	Résultats de mesure	

Comme illustré dans le Tableau XXVII, deux des six catégories des concepts de métrologie (grandeurs et unités, et étalons) sont reliées à l'aspect design d'une méthode de mesure, tandis que les trois autres catégories (mesurage, instruments de mesure et caractéristiques des instruments de mesure) portent non pas sur le design en soi, mais sur l'application de la méthode par un instrument de mesure. Enfin, les résultats obtenus par l'intermédiaire de l'instrument de mesure portent sur l'obtention des résultats obtenus par l'application de la méthode de mesure. Il y a bien entendu, à un niveau plus détaillé, beaucoup plus de concepts dans le VIM. D'autre part, le modèle de processus de mesure de [6;76] est plus large que celui du VIM [12;57] mais il n'est pas décrit à un niveau aussi détaillé. En ce sens, il inclut beaucoup plus de concepts que les grandeurs et unités et étalons dans l'étape 1. L'étape 4 est celle qui représente le plus d'intérêt pour les gestionnaires en utilisant par exemple les modèles de productivité et d'estimation nécessaires à la gestion d'un projet logiciel (voir Figure 29), mais cette étape n'est pas incluse dans le VIM.

En résumé, le VIM [57] ne traite ni la totalité de design d'une méthode de mesure ni l'exploitation des résultats de mesure. Cependant, il traite beaucoup plus en détail des instruments de mesure et de leurs caractéristiques. Par conséquent, ces deux modèles

sont complémentaires et peuvent être combinés dans un outil analytique pour réviser et pour proposer des améliorations à différentes mesures en génie logiciel.

Dans le cadre de notre projet de recherche, nous utilisons le terme « vérification » plutôt que « validation ». Par exemple, nous nous référons au cadre de Kitchenham *et al.* [82] comme un cadre de vérification plutôt qu'un cadre de validation.

Nous proposons ainsi un cadre de référence de vérification d'une méthode de mesure qui intègre les concepts du VIM avec les concepts fondamentaux des mesures spécifiques au logiciel. Un tel cadre devrait rencontrer les différentes étapes du processus de mesure, et énoncer clairement les quatre concepts de mesure à vérifier (Tableau XXVIII), et en utilisant les différents critères pertinents à chaque entité : (1) design d'une méthode de mesure, (2) design de l'instrument de mesure, (3) les résultats de mesure, et (4) l'utilisation des résultats de mesure dans un contexte spécifique pour un objectif spécifique. Les résultats de chaque étape nécessitent d'être vérifiés avant de passer à l'étape suivante.

Tableau XXVIII

Différentes entités de mesure sujettes aux critères de vérification

Cadre de référence de vérification d'une mesure			
Étape 1 Design	Étape 2 Application	Étape 3 Résultat	Étape 4 Utilisation
Vérification du design d'une méthode de mesure	Vérification du mesurage, incluant le design de l'instrument de mesure	Vérification des résultats obtenus du mesurage	Vérification de l'utilisation des résultats de mesure

Le Tableau XXVIII spécifie les différents concepts identifiés dans chacune des quatre étapes de vérification de mesure pour :

- faciliter la comparaison entre les mesures existantes;
- faciliter l'évaluation des mesures existantes d'un point de vue de la métrologie;

- supporter la définition de nouvelles mesures de logiciels et la sélection parmi des mesures existantes.

5.3 Sommaire et observations

Dans ce chapitre, nous avons proposé un cadre de référence de vérification pour les méthodes de mesure en nous basant autant sur les concepts fondamentaux du VIM que sur ceux du processus de mesure proposé par Abran *et al.* Ce cadre vise à :

- identifier un inventaire plus exhaustif des critères pour les mesures proposées;
- mieux vérifier la qualité des mesures proposées;
- mieux construire les mesures en génie logiciel;
- évaluer la conformité des concepts de mesures de logiciels par rapport aux concepts du VIM.

Le cadre de référence proposé permet aussi d'organiser le processus de mesure de telle sorte qu'on peut s'assurer des résultats à utiliser s'ils sont obtenus sur une base vérifiable. Nous avons utilisé les concepts du VIM qui sont nécessaires pour le mesurage d'une façon générale et la mesure du logiciel en particulier. Nous avons remarqué que l'application des concepts de la métrologie au logiciel a été peu utilisée en mesure du logiciel, incluant très peu de méthodes analytiques existantes pour quantifier l'incertitude du mesurage des logiciels.

Chacune des sources incluses (VIM-Abran *et al.*) ont leurs forces mais elles ne couvrent pas individuellement tout le champ d'intérêt. Nous avons proposé ainsi de combiner les forces de ces sources pour obtenir un cadre de vérification plus précis et plus exhaustif.

CHAPITRE 6

UTILISATION DU CADRE DE RÉFÉRENCE POUR ANALYSER LES NORMES ISO DE MESURE

6.1 Introduction

Au niveau de la normalisation internationale pour la mesure des logiciels, il existe des normes pour deux types de mesure des logiciels : dans le domaine de la qualité du logiciel, il y a différents types de mesure des logiciels identifiés dans ISO 9126 (attributs de qualité), des mesures de la taille fonctionnelle de logiciels dont les critères ont été définis dans ISO 14143 (sur les mesures de la taille fonctionnelle) et quatre méthodes spécifiques dans des normes ISO distinctes.

La méthode COSMIC-FFP (ISO 19761) est la première méthode de taille fonctionnelle qui est conçue par un groupe international d'experts sur une base théorique, inspirée des expériences pratiques de toutes les méthodes existantes de points de fonction, désignée à être conforme à la norme ISO 14143-1 et désignée à être appliquée dans tous les domaines de gestion des logiciels.

Dans ce chapitre, nous analysons la qualité même de ces types de mesure en utilisant le cadre de référence de mesure tel que présenté dans la section 5.2 du chapitre 5. Nous examinerons d'abord le design des mesures proposées pour quantifier les attributs de la qualité d'un produit logiciel (dans les séries 9126-2 à 4) particulièrement pour vérifier quelle est la méthode ISO de mesure de la qualité d'un produit logiciel, quels sont les mesures reconnues par ISO et comment les appliquer.

Par la suite, nous abordons l'analyse de la norme ISO 14143. Enfin nous vérifions la méthode COSMIC selon le modèle du VIM.

6.2 ISO/IEC 9126 : Qualité d'un produit logiciel

Depuis quelques décennies, plusieurs mesures de la qualité d'un produit logiciel ont été proposées pour permettre un meilleur contrôle du processus de développement de logiciels. Le sous-comité SC7 du comité JTC1 de ISO/IEC (*International Electronic Commission*) a publié un ensemble de normes et de rapports techniques (ISO/IEC 9126) portant sur la mesure de la qualité d'un produit logiciel : ainsi, ISO 9126-1 [60] spécifie les caractéristiques et les sous-caractéristiques de la qualité d'un produit logiciel, tandis que les mesures proposées par ISO pour la mesure de la qualité du logiciel sont documentées dans les rapports techniques ISO TR 9126-2 à 4 [71-73].

Cette série de documents ISO classe les caractéristiques de qualité d'un produit logiciel pour permettre de les mesurer en utilisant un modèle commun (plus précisément, un ensemble de modèles). Cela veut dire, qu'en utilisant l'ensemble des documents ISO 9126 [60;71-73], on devrait être capable d'utiliser le même vocabulaire et les mêmes modèles pour déterminer les caractéristiques et les sous-caractéristiques de la qualité d'un produit logiciel, ainsi que de choisir les mesures pour les quantifier.

Toutefois ces documents ISO ont été développés en isolation par le sous-comité du génie logiciel. Ces documents ne font pas référence au corpus des connaissances en métrologie [57]. L'analyse structurée de cette norme ISO l'illustre en prenant comme exemple la caractéristique « productivité » définie dans 9126-4 [73] et explore quelques-unes des différentes mesures proposées pour mesurer cette caractéristique.

6.2.1 Survol d'ISO 9126

La norme ISO/IEC 9126-1 définit un méta-modèle global de la qualité d'un produit logiciel, laquelle qualité peut être vue selon trois perspectives : qualité interne, qualité externe et qualité en utilisation (telles que présentées dans le Tableau XXIX). Chacun des modèles de qualité interne et externe se décompose en caractéristique et en sous-caractéristiques. La structure des modèles de qualité ISO pour la qualité interne et

externe est la même au niveau des six caractéristiques et leurs sous-caractéristiques. Ces sous-caractéristiques de la qualité peuvent être évaluées par des *métriques* internes ou externes. Le modèle de la qualité en utilisation spécifie uniquement quatre caractéristiques (efficacité, productivité, sûreté et satisfaction) mais n'élabore pas un niveau plus bas que le niveau des caractéristiques.

Tableau XXIX

Structure des caractéristiques et des sous-caractéristiques de la qualité dans ISO/IEC 9126

Méta-modèle de qualité (ISO 9126-1)		
Qualité externe et interne (ISO 9126-2 et ISO 9126-3)		Qualité en utilisation (ISO 9126-4)
Caractéristiques	Sous-caractéristique	Caractéristiques
Fonctionnalité	Convenance Exactitude Interopérabilité Sécurité Conformité de fonctionnalité	Efficacité Productivité Sûreté Satisfaction
Fiabilité	Maturité Tolérance de fautes Récupérabilité Conformité de la fiabilité	
Rentabilité (<i>Usability</i>)	Compréhensible Apprenabilité Opérabilité Attraction Conformité de la rentabilité	
Efficience	Comportement du temps Utilisation de ressource Conformité de l'efficience	
Maintenabilité	Analysabilité Variabilité Stabilité Testabilité Conformité de la maintenabilité	
Portabilité	Adaptabilité Installabilité Remplaçabilité Coexistence Conformité de la portabilité	

Par ailleurs, l'objet des rapports techniques TR 9126- 2 à 4 est de fournir un inventaire de mesures approuvées par ISO pour spécifier et évaluer quantitativement la qualité d'un produit logiciel pour chacune des caractéristiques de qualité. Ces mesures sont énumérées dans les rapports techniques de ISO/IEC 9126-2 (mesures de la qualité externe), 9126-3 (mesures de la qualité interne) et 9126-4 (mesures de la qualité en utilisation). Ces rapports techniques ne prétendent pas toutefois fournir un inventaire exhaustif de toutes les mesures pour les attributs de qualité mais uniquement celles pour lesquels un consensus minimum international a été établi au niveau ISO; il est important de noter que ce consensus minimum n'a pas été suffisant pour reconnaître à ces documents le statut d'une norme internationale, mais uniquement un statut de rapport technique mis à la disposition du public pour fin d'information et pour contribuer à l'évolution des connaissances dans ce domaine.

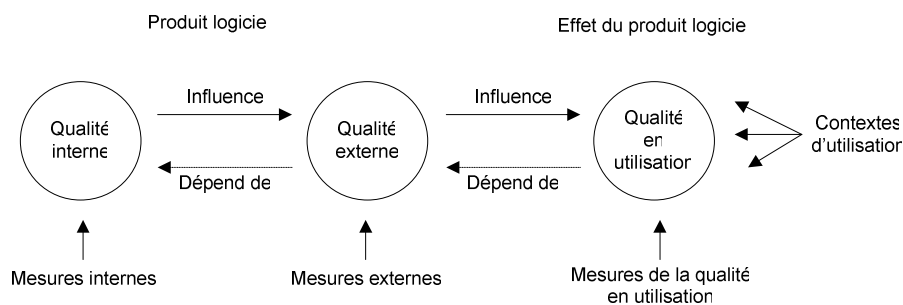


Figure 30 Relation entre les types de mesure [60]

La série 9126 est basée sur le modèle décrit dans la Figure 30 où la qualité interne du produit logiciel influence la qualité externe qui influence, à son tour, la qualité en utilisation. De plus, les mesures de la qualité interne, qui s'appliquent au produit logiciel non exécutable durant ses étapes de développement, ont comme objectif de mesurer la qualité des livrables intermédiaires; l'hypothèse est que cette qualité interne devrait être un indicateur de la qualité du produit final. Les mesures de la qualité externe peuvent être utilisées pour mesurer la qualité du produit logiciel en mesurant le comportement de son système. Ces mesures s'appliquent durant les étapes de tests et les étapes

opérationnelles du cycle de vie du logiciel. Une autre hypothèse est que les mesures de la qualité interne ayant un lien avec les mesures de la qualité externe peuvent être utilisées pour prévoir les valeurs des mesures externes. Les mesures de la qualité en utilisation sont définies pour mesurer l'étendue pour lequel le produit logiciel rencontre les exigences spécifiées de l'utilisateur pour atteindre les objectifs avec efficacité, productivité, sûreté et satisfaction dans un contexte spécifique d'utilisation.

Comment vérifier la qualité du design de ces multiples mesures proposées par ISO? Tout d'abord, dans le document ISO 9126-2 à 4, l'expression utilisée est celle d'un terme non usuel en sciences et en génie, « métrique » de logiciels à la place des termes plus précis et bien défini qui sont requis pour préciser lesquels des multiples sous-concepts de « mesure » qui sont pertinents dans divers contextes (e.g. méthode de mesure, résultats de mesure, application de la mesure, utilisation des résultats de mesure). Dans cette série ISO 9126, l'utilisation du terme « métrique » est ambiguë et désigne plusieurs significations différentes (telles qu'un modèle, une méthode, un design de mesure), et porte donc le lecteur à confusion. Une analyse de ces documents ISO permet de constater que, certaines de ces « métriques » ont comme objectif de mesurer un attribut d'une façon indépendante du contexte (Mesures de base⁵) alors que d'autres portent sur un ensemble (e.g. des combinaisons) de plusieurs attributs, qui sont dépendantes des mesures de base (Mesure dérivée⁶).

Il est important d'observer que ces rapports techniques ISO ne documentent toutefois pas les critères métrologiques de qualité de ces mesures, tels que : l'exactitude, la répétitivité et la reproductibilité, et qu'il n'y a aucune démonstration que les mesures répertoriées dans ces rapports rencontrent effectivement les critères de la métrologie tels

⁵ La mesure de base est définie selon ISO 15939 [65] comme étant « une mesure définie en termes d'un attribut et la méthode de sa quantification ». Cette définition se base à son tour sur le concept « grandeur de base » du VIM.

⁶ La mesure dérivée est définie selon ISO 15939 [65] comme étant « une mesure définie comme une fonction de deux ou plusieurs valeurs des mesures de base ». Cette définition se base sur le concept « grandeur dérivée » du VIM.

que généralement acceptés dans les autres disciplines. Pour notre étude des mesures proposées par ISO, nous allons analyser le contenu des séries ISO 9126-2 à 4 [71-73] pour vérifier si cette documentation porte sur toutes les étapes du modèle du processus de mesure, si les concepts de la métrologie sont couverts et lesquels.

6.2.2 Analyse de haut niveau de ISO 9126-2 à 4

Une première analyse de haut niveau montre que les catégories des mesures de 9126-2 à 4 [71-73] sont toutes documentées en utilisant la même structure de documentation :

- nom de la mesure;
- objectif de la mesure;
- méthode d'application;
- mesure, formule et comptage des éléments de données;
- interprétation d'une valeur mesurée;
- type d'échelle de mesure;
- type de mesure (*dimension*);
- intrant au mesurage;
- référence dans la phase du cycle de vie (SLCP) de ISO 12207;
- audience cible.

La mise en correspondance de ISO 9126-2 à 4 par rapport au cadre de référence proposé au Tableau XXVIII est présentée dans le Tableau XXX. Chaque ligne du Tableau XXX représente les éléments de mesure identifiés dans les rapports techniques ISO TR 9126-2 à 4 [71-73]. Un « X » indique qu'un élément de mesure est couvert par le cadre de référence dans la colonne respectée. Ainsi, l'étape 3 : résultats de mesure qui sont normalement obtenus après l'application des règles de la méthode de mesure ne sont pas couverts dans les rapports techniques 9126-2 à 4 [71-73]. Dans un premier temps, cela veut dire que ces rapports techniques ne documentent pas toutes les étapes distinctes requises pour le processus de mesure.

Tableau XXX

Mise en correspondance de haut niveau de ISO 9126-2 à 4 et cadre de référence

Cadre de référence	Étape 1 Design	Étape 2 Application	Étape 3 Résultats	Étape 4 Utilisation
ISO 9126 -2 à 4				
Nom de la mesure	X			
Objectif de la mesure	X			
Méthode d'application		X		
Mesure, formule et comptage des éléments de données	X			
Interprétation d'une valeur mesurée				X
Type d'échelle de mesure	X			
Type de mesure	X			
Input au mesurage		X		
ISO 12207 référence SLCP		X		
Audience cible	X			X

6.2.3 Analyse détaillée de ISO 9126-2 à 4

La série 9126-2 à 4 [71-73] présentent plus de 200 mesures dérivées décrites sur 1 ou 2 lignes seulement (+200 concepts de mesure) avec des formules de comptage.

Nous avons également observé que toutes les mesures de base dans les séries 9126-2 à 4 spécifient un comptage pour l'assignation des nombres, laquelle assignation numérique utilisée ne peut être qualifiée comme « méthode de mesure » selon la définition ISO de la métrologie VIM [57] d'une méthode de mesure en tant qu'une succession logique des opérations, décrites d'une manière générique, mises en œuvre lors de l'exécution de mesurages.

À titre illustratif, nous présentons comme exemple l'analyse des mesures proposées pour deux sous-caractéristiques d'une caractéristique (soit la « fonctionnalité ») dans le modèle ISO 9126 de la « qualité en utilisation » [73] :

- la « productivité »;
- l'« efficacité ».

Dans le modèle ISO de la « qualité en utilisation » [73], la productivité est définie comme la mesure des ressources (telles que le temps de la tâche, l'effort de l'utilisateur, le coût du matériel, etc.) allouées aux utilisateurs en relation avec l'efficacité atteinte dans un contexte spécifique d'utilisation⁷. La productivité est ainsi définie par un ratio en utilisant l'unité de temps (exemple : productivité = Points de Fonction/homme-mois). En science économique, la productivité est définie par le taux de l'output par unité de l'input utilisé. Selon Fenton [41], l'idée de comparer l'input par rapport à l'output est importante pour les développeurs de logiciels. Cependant, la productivité ne peut être considérée en tant qu'une mesure de base, c'est plutôt une mesure dérivée (productivité = output/input = taille/effort où les mesures de base sont celles de l'output et de l'input).

Pour obtenir une mesure dérivée, il faut tout d'abord obtenir indépendamment chacune des mesures de base qui la compose. Pour chaque mesure dérivée, chaque mesure de base est déterminée individuellement des autres (exemple : dans la mesure dérivée de la productivité, la mesure de l'input (transformée en chiffre) est obtenue, sans avoir connaissance de la valeur de la mesure de base de l'output).

L'utilisation du ratio de productivité comme une mesure du type d'échelle ratio permet de travailler avec le raisonnement de l'analyse statistique et les opérations arithmétiques admissibles pour ce type d'échelle. Mais un ratio ne peut être considéré comme une méthode de mesure en soi. Une méthode de mesure est cependant essentielle pour obtenir une mesure de base qui puisse avoir des qualités métrologiques telles que la répétitivité, la reproductibilité et la précision.

Un exemple d'une mesure de base de ISO/IEC 9126 qui est utilisée dans les mesures dérivées (productivité, efficacité) est la mesure du temps de la tâche. Les correspondances de cette mesure par rapport aux concepts du cadre de référence sont

⁷ La productivité, selon Fenton, est considérée comme la production d'un ensemble de composants dans une période déterminée de temps.

indiquées par des « X » dans le Tableau XXXI. Ainsi, l'objectif de détermination du temps de la tâche concerne la définition des objectifs de l'étape 1 du cadre de référence. La plupart des éléments (mesure, formule et comptage des éléments de données, type d'échelle de mesure et type de mesure) adressent les règles d'assignation numérique de l'étape 1 de design d'une mesure, et l'audience cible (les utilisateurs et les concepteurs de l'interface) porte sur la caractérisation de l'attribut à mesurer de l'étape 1 et l'utilisation des résultats de l'étape 4. Par contre, le méta-modèle n'est pas spécifié. La méthode d'application et les références selon ISO 12207 portent sur l'application de ces règles d'assignation, et l'input au mesurage en termes de rapport de l'opération adresse la collection de la documentation de l'étape 2. L'étendue des valeurs fournies pour leurs interprétations adresse l'étape 4 pour exploiter les résultats trouvés dans des modèles analytiques, qualitatifs ou quantitatifs. Nous remarquons également que les concepts reliés aux résultats de mesure ne sont pas explicitement définis. Cela veut dire que la structure des éléments de mesure dans ISO 9126-2 à 4 n'est pas assez complète.

Tableau XXXI

Mesure du « temps de la tâche » pour la caractéristique de « productivité » dans 9126-4

Cadre de référence ISO/IEC TR 9126-4		Étape 1 Design	Étape 2 Application	Étape 3 Résultats	Étape 4 Utilisation
Nom de la mesure	Temps de la tâche	X			
Objectif de la mesure	Combien de temps alloués pour compléter une tâche ?	X			
Méthode d'application	Test de l'utilisateur		X		
Mesure, formule et comptage des éléments de données	$X = Ta$ avec Ta : temps de la tâche	X			
Interprétation d'une valeur mesurée	$0 \leq X$ Le plus petit est le meilleur				X

Tableau XXXI (Suite)

Cadre de référence		Étape 1	Étape 2	Étape 3	Étape 4
ISO/IEC TR 9126-4		Design	Application	Résultats	Utilisation
Type d'échelle de mesure	Intervalle	X			
Type de mesure	T= temps	X			
Input au mesurage	Rapport de l'opération (test) Enregistrement du contrôle de l'utilisateur		X		
Référence SLCP de l'ISO 12207	Clause 6.5 validation Clause 5.3 test de qualification Clause 5.4 opération		X		
Audience cible	Utilisateur Concepteur de l'interface	X			X

Le deuxième exemple choisi dans le modèle ISO de la « qualité en utilisation », c'est la mesure de l'efficacité (*efficiency*) de la tâche : cette mesure est définie dans ISO TR 9126-4 [73] pour mesurer la proportion de l'objectif atteint pour chaque unité de temps et elle est définie par un ratio⁸. Cette mesure dans ISO TR 9126-4 est exprimée par la formule suivante : $X = \frac{M_1}{T}$ avec M_1 : efficacité de la tâche et T : temps de la tâche.

Dans la formule ci-dessus, X représente l'attribut mesuré et $\frac{M_1}{T}$ représente la formule utilisée qui est composée d'autres attributs. Autrement dit, au nom de l'attribut est associé une formule et dans ce cas-ci, la formule est un ratio qui donne un nombre, et il y a des conditions pour l'analyse et l'interprétation de ce nombre. Mais, une formule

⁸ Lorsqu'on obtient une valeur élevée du ratio, cela indique que la proportion élevée de la tâche est atteinte dans un petit intervalle de temps : donc, cela permet de faire des comparaisons.

mathématique (ou algébrique) ne correspond pas à une méthode de mesure telle que définie en métrologie, et documentée dans le Vocabulaire ISO en métrologie (VIM)⁹.

D'autre part, l'attribut mesuré T représente le temps qui est pris comme une grandeur de base dans plusieurs domaines scientifiques, reconnue par le SI et lui est attribué l'unité de base (seconde) : c'est une mesure bien définie qui rencontre la plupart des différentes étapes du processus de mesure (Tableau XXXI). Mais, elle porte plutôt sur le projet (processus de développement) et non pas directement sur la mesure du logiciel. Par contre, M_1 qui représente l'efficacité de la tâche (Tableau XXXIII) correspond à une « mesure dérivée », elle-même composée d'autres mesures de base (Tableau XXXII et Tableau XXXIII); ces tableaux présentent en ligne les éléments identifiés dans 9126-4 et correspondants à la mesure des attributs composés de l'attribut « productivité ». Pour chaque ligne, nous indiquons les sous-étapes correspondantes aux étapes couverts du cadre de référence dans la colonne respective. Il est à noter que le Tableau XXXII utilise M_1 mais ne le définit pas directement.

Dans la structure des mesures de 9126 [71-73], on remarque que la plupart des mesures sont définies par une seule formule algébrique, sans référence à des méthodes de mesure, et encore moins à des méthodes qui permettent d'obtenir des résultats de mesure qui soient précis, reproductibles et répétables.

⁹ Dans le VIM, une méthode de mesure est définie par une « succession logique des opérations, décrites d'une manière générique, mises en œuvre lors de l'exécution de mesurages ».

Tableau XXXII

Mesure de « l'efficacité de la tâche » pour la « productivité » dans TR 9126-4

Cadre de référence ISO/IEC TR 9126-4		Étape 1 Design	Étape 2 Application	Étape 3 Résultats	Étape 4 Utilisation
Nom de la mesure	Efficiéce de la tâche				
Objectif de la mesure	Comment les utilisateurs sont-ils efficients ?	Définition des objectifs			
Méthode d'application	Test de l'utilisateur		Application des règles d'assignation numérique		
Mesure, formule et comptage des éléments de données	$X = \frac{M_1}{T}$ Avec M_1 : efficacité de la tâche T : temps de la tâche	Règles d'assignation numérique			
Interprétation d'une valeur mesurée	$0 \leq X$ Le plus grand est le meilleur				Exploitation des résultats dans un modèle
Type d'échelle de mesure					
Type de mesure	T= temps X=	Règles d'assignation numérique			
Input au mesurage	Rapport de l'opération (test) Enregistrement du contrôle de l'utilisateur		Collection de la documentation		
Référence SLCP de l'ISO 12207	Clause 6.5 validation Clause 5.3 test de qualification Clause 5.4 opération		Application des règles d'assignation numérique		
Audience cible	Utilisateur Concepteur de l'interface	Caractérisation de l'attribut à mesurer			Exploitation des résultats dans un modèle

La mesure de l'« efficacité de la tâche » (Tableau XXXII) est utilisée comme un indicateur, et non comme une représentation réelle de l'attribut d'une entité donnée (du logiciel). Par exemple, pour la mesure de l'efficacité de la tâche M_1 , il est mentionné que chaque composant potentiellement manquant ou incomplet lui est associé un poids A_i basé sur l'écart déduit de la valeur de l'output de l'utilisateur. Dans ce contexte, les poids mentionnés ne se réfèrent à aucune mesure.

Tableau XXXIII

Mesure de « l'efficacité de la tâche M_1 » pour la caractéristique « efficacité » dans 9126-4

Cadre de référence ISO/IEC TR 9126-4		Étape 1 Design	Étape 2 Application	Étape 3 Résultats	Étape 4 Utilisation
Nom de la mesure	Efficacité de la tâche				
Objectif de la mesure	Quelle est la proportion des objectifs atteints correctement de la tâche ?	Définition des objectifs			
Méthode d'application	Test de l'utilisateur		Application des règles d'assignation numérique		
Mesure, formule et comptage des éléments de données	$M_1 = \left 1 - \sum A_i \right $ Avec A_i : valeur proportionnelle de chaque composant manquant ou incorrect dans l'output de la tâche	Règles d'assignation numérique			
Interprétation d'une valeur mesurée	$0 \leq M_1 \leq 1$ La plus proche de 1 est la meilleure				Exploitation des résultats dans un modèle
Type d'échelle					

Tableau XXXIII (Suite)

Cadre de référence ISO/IEC TR 9126-4		Étape 1 Design	Étape 2 Application	Étape 3 Résultats	Étape 4 Utilisation
Type de mesure	$A=$	Règles d'assignation numérique			
Input au mesurage	Rapport de l'opération (test) Enregistrement du contrôle de l'utilisateur		Collection de la documentation		
Référence SLCP de l'ISO 12207	Clause 6.5 validation Clause 5.3 test de qualification Clause 5.4 opération		Application des règles d'assignation numérique		
Audience cible	Utilisateur Concepteur de l'interface	Caractérisation de l'attribut à mesurer			Exploitation des résultats dans un modèle

D'après cette analyse, nous remarquons que la caractérisation du concept à mesurer n'est pas simple. D'ailleurs, Zuse se réfère à cette difficulté par la « limite de l'intelligence » [114]. La mise en correspondance par rapport à l'étape 1 du cadre de référence montre que le concept relié à l'objectif de mesure de l'attribut à mesurer est énoncé (i.e. le nom donné à la mesure – exemple : la partie gauche de l'équation algébrique) tandis que les règles d'assignation numérique sont partiellement adressées en termes des formules et du type de mesure. Cependant, les concepts reliés à la catégorie « grandeurs et unités » ne sont pas définis pour ces mesures dans ISO TR 9126-2 à 4 [71-73]. Le méta-modèle de la mesure elle-même, qui permet de structurer les entités types du produit logiciel et les règles d'identification de ces entités, ne sont pas documentées ni discutées. Il est alors plausible que l'application de ces mesures, sans description de leurs méta-modèles et des règles d'identification, puisse conduire à des résultats erronés de mesure, car l'utilisateur sera amené à interpréter selon sa propre perspective les différents termes contenus dans la formule algébrique. Le Tableau XXXIV présente un récapitulatif de l'analyse détaillée de ISO 9126-2 à 4 [71-73]. Parmi les forces de ISO TR 9126-2 à 4 :

- elle décrit informellement par le positionnement d'une mesure à l'intérieur de la structure caractéristique/sous-caractéristiques d'un des niveaux de mesure (interne, externe et en utilisation);
- elle fournit des formules de « comptage » et des unités implicites. Les données sont identifiées selon les références ISO 12207;
- elle prescrit une formule algébrique pour un type de mesure donné et définit un méta-modèle des caractéristiques de la qualité pour 9126-1.

Par contre, les rapports techniques ne spécifient pas la structure du méta-modèle (du design d'une mesure) nécessaire pour identifier les caractéristiques de représentation d'un logiciel ou une partie du logiciel ainsi que l'ensemble de leurs relations. Dans ces rapports, il y a une prescription des formules algébriques et non pas des méthodes de mesure. De plus, la dimension de mesure n'est pas spécifiée pour vérifier s'il s'agit d'une grandeur de base ou d'une dérivée.

Tableau XXXIV

Récapitulatif de l'analyse détaillée de TR 9126-2 à 4

Étapes et sous étapes du cadre de référence	ISO/IEC TR 9126-2 à 4	
	Forces	Faiblesses
Étape 1 : design d'une méthode de mesure		
Définition des objectifs	Décrit informellement par le positionnement d'une mesure à l'intérieur de la structure caractéristique/sous-caractéristiques d'un des niveaux de mesure (interne, externe et en utilisation)	
Design ou sélection du méta-modèle		Ne spécifie pas la structure du méta-modèle
Caractérisation de l'attribut à mesurer		Prescrit une formule algébrique et non pas une méthode de mesure L'entité à mesurer, ainsi que son attribut, est implicite

Tableau XXXIV (Suite)

Étapes et sous étapes du cadre de référence	ISO/IEC TR 9126-2 à 4	
	Forces	Faiblesses
Définition des règles d'assignation numérique avec la définition des grandeurs et unités	Fournit des formules de « comptage » Fournit des unités implicites	Ne spécifie pas la dimension de mesure : s'il s'agit d'une grandeur de base ou dérivée
Référence à étalon		N'est pas défini
Étape 2 : Application de la méthode de mesure		
Collection de la documentation	Identifie la source des données utilisée et selon les références ISO 12207	
Construction du modèle logiciel		N'est pas adressé explicitement
Application des règles d'assignation numérique	Prescrit une formule algébrique	
Utilisation d'un instrument de mesure		N'est pas défini
Étape 3 : Résultat de mesure		
Vérification des résultats obtenus selon les critères du VIM	Utilisation du type de mesure pour documenter les résultats obtenus	
Étape 4 : Exploitation des résultats de mesure		
Utilisation de modèles qualitatifs ou quantitatifs	Utilisation du méta-modèle des caractéristiques de qualité de 9126-1	

6.3 Requis de métrologie

Est-ce que les designs de ces mesures peuvent donner des résultats de mesures robustes en termes de qualité métrologique? Plusieurs concepts de la métrologie n'ont pas été documentés dans les documents ISO 9126 portant sur la mesure de la qualité du logiciel (Tableau XXXIV). Comme exemple, il manque, pour toutes les mesures proposées, les unités de mesure, les méthodes de mesure, les étalons, la réalisation des références et la traçabilité par une chaîne d'étalonnage. Tous ces concepts ont été jugés comme nécessaires dans le domaine de connaissance de la mesure dans diverses disciplines scientifiques et du génie.

Tous ces concepts devraient être requis dans le domaine de mesure de la qualité du produit logiciel. La définition des unités de mesure est importante lors de mesurage et pour l'utilisation des types d'échelle de mesure, l'assignation des règles numériques, de même que l'utilisation des étalons.

Le cadre de référence de mesure a donc été utile pour analyser la qualité des « métriques » proposées pour mesurer les attributs de la qualité de 9126. Il nous a permis d'identifier les problèmes de mesure à résoudre. Les « métriques » pour mesurer les attributs de mesure de la qualité dans 9126-2 à 4 ne sont donc pas complètement définis. Malgré ces faiblesses les documents ISO 9126 demeurent importants dans le développement des mesures de la qualité de logiciel.

6.4 ISO/IEC 14143 : Méthodes de mesure de la taille fonctionnelle du logiciel

La mesure de la taille fonctionnelle (MTF) est un type de mesure utilisé pour mesurer la taille du logiciel en quantifiant les requis fonctionnels de l'utilisateur. Plusieurs méthodes se basant sur les fonctionnalités de logiciel ont été développées. La première méthode d'analyse des points de fonction (FPA) fut introduite en 1979 par A. Albrecht de IBM. Cette méthode a été ensuite raffinée par les groupes IFPUG, MKII, NESMA et COSMIC-FFP. De nombreuses variations ont été proposées par les chercheurs mais ces variations ne se sont pas rendues jusqu'à un statut de normes ISO. L'utilisateur se trouve ainsi perplexe quant au choix à faire parmi ces méthodes. La série des documents ISO/IEC 14143-1 à 5 [62-66] a été développée pour fournir un processus de vérification de l'étendue auquel certaines propriétés de performance des méthodes MTF. Ce processus permet d'aider l'utilisateur final à choisir la méthode la plus appropriée à ses besoins. Quelles sont les caractéristiques décrites dans la série ISO 14143-1 à 5 [62-66]?

6.4.1 Survol de ISO/IEC 14143

L'objectif derrière le développement des documents ISO 14143 [62-66] consiste à identifier et documenter les critères de design des méthodes MTF qui sont requis pour

qu'une méthode de mesure de la taille fonctionnelle du logiciel puisse être reconnue comme une norme internationale. La série de documents ISO 14143-1 à 5 est composée de cinq parties :

ISO 14143-1 « définition des concepts » [62] spécifie la définition d'un ensemble de concepts requis par ISO pour reconnaître une mesure de la taille fonctionnelle. Cette perspective de définir des critères spécifiques est unique parmi les normes ISO de mesures en génie logiciel.

ISO 14143-2 « évaluation de la conformité d'une méthode MTF » [63] spécifie un ensemble de procédures pour évaluer une méthode MTF en terme de sa conformité par rapport à ISO 14143-1. L'objet de 14143-2 consiste à :

- établir un cadre pour l'évaluation de la conformité d'une méthode MTF candidate par rapport aux provisions de 14143-1;
- décrire un processus d'évaluation de la conformité de la méthode MTF candidate par rapport aux exigences (type) de 14143-1, si elle remonte le même type;
- décrire les exigences pour accomplir l'évaluation de la conformité pour s'assurer de la répétabilité du processus d'évaluation ainsi que la consistance des décisions sur la conformité et le résultat final;
- s'assurer que l'output du processus d'évaluation de la conformité est objectif, partagé, consistant, répétable, complet et « auditable »;
- fournir des guides informatifs pour déterminer la compétence des équipes d'évaluation de la conformité;
- fournir un exemple de vérification pour aider à l'évaluation de la conformité d'une méthode MTF candidate;
- fournir un exemple de gabarit pour le rapport d'évaluation de la conformité. Les évaluations de la conformité sont conduites par une équipe d'évaluation de la conformité ayant des compétences décrites dans 14143-2.

ISO 14143-3 « guide de vérification » [64] spécifie un ensemble structuré de caractéristiques qui devrait être documenté pour toute MTF proposée en tant qu'une méthode de mesure. ISO 14143-3 consiste à :

- établir un cadre pour la vérification de certaines propriétés de performance d'une méthode MTF;
- définir plusieurs propriétés de performance auxquelles une méthode MTF peut être vérifiée;
- décrire les types de tests qui peuvent être accomplis;
- définir le processus de vérification d'une méthode MTF;
- fournir un modèle d'un exemple pour la vérification du rapport.

La vérification est conduite par une équipe de vérification qui a les compétences décrites dans ISO 14143-3. Ainsi, les propriétés de performance portent sur la répétabilité, la reproductibilité, l'exactitude, la convertibilité, le seuil de tolérance et l'applicabilité aux domaines fonctionnels. L'objet de 14143-3 est de s'assurer que l'output de vérification est objectif, impartial, consistant et répétable.

L'objectif derrière l'application de 14143-3 est de permettre à l'utilisateur de sélectionner une méthode MTF qui rencontre le « meilleur » de ses exigences. Ainsi, le résultat de la vérification d'une telle méthode devrait indiquer :

- l'étendue à laquelle une propriété de performance est rencontrée;
- si la propriété de performance rencontre l'étendue énoncée.

Pour cette raison, une méthode MTF peut être considérée vérifiée ou non vérifiée pour une propriété de performance particulière.

D'une façon générale, l'objectif de vérification est de fournir une évidence objective à l'étendue de laquelle une méthode MTF montre certaines propriétés de performance. Le niveau d'acceptabilité peut être dépendant dans un contexte et selon l'objectif de

vérification du sponsor (exemple : le niveau peut être acceptable pour un objectif et inacceptable pour un autre).

La vérification d'une méthode MTF, pour déterminer la justesse des états de la méthode MTF et/ou pour appliquer les tests requis par les sponsors de vérification, contient les étapes suivantes :

- constituer l'équipe de vérification (basée sur les compétences);
- assembler les inputs de vérification (les inputs sont identifiés ou produits);
- conduire la vérification (vérification est conduite);
- compiler l'output de vérification (le rapport de vérification est produit).

ISO 14143-4 « modèle de référence » [65] spécifie un modèle de référence de descriptions fonctionnelles, auquel les résultats d'une MTF peuvent être comparés pour analyser la convertibilité entre différentes propositions.

ISO 14143-5 « domaines fonctionnels » [66] spécifie les caractéristiques des domaines fonctionnels pour caractériser le(s) domaine(s) d'application de chaque MTF proposée.

En résumé, les parties 3, 4 et 5 d'ISO 14143 sont des rapports techniques et constituent des outils pour documenter certains critères de qualité des méthodes de mesure MTF de logiciels.

6.4.2 Analyse de ISO/IEC 14143

Le Tableau XXXV présente la mise en correspondance des parties de la norme ISO 14143 selon notre proposition d'un cadre de référence de vérification. Chaque ligne de ce tableau présente les différentes parties de 14143, les « X » nous indiquent la correspondance de chaque partie selon les étapes du cadre de référence dans la colonne respective. Ainsi, nous remarquons qu'il n'y a aucun concept identifié dans l'étape 4.

Par contre, les concepts reliés à la définition des concepts d'une MTF sont bien décrits. La norme ISO 14143-1 présente un consensus sur la définition des concepts fondamentaux d'une méthode MTF, et comment décrire les principes généraux pour appliquer cette méthode. Elle traite d'un seul type de mesure, et elle le traite de façon beaucoup plus détaillée et en profondeur que dans la série ISO 9126 sur la mesure de la qualité du logiciel. L'analyse de la norme ISO 14143-1 a été réalisée par Abran *et al.* [6] en utilisant le modèle de processus de mesure. Selon cette analyse, ISO 14143-1 définit clairement le concept de mesure de la taille fonctionnelle selon la perspective d'une méthode de mesure.

La série des documents ISO 14143-2 à 5 fournit la base pour évaluer si une méthode de mesure particulière est conforme à la définition ISO d'une MTF ainsi que les critères de « qualité » des méthodes de mesure. Cette série rencontre une partie des caractéristiques des résultats de mesure selon le VIM : elle rencontre un sous-ensemble du Tableau XI (exactitude, répétabilité, reproductibilité), quelques éléments des caractéristiques des instruments de mesure (seuil de mobilité), mais elle ne dresse pas les concepts liés aux instruments de mesure et aux étalons. Il y a encore du travail à effectuer pour que cette série ISO TR 14143-2 à 5 devienne une norme.

Tableau XXXV

Positionnement de ISO 14143-1 à 5 selon le cadre de référence

Cadre de référence	Étape 1 Design	Étape 2 Application	Étape 3 Résultat	Étape 4 Utilisation
ISO 14143-1 à 5				
Partie 1 : Définition des concepts	X			
Partie 2 : Évaluation de la conformité de MTF par rapport à 14143-1	X			
Partie 3 : Vérification des MTF		X	X	
Partie 4 : Modèle de référence		X	X	
Partie 5 : détermination des domaines fonctionnels pour l'utilisation avec MTF	X		X	

La revue de la littérature montre qu'il n'y a pas d'autres mesures ou d'autres types de mesure, qui ont été aussi bien décrites et documentées que dans 14143 et en utilisant autant de concepts pertinents à la métrologie. ISO 14143 pourrait donc être utilisée comme exemple pour définir et documenter d'autres types de mesure du logiciel.

6.5 ISO/IEC 19761 : COSMIC-FFP

Dans le but d'aider à identifier les forces des méthodes spécifiques proposées pour mesurer la taille fonctionnelle (MTF) du logiciel, nous proposons une approche analytique basée sur notre modèle initial [12] de l'ensemble des concepts de métrologie documenté dans le VIM [57].

Nous allons maintenant illustrer de façon pratique cette approche par une étude de cas en utilisant une méthode MTF spécifique du logiciel, soit la norme ISO/IEC 19761 : COSMIC-FFP. Cette sélection de COSMIC-FFP comme étude de cas est basée sur les deux critères suivants :

- 1) Lorsque comparées aux autres types de mesures de logiciels, les méthodes MTF de logiciels sont supportées par des descriptions opérationnelles beaucoup plus détaillées.
- 2) D'autre part, seul ce type de mesures de logiciels a franchi avec succès le processus de normalisation internationale (telle que ISO 14143) et a déjà atteint le statut officiel des normes internationales ISO (il est à noter que les définitions des mesures dans la série ISO 9126 sont contenues dans des rapports techniques et n'ont pas encore atteint le statut de normes internationales).

Nous utilisons la MTF la plus récemment développée, c'est-à-dire la norme ISO 19761 [69], comme cas de figure pour explorer si une telle mesure du logiciel couvre en elle-même tous les concepts fondamentaux de la métrologie, si ce n'est la majorité. Il s'agit également de la norme développée par les chercheurs du GÉLOG et pour laquelle nous

avons le plus d'expertise tant au point de vue théorique que pratique, tant dans son application que dans l'utilisation dans un contexte de formation en génie logiciel.

6.5.1 Survol de COSMIC-FFP : ISO 19761

La taille d'un logiciel peut être évaluée, par exemple, soit par des mesures de longueur (i.e. les lignes de code source dans un module, le nombre de pages dans un document de spécification des exigences) ou de fonctionnalités (i.e. le nombre de points de fonction dans une spécification). Les mesures de la taille fonctionnelle peuvent être directement dérivées des spécifications et peuvent être obtenues dès le début du cycle de vie de développement du logiciel, ce qui les rend utiles pour les objectifs de planification et durant tout le cycle de vie du projet. Les exigences pour une méthode MTF ont été codifiées dans ISO 14143-1 [62].

La première génération des mesures de la taille fonctionnelle a été développée vers la fin des années 1970. La seconde génération a émergé au début des années 2000, et a été rapidement adoptée comme une norme internationale [69]: *ISO/IEC 19761: 2003 COSMIC-FFP : A Functional Size Measurement Method*. Cette méthode COSMIC-FFP est basée sur l'application d'un ensemble de modèles, de règles et de procédures pour un logiciel donné, tel qu'il est perçu par ses utilisateurs, c'est-à-dire à travers les fonctionnalités utilisateurs requises (*Functional User Requirements –FUR*). Selon les règles obligatoires de ISO 14143-1 [62] sur le design d'une méthode MTF, les résultats obtenus d'une MTF doivent être indépendants de la technologie. Cette norme COSMIC-FFP est applicable pour mesurer différents types de logiciels (logiciels de gestion, temps réel, applications Internet et Web, etc.) indépendamment des technologies et des approches de développement et des décisions d'implantation dans les artefacts opérationnels du logiciel à mesurer.

COSMIC-FFP prend en considération que les FUR peuvent être décomposés en un ensemble de processus fonctionnels, et chacun de ces processus est un ensemble unique

de sous-processus accomplissant soit un mouvement de données soit une manipulation de données (Figure 31). Le modèle du logiciel COSMIC-FFP distingue quatre types de mouvement de données (i.e. entrée, sortie, lecture, écriture) tel qu'illustrés dans le modèle de contexte (Figure 32).

Par convention, un mouvement de données COSMIC est considéré comme un mouvement d'un seul groupe de données (quelque soit son nombre d'attributs). Les entrées transfèrent un groupe de données à partir de l'utilisateur vers l'intérieur de la frontière; les sorties transfèrent un groupe de données de l'intérieur de la frontière vers l'utilisateur; les lectures et les écritures transfèrent un groupe de données à partir et vers les unités de stockage.

Dans COSMIC-FFP, à chaque mouvement de données est assignée une seule « unité de mesure » qui, par convention, est égale à 1 Cfsu (*Cosmic functional size unit*). Ainsi, la taille totale du logiciel à mesurer correspond à l'addition de tous les mouvements de données reconnus par la méthode COSMIC-FFP. Voir [4;69] pour plus de détails au niveau des règles de mesure.

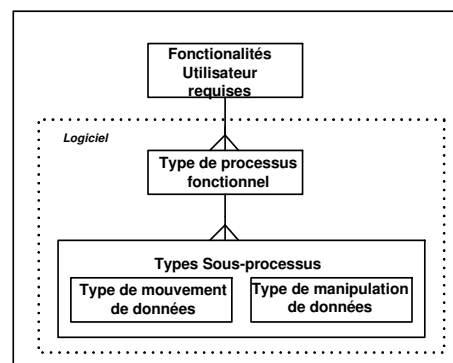


Figure 31 Un modèle général pour mesurer la taille fonctionnelle du logiciel [4;69]

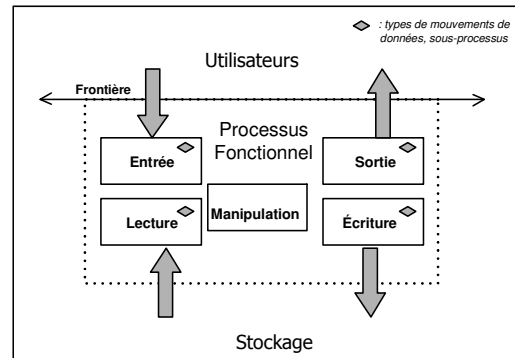


Figure 32 Les types de mouvements de données et certaines de leurs relations [4;69]

6.5.2 Analyse du design de COSMIC-FFP

6.5.2.1 Grandeurs et unités

Notre première analyse porte sur le design de COSMIC-FFP en utilisant l'ensemble des concepts de la métrologie sur les « grandeurs et unités », comme décrit dans le Tableau X du chapitre 4.

Le design de COSMIC-FFP nous permet de quantifier des grandeurs mesurables, équivalentes à un mouvement d'un seul groupe de données dans un processus fonctionnel, et ce dans des unités bien définies (Cfsu) (Tableau XXXVI). Cependant, en mesure de taille fonctionnelle, il n'y a pas encore officiellement de mesure dérivée et, par conséquent, le système de grandeurs ne comprend que la seule grandeur de base définie, soit le Cfsu. Dans COSMIC-FFP, le symbole de cette grandeur de base est la représentation visuelle de Cfsu, utilisée pour représenter l'unité de mesure = 1 Cfsu.

Il est à noter que dans d'autres MTF, comme les points de fonction [55], il n'y a pas d'unité de mesure, mais plutôt des fonctions d'échelle dont les poids varient de 3 à 15 [55].

Dans l'état de l'art actuel des mesures de la taille fonctionnelle, il n'y a qu'une seule unité de mesure; par conséquent il n'y a ni unité dérivée, ni unité hors système. Pour le moment, le système d'unités est donc des plus simples puisqu'il n'inclut aucun autre type d'unité. Le Tableau XXXVI nous montre que les sous-multiples d'unités (exemple, centigramme par rapport à kilogramme) ne sont pas encore formellement définis. De même, dans cette méthode il n'y a qu'un seul niveau de granularité et une échelle de repérage formellement reconnue, soit le niveau du mouvement d'un seul groupe de données, indépendamment de combien d'attributs contient ce groupe de données (bien que le guide COSMIC [4] reconnaisse que certains mesureurs peuvent définir leurs propres niveaux plus fins de granularité, non standardisés bien entendu).

Dans la dernière section du Tableau XXXVI, l'analyse de l'ensemble des concepts relatifs à la « valeur d'une grandeur » est un peu plus complexe car, à notre avis, elle contient à la fois des concepts reliés au design d'une méthode de mesure, et d'autres reliés à son application dans des cas spécifiques. Par ailleurs, dans COSMIC-FFP, le concept d'échelle de repérage qui constitue un ensemble de repères, défini par convention comme référence, correspond à l'échelle d'un mouvement d'un groupe de données (entrée, sortie, lecture et écriture, qui sont notés par convention en utilisant respectivement « E » pour entrée, « S » pour sortie, « L » pour lecture et « C » pour écriture). Chaque mouvement d'un groupe de données se voit attribuer, par convention dans COSMIC-FFP, une taille de 1 Cfsu. Il y a une définition normalisée de ce qui est reconnu comme « un groupe de données ». Ceci représente un ensemble discret de repères composé de : $E=S=L=C=1$.

Dans COSMIC-FFP, la « valeur numérique » du logiciel à mesurer correspond à l'addition, dans une même couche du logiciel, des valeurs assignées à chacun des mouvements identifiés de groupes de données. Cette addition donne la valeur numérique du logiciel à mesurer. En résumé, la « valeur numérique » et « l'échelle de repérage » sont explicitement définies dans la norme COSMIC-FFP.

Tableau XXXVI

Concepts « grandeurs et unités » de métrologie dans COSMIC-FFP

Métrologie [57]	Guide implantation COSMIC [4]	Clause ISO 19761 [69]
Système de grandeurs Grandeur de base Grandeur dérivée	(Ne comprend pour le moment qu'une seule grandeur de base) Taille fonctionnelle (Non encore définie)	2.5 Phase de mesure COSMIC-FFP
Dimension d'une grandeur Grandeur de dimension un/ Grandeur sans dimension	(Non-explicite) (Non-déterminé)	2.7 Contexte de mesure de la taille fonctionnelle
Unité (de mesure) Symbole d'une unité Système d'unités Unité (dérivée) cohérente Système cohérent d'unités Système international d'unités, (SI) Unité de base Unité dérivée Unité hors système Multiple d'une unité Sous-multiple d'une unité	= 1 Cfsu = Cfsu (Non-applicable) (Non-applicable) (Non-applicable) (Non-applicable) = 1 Cfsu (Aucune) (Aucune) (Non encore normalisé) (Non encore défini)	2.5 Phase de mesure COSMIC-FFP
Valeur (d'une grandeur) Valeur vraie Valeur conventionnellement vraie Valeur numérique Échelle de repérage	Valeur de la taille fonctionnelle (Non –déterminé) = Obtenue par jugement d'experts = Résultat de mesurage : taille fonctionnelle = Échelle = un mouvement d'un groupe de données sans classement (indépendamment du type de mouvement). Chaque mouvement d'un groupe de données se voit attribuer la valeur de 1 Cfsu.	2.2 Modèle de processus de mesure COSMIC-FFP

6.5.2.2 Étalon

Dans le domaine de la métrologie en sciences et en ingénierie, il est pris pour acquis qu'il doit y avoir des étalons pour l'étalonnage des instruments de mesure, et pour s'assurer de la cohérence des résultats de mesure au-delà des individus, des organisations et des nations. Mais il s'agit là d'un concept qui n'a pas encore été discuté dans le domaine des mesures en génie logiciel. Ce qui peut se rapprocher le plus de ce concept du Tableau XV, et de ses sous-concepts, c'est la préparation d'étude de cas avec une méthode de mesure de logiciels. Cependant, pour la première fois en génie logiciel des travaux de recherche sont en cours au GÉLOG sur la réalisation d'un étalon pour la mesure du logiciel et cette problématique a été présentée dans [80].

6.5.3 Analyse de mesurage avec COSMIC-FFP

D'après le VIM [57], le terme « mesurage » se réfère à la catégorie des termes pour l'ensemble des opérations requises afin d'obtenir un résultat de mesure (voir Figure 9), et ceci est « instancié » à travers le processus général décrit dans la Figure 8. La Figure 9 nous montre ensuite, en utilisant la représentation graphique d'un processus, les différents concepts reliés à celui de « mesurage ». Il est à noter qu'en métrologie, la grandeur à mesurer par le moyen d'un ensemble d'opérations (et l'instrument de mesure) est aussi appelée « mesurande », c'est-à-dire la grandeur en entrée qui est appliquée à l'instrument de mesure.

En se basant sur les détails de la Figure 9, un mode opératoire prend en input un mesurande, correspondant dans la norme COSMIC-FFP aux fonctionnalités requises pour l'utilisateur (FUR), pour en produire le résultat représentant la valeur numérique de la taille fonctionnelle. Le mode opératoire pour un mesurage spécifique doit tenir compte de l'opérateur du processus (ici, le mesureur), de la méthode de mesure elle-même (ici la norme) et des grandeurs d'influence (ici, les conditions qui pourraient influencer/biaiser les résultats de mesure). La méthode de mesure COSMIC-FFP est

explicite et la grandeur d'influence, dans le cas d'un mesurage, comprend par exemple l'expertise de l'utilisateur, la qualité de la documentation disponible pour le mesurage, le temps alloué, etc.

Dans COSMIC-FFP, la mesure d'une grandeur (mesurande) telle qu'elle est perçue par l'utilisateur, c'est-à-dire à travers les besoins fonctionnels (FUR), doit être transformée à travers un ensemble prescrit d'opérations logiques pour fournir une valeur numérique (un nombre représentant la taille fonctionnelle d'un logiciel). Ce nombre doit être associé à une unité de taille (Cfsu) pour représenter le résultat de la mesure.

Dans la norme ISO 19761 [69], les « définitions » normalisées des concepts relatifs à cette méthode sont spécifiés dans la clause 3, tandis que les séquences logiques des opérations sont décrites comme étant les « activités de mesure » en clause 6. Selon notre point de vue, ces « définitions » et ces « activités de mesure » devraient rencontrer les concepts de métrologie pour les « principes de mesure » et la « méthode de mesure » définis dans le VIM [57]. Cependant, ces deux concepts du VIM, qui correspondent aux fondements d'une mesure selon la perspective de métrologie (Figure 8) ne sont pas décrits dans un niveau plus fin de détails pour vérifier s'il y a ou s'il n'y a pas de correspondance totale des concepts dans cette norme [69].

Il est à noter que, dans la littérature de mesure du logiciel, les concepts « signal de mesure » et « valeur transformée » ne sont pas explicitement discutés. Bien que, ces concepts ne soient pas traités dans la norme 19761 [69], ils sont cependant explicitement présentés et discutés dans le « Guide COSMIC d'implantation de la norme » [4] et ils correspondent à l'ensemble des concepts qui sont catégorisés dans la « phase de mise en correspondance » entre la documentation des FUR et leur arrimage au modèle COSMIC-FFP de ces FUR. De façon plus explicite, le Guide COSMIC [4] précise que la « valeur transformée » est obtenue par la séquence suivante d'opérations : « Le mesureur doit identifier les limites du logiciel à mesurer, identifier tous les processus fonctionnels, les déclencheurs d'événements et les groupes de données, mettre en correspondance dans un

modèle de contexte logiciel en se basant sur les règles COSMIC, identifier les couches, identifier dans chaque processus fonctionnel les sous-processus, déterminer la mesure de la taille COSMIC, additionner les résultats de mesure » [4]. Un sommaire de ces correspondances des concepts de mesurage est présenté au Tableau XXXVII.

Tableau XXXVII

Concepts de métrologie « mesurage » dans COSMIC-FFP

Métrologie [57]	COSMIC-FFP [69]
Mesurande	FUR dans les artefacts du logiciel à mesurer
Signal de mesure	Phase de mise en correspondance : Modèle du contexte de mesure et modèle du logiciel de COSMIC-FFP
Valeur transformée	Non encore examinée
Mode opératoire	Phase de mesure : règles et méthodes à appliquer à l'output de la phase de mise en correspondance comme représentés dans le modèle général de mesure de la taille fonctionnelle COSMIC-FFP du logiciel
Résultat d'un mesurage	Taille fonctionnelle du modèle général du logiciel au FUR, valeur numérique
Opérateur	Mesureur
Méthode de mesure	ISO 19761: COSMIC-FFP
Grandeur d'influence	Par exemple : expertise du mesureur, qualité de la documentation des FUR, temps alloué pour mesurer, etc.

6.5.4 Analyse de COSMIC-FFP : instruments de mesure et résultats de mesure

Afin d'explorer les concepts de métrologie reliés aux instruments et aux résultats de mesure, nous allons utiliser un cas particulier, soit celui du prototype d'automatisation de COSMIC-FFP dans l'environnement RUP-Rational Rose décrit en [16] et développé au GÉLOG.

6.5.4.1 Instruments de mesure

Une mesure est normalement réalisée avec des instruments de mesure, lesquels instruments de mesure dans les milieux scientifiques et d'ingénierie (et aussi commerciaux) sont étalonnés en utilisant des étalons de référence. La Figure 10 illustre

par exemple la « chaîne de mesure » qui représente la suite d'éléments d'un système de mesure ou d'un instrument de mesure. Dans [16], l'équivalent d'une chaîne de mesure y est décrite et comprend comme stimulus à l'entrée la description des FUR de l'étude de cas utilisée, le prototype de mesure lui-même, ainsi que les résultats de mesure comme signal de sortie. Le système de mesure est l'équivalent de l'ensemble complet des éléments du prototype et les procédures manuelles pour exécuter un mesurage spécifié. Dans ce système, le détecteur (qui nous indique la présence d'un phénomène) est la fonction du prototype qui extrait les éléments à mesurer, le transducteur de mesure (qui fait correspondre à une grandeur d'entrée une grandeur de sortie) correspond à la solution de mise en correspondance entre les éléments de COSMIC et les concepts RUP-UML dans [16], l'instrument de mesure (qui nous permet de réaliser des mesurages) correspond à l'ensemble des fonctionnalités requises pour implanter les règles de mesure COSMIC-FFP, et les mesures matérialisées (qui permettent de reproduire ou encore de fournir une ou plusieurs valeurs) correspondent aux résultats de mesure, présentés en sortie de l'écran et conservé en mémoire. L'appareil intégrateur, qui par définition détermine la valeur du mesurande en intégrant une grandeur en fonction d'une autre grandeur, n'est pas déterminé dans le prototype COSMIC/RUP, car il ne traite d'aucune autre grandeur.

Pour le détail d'un instrument de mesure, le dispositif d'affichage correspond à l'écran d'affichage des résultats de mesure, et l'appareil enregistreur est l'équivalent de la fonction prototype qui permet l'enregistrement dans la base de données. La notion « d'échelle d'un appareil de mesure » est aussi incluse dans l'ensemble des concepts reliés aux instruments de mesure : elle comporte de nombreux sous-concepts comme illustrés dans le Tableau XII. Cependant, l'application de ces concepts de métrologie dépend considérablement de la présence des multiples et des sous-multiples d'une unité de mesure. Dans l'état de l'art actuel de mesure de la taille fonctionnelle du logiciel, ces sous-concepts ne sont pas encore présents. Les divers détails de correspondance dans cette étude de cas sont présentés dans le Tableau XXXVIII.

Tableau XXXVIII

Concepts des « instruments de mesure » et COSMIC/ RUP

Métrologie [57]	Prototype COSMIC/ RUP [16]
Chaîne de mesure	FUR + Prototype COSMIC/RUP + Résultats de la taille fonctionnelle [16]
Système de mesure : Décteur	Ensemble des éléments du prototype logiciel + procédures manuelles Fonction du prototype qui extrait les éléments à mesurer
Transducteur de mesure	Solution de mise en correspondance (entre COSMIC et les concepts RUP-UML dans [16])
Instrument de mesure	Dans le prototype COSMIC/ RUP, l'ensemble des fonctionnalités pour implanter les règles de mesure COSMIC-FFP
Mesure matérialisée	Résultats de mesure, présentés en sortie de l'écran et conservé en mémoire
Appareil intégrateur :	Ne figure pas dans le prototype (puisqu'il ne traite aucune « autre grandeur »)
Mesurande	Ensemble des FURs
Autre grandeur	Non indiqué dans ce prototype
Détail d'un instrument de mesure* :	
Dispositif d'affichage/ indicateur (+ index)	Écrans d'affichage des résultats de mesure
Appareil enregistreur/ Dispositif enregistreur	Fonction prototype qui permet l'enregistrement dans la base de données
*Ce tableau n'inclut pas tous les détails de la Figure 12	

6.5.4.2 Résultats de mesure

Dans tous les cas d'instanciation de mesures, un résultat de mesure est toujours associé à une incertitude de mesure car il n'existe pas en pratique de processus parfait de mesurage. L'incertitude de mesure est définie en métrologie comme un paramètre caractérisant la dispersion des valeurs qui pourraient raisonnablement être attribuées au mesurande [57], c'est-à-dire un intervalle centré sur la valeur mesurée et dans lequel il est fort probable de retrouver la valeur vraie (ou la valeur conventionnellement vraie).

Dans l'état actuel des connaissances en mesure de la taille fonctionnelle des logiciels, la valeur vraie est généralement obtenue par jugements d'experts en mesure. La différence des résultats obtenus par le mesureur (ou l'instrument de mesure) et par l'expert représente l'erreur (Tableau XI). L'équivalent dans [16] de la catégorie « résultats de

mesure » est présenté dans le Tableau XXXIX. Dans cette catégorie, l'indication qui représente la valeur indiquée par l'instrument de mesure correspond aux résultats détaillés et totalisés selon les gabarits proposés dans le Guide COSMIC d'implantation de la norme [4]. De même, le résultat brut (résultat avant correction de l'erreur systématique) correspond aux résultats obtenus avant l'intervention humaine pour ajouter l'information manquante. Par contre, le résultat corrigé (résultat après correction de l'erreur systématique) correspond aux résultats de mesure révisés après l'ajout de l'information manquante.

Concernant le mode de vérification des résultats, le concept d'exactitude du résultat du mesurage (étroitesse de l'accord entre le résultat et une valeur vraie du mesurande) n'est pas testé avec un échantillon assez large. De même que pour le concept répétabilité (étroitesse de l'accord entre le résultat des mesurages successifs du même mesurande, mesurages effectués dans les mêmes conditions de répétabilité) et reproductibilité (étroitesse de l'accord entre le résultat des mesurages du même mesurande, mesurages effectués en faisant varier les conditions de mesure), le prototype d'outil logiciel donne normalement les mêmes résultats dans les mêmes conditions de répétabilité, mais, il doit être vérifié par d'autres expérimentations à plus grande échelle.

D'autre part, les concepts reliés à l'incertitude de mesure ne sont pas encore explorés. En d'autres termes, dans [16], les résultats d'une étude de cas sont présentés. Toutefois, il s'agit d'une petite étude de cas pour fin de démonstration de la faisabilité technique des concepts d'automatisation de la norme COSMIC-FFP [69] dans l'environnement choisi RUP/Rational Rose. Pour obtenir des résultats statistiquement significatifs sur les concepts de la topologie détaillée des résultats de mesure pour ce prototype d'instrument de mesure, un ensemble plus large d'études de cas est requis. Un sommaire de la mise en correspondance de cette étude de cas est présenté dans le Tableau XXXIX.

Tableau XXXIX

Concepts des « résultats de mesure » et prototype COSMIC/RUP

Métrologie [57]	Prototype COSMIC/ RUP [16]
Types des résultats de mesure	Résultats détaillés, et sommarisés selon les gabarits proposés dans le Guide COSMIC d'implantation de la norme [4]
Indication	
Résultat brut	Résultats de mesure, avant l'intervention humaine pour ajouter l'information manquante
Résultat corrigé	Résultats de mesure révisés, après l'ajout de l'information manquante
Mode de vérification des résultats	
Exactitude de mesure	Dans [16], cette caractéristique n'a été testée qu'à une petite échelle avec une seule étude de cas dont la valeur vraie avait été déjà déterminée par des experts. Il n'y a pas assez de cas pour avoir une connaissance quantitative statistiquement significative de cette caractéristique
Répétabilité	Un outil logiciel donne normalement les mêmes résultats dans les mêmes conditions de répétabilité (Nécessite d'être vérifié par d'autres expérimentations à plus grande échelle)
Reproductibilité	Idem à ci-dessus
Incertitude de mesure et huit autres concepts reliés	Caractéristiques non encore explorées

6.5.5 Caractéristiques des instruments de mesure

La catégorie des concepts relatifs aux « caractéristiques des instruments de mesure » a été modélisée soit en caractéristiques d'utilisation, soit en caractéristiques de contrôle comme décrites dans le chapitre 4 au Tableau XIV.

6.5.5.1 Caractéristiques d'utilisation

Dans le prototype COSMIC/RUP [16], la présence de certaines caractéristiques d'utilisation d'un instrument de mesure peuvent être constatées. Par exemple, en termes de caractéristiques d'opération et pour que le prototype fonctionne, il faut modéliser les FUR selon le processus RUP basé sur les formalismes UML.

Pour les conditions limites, le prototype ne traite actuellement qu'une seule couche de logiciel à la fois dans des conditions assignées de fonctionnement. Pour évaluer le fonctionnement du prototype ou pour l'inter-comparaison des résultats de mesure obtenus, les conditions de référence de [16] nous indiquent qu'un processus fonctionnel doit avoir plus que deux mouvements de données.

Au niveau des concepts d'affichage (calibre, intervalle de mesure, valeur nominale et l'étendue de la mesure) ils ne sont pas explorés faute des conditions expérimentales pertinentes ou probablement ces caractéristiques sont non encore explorées par le constructeur du prototype, ni dans son mémoire de maîtrise. Ainsi, un futur projet de recherche, avec beaucoup plus de cas, pourrait explorer tous ces « critères » de façon plus adéquate.

Au niveau des caractéristiques dynamiques et pour les caractéristiques de transfert, il y a de nouveaux niveaux d'unités de mesure qui ont été définis dans le prototype (Unités Ufsu au niveau du diagramme de cas d'utilisation, et unités Sfsu au niveau du diagramme de séquence), mais ces caractéristiques n'ont pas encore été analysées.

Pour la constance (aptitude d'un instrument de mesure à conserver ses caractéristiques métrologiques constantes au cours du temps), le prototype COSMIC/RUP [16] devrait normalement préserver ses caractéristiques métrologiques au cours du temps (même, par exemple lors d'un changement de version dans chacun de ses composants logiciels). Pour le concept discrétion (aptitude d'un instrument de mesure à ne pas modifier le mesurande), le prototype [16] est un instrument discret pour la mesure de processus fonctionnel. Pour le concept temps de réponse, il existe un intervalle de temps entre le moment du signal d'entrée et le moment du signal de sortie. Par contre, dérive (variation lente d'une caractéristique métrologique de l'instrument) et zone morte (intervalle maximal auquel on peut faire varier le signal d'entrée sans provoquer de variation de la réponse de l'instrument) ne sont pas encore examinés dans [16]. Un résumé de ces caractéristiques est présenté dans le Tableau XL.

Tableau XL

Concepts de caractéristiques d'utilisation et prototype COSMIC/RUP

Métrologie [57]	Prototype COSMIC/ RUP [16]
Caractéristiques d'opération :	
Conditions assignées de fonctionnement	Il faut modéliser les FURs selon le processus RUP basé sur les formalismes UML
Conditions limites	Le prototype ne traite actuellement qu'une seule couche de logiciel à la fois
Conditions de référence	Par exemple : un processus fonctionnel doit avoir plus que deux mouvements de données
Caractéristiques d'affichage :	
Calibre	Non encore exploré par le prototype
Intervalle de mesure	Non encore exploré par le prototype
Valeur nominale	Non applicable
Étendue de mesure	Non encore explorée par le prototype
Constante	Non encore examinée par le prototype
Résolution	Non encore explorée par le prototype
Caractéristiques dynamiques :	
Caractéristique de transfert	De nouveaux niveaux d'unités de mesure ont été définis dans le prototype (Ufsu et Sfsu) mais les caractéristiques de transfert n'ont pas encore été analysées.
Constance	Le prototype devrait préserver ses caractéristiques métrologiques au cours du temps (même, par exemple lors d'un changement de version dans chacun de ses composants logiciels).
Discrétion	Le prototype COSMIC/RUP est un instrument discret pour la mesure de processus fonctionnel.
Dérive	Non encore examiné
Temps de réponse	Il existe un intervalle de temps entre le moment du signal d'entrée et le moment du signal de sortie.
Zone morte	Non encore examiné

6.5.5.2 Caractéristiques de contrôle

Dans le prototype COSMIC/RUP [16], les résultats de l'analyse selon les caractéristiques de contrôle d'un instrument de mesure du Tableau XIV sont présentés dans le Tableau XLI. Pour les caractéristiques qualitatives, un cas particulier a été identifié pour le concept de la sensibilité (i.e. ce qui représente le quotient de l'accroissement de la réponse d'un instrument de mesure par l'accroissement correspondant du signal d'entrée) : par exemple pour indiquer si oui ou non il est

possible de bien catégoriser les mouvements de lecture ou d'écriture. Cependant, il a été souligné dans le prototype COSMIC/RUP [16] que cette problématique n'avait pas d'impact (exemple de sensibilité= 0) sur la valeur numérique puisque chaque mouvement de groupe de données =1 indépendamment de sa catégorie (le résultat de mesure est donc insensible à une mauvaise classification des types de mouvements de groupe de données).

Le seuil de mobilité correspond à la taille minimale d'un changement à un $FUR = 1$ Cfsu. Le concept exactitude de l'instrument de mesure (aptitude de l'instrument à donner des réponses proches de la valeur vraie) n'est pas totalement couvert dans le prototype COSMIC/RUP [16]. En effet, ce concept n'a été analysé qu'avec une seule étude de cas, et de peu d'envergure. Plus d'études de cas devront être construites et les résultats analysés, pour déterminer par exemple l'exactitude des résultats mesurés par le prototype, et sous quelles conditions. Par ailleurs, la justesse (aptitude de l'instrument à donner des indications exemptes d'erreur systématique), n'a pas été encore examinée.

Finalement, le prototype [16] répond au concept fidélité (aptitude de l'instrument à donner des indications très voisines lors de l'application répétée du même mesurande dans les mêmes conditions de mesure) puisqu'il donne la même valeur pour les mêmes conditions de mesure. D'autre part, la mise en correspondance avec les caractéristiques quantitatives (classe d'exactitude, erreur (d'indication) d'un instrument de mesure, erreurs maximales tolérées/ limites d'erreur tolérées, erreur au point de contrôle, erreur à zéro, erreur intrinsèque, erreur de justesse, et erreur réduite conventionnelle) n'est pas explorée faute de conditions expérimentales pertinentes.

Tableau XLI

Concepts des caractéristiques de contrôle et prototype COSMIC/RUP

Métrologie [57]	Prototype COSMIC/ RUP [16]
Caractéristiques qualitatives :	
Sensibilité	Un cas particulier a été identifié, par exemple pour indiquer si oui ou non il est possible de bien catégoriser les mouvements de lecture ou d'écriture. Cependant, il a été souligné dans [16] que cette problématique n'avait pas d'impact (exemple de sensibilité = 0) sur la valeur numérique puisque chaque mouvement de groupe de données = 1 indépendamment de sa catégorie.
(seuil de) mobilité	1 Cfsu, la taille minimale d'un changement à un FUR
Exactitude d'un instrument de mesure	N'a été analysée qu'avec une seule étude de cas, et de peu d'envergure. Plus d'études de cas devront être construites et les résultats analysés, pour déterminer par exemple l'exactitude des résultats mesurés par le prototype, et sous quelles conditions
Justesse	Non encore examiné
Fidélité	Le prototype donne la même valeur pour les mêmes conditions de mesure
Caractéristiques quantitatives (de contrôle et d'estimation) :	
Classe d'exactitude	Non encore examinée
Erreur (d'indication) d'un instrument de mesure	Non encore examinée
Erreurs maximales tolérées/ limites d'erreur tolérées	Non encore examinées
Erreur au point de contrôle	Non encore examinée
Erreur à zéro	Non encore examinée
Erreur intrinsèque	Non encore examinée
Erreur de justesse	Non encore examinée
Erreur réduite conventionnelle	Non encore examinée

6.6 Sommaire et observations

Dans les autres disciplines du génie, le design des méthodes et des instruments de mesure est basé sur la compréhension des attributs à mesurer. Par exemple, mesurer le temps (en tant qu'une grandeur de base) se base sur l'établissement d'un modèle de description du temps ainsi que l'outil pour sa mesure. Cependant, les attributs du logiciel

ne sont pas encore compris aussi bien que les attributs des autres domaines de l'ingénierie. Les mesures doivent nous fournir l'information fiable et leurs significations doivent être correctement comprises et interprétées. La compréhension des fondements de chaque mesure est nécessaire.

Dans ISO 9126, les concepts de mesure du logiciel manquent encore de précision et de clarté, et il n'y a pas encore de consensus détaillé sur la modélisation de l'attribut de l'entité à mesurer. Il y a principalement des formules de comptage mais pas de méthode de mesure et il n'y a pas de référence à des étalons.

D'autre part, la norme ISO 14143 traite un seul type de mesure qui est la MTF, et de façon beaucoup plus détaillée.

L'analyse de la norme ISO COSMIC-FFP (ISO 19761), qui est une MTF spécifique a montré que :

- d'une part, le design de la méthode COSMIC-FFP couvre la majorité des concepts de métrologie décrits dans le VIM relatifs au design des méthodes de mesure;
- d'autre part, des études de cas à plus grande échelle devraient être requises pour l'étude des caractéristiques des instruments de mesure, caractéristiques telles qu'identifiées dans le VIM.

La mesure est reconnue comme un concept fondamental dans toutes les disciplines d'ingénierie. Elle nous permet de trouver l'information requise pour prendre les bonnes décisions et les actions appropriées dans un projet. Il existe de nombreuses mesures de logiciels proposées à l'industrie pour décrire quantitativement les différentes caractéristiques du logiciel.

Beaucoup de travail reste à donc effectuer pour étudier aussi bien le design des méthodes de mesure de logiciels que les caractéristiques des instruments de mesure pour l'instrumentation de la mesure en industrie. En effet, la plupart des concepts de

métrologie reliés aux instruments de mesure devraient être explorés adéquatement par les ingénieurs du logiciel.

Le cas pratique illustre également comment le modèle du VIM pourrait être utilisé non seulement pour analyser les autres méthodes MTF, mais également pour toutes les autres mesures de logiciels proposées à l'industrie. Nous suggérons donc d'utiliser l'ensemble des concepts de métrologie documentés dans le VIM comme des critères pour analyser les forces et les faiblesses des autres méthodes MTF ainsi que toute autre mesure du logiciel.

CHAPITRE 7

ANALYSE DE ISO 15939 ET DE COCOMOII

7.1 Introduction

Dans le domaine de la gestion du logiciel, il y a différents types de mesure du logiciel qui relient les normes telles que : ISO 15939 (processus de mesure), ISO 12207 (cycle de vie du système) et ISO 15504 (capacité de détermination de la maturité et norme d'évaluation du processus).

Dans le volet de l'utilisation des résultats de la mesure, l'estimation du logiciel est un des exemples souvent cités tant dans la littérature que par les praticiens. L'un des modèles d'estimation les plus cités dans la littérature est le modèle COCOMO II. Dans ce chapitre, nous analysons du point de vue métrologique la norme ISO 15939 et le modèle COCOMO II.

7.2 Processus de mesure de logiciels (ISO 15939)

La mesure du logiciel est un outil essentiel dans la gestion et l'amélioration continue des processus et produits logiciels. Les gestionnaires de projets logiciels s'intéressent de plus en plus à l'utilisation des mesures et des informations précises pour la prise de décision. Pour satisfaire aux besoins d'information des gestionnaires des projets, la communauté du génie logiciel a défini dans la norme ISO/IEC 15939 [67] un modèle d'un processus de mesure du logiciel destiné à être applicable à la discipline du génie logiciel. Il est à noter que cette norme a par la suite été adoptée comme une base pour le domaine de mesure et processus d'analyse du CMMI [32]. La norme ISO 15939 décrit un processus de mesure qui se doit d'être flexible, ajustable et adaptable aux besoins des divers utilisateurs.

Plusieurs gestionnaires reconnaissent la mesure du logiciel en tant qu'un outil permettant de les aider à comprendre et diriger leurs projets de développement de logiciels. Plusieurs de ces gestionnaires s'intéressent à l'utilisation du produit d'informations dans des modèles de qualité en tant qu'indicateurs pour la prise de décision. Ces indicateurs de qualité sont des éléments essentiels pour le succès du processus de gestion des projets logiciels, car celui-ci a une influence significative sur les opérations de l'entreprise. Étant donné l'importance de mesures appropriées pour la gestion et l'estimation des attributs de qualité, divers auteurs en génie logiciel ont proposé plusieurs exemples pour implanter des mesures du logiciel, comme l'approche *Goal-Questions-Metrics* (GQM) [17;18]). La norme ISO 15939 est considérée comme une nouvelle technologie pour planifier un processus de mesure et l'améliorer continuellement.

Quel est le niveau de maturité du processus de cette norme 15939 (en termes de concepts de métrologie)? Et, quels sont les critères de vérification pour aider les gestionnaires à choisir une bonne mesure? Pour répondre à ces questions, nous utilisons le cadre de référence de mesure (Tableau XXVIII) comme un outil d'analyse. D'autre part, une autre question qui se pose, à savoir : qu'entendons-nous par une « bonne » mesure?

Dans le langage de la théorie de mesure [41], le concept d'une bonne mesure est la conformité entre les représentations et les attributs représentés. Les représentations sont celles que nous formons dans le but de représenter le plus fidèlement possible ce qui existe dans le monde empirique, et le décrire dans le monde mathématique.

Dans les domaines scientifiques, un processus de mesure ou un mesurage consiste à obtenir une comparaison quantitative entre un étalon prédéfini et le mesurande. Le terme mesurande est utilisé pour désigner l'attribut à mesurer, c'est-à-dire l'input au processus de mesure. L'action de mesurage produit un résultat. L'étalon de comparaison doit être de même caractéristique que le mesurande : par exemple, le mètre définit clairement l'étalon de longueur. Dans ce contexte, une « bonne » mesure est définie par référence à un étalon. En génie logiciel, qu'est-ce qui est connu en mesure?

7.2.1 Survol de la norme ISO 15939

La norme ISO 15939 [67] définit un processus de mesure du logiciel applicable aux domaines de gestion et de l'ingénierie du logiciel. Ce processus est décrit selon un modèle qui définit les activités requises pour spécifier quelle est l'information de mesure requise, comment appliquer les résultats de mesure et de l'analyse, et comment déterminer si les résultats de mesure sont valides.

Le but de la norme 15939 est de permettre d'identifier, définir, sélectionner, appliquer et améliorer la mesure du logiciel au sein de la structure organisationnelle. Il est spécifié également que le processus de cette norme supporte la définition d'un ensemble approprié de mesures qui adresse les besoins d'informations. Ce processus vise à adresser les exigences en information de l'organisation ou du projet, et sert comme une méthode pour définir le processus logiciel et le produit d'informations de mesure.

Toutefois, ISO 15939 ne prescrit pas un modèle organisationnel pour la mesure. Selon son but, la norme 15939 est limitée à définir les éléments fondamentaux dans le contexte de mesure de logiciels et à décrire les activités et les tâches de mesure pour permettre une base objective (précise) de communication et aider ainsi les gestionnaires à la prise de décisions dans les projets logiciels. Dans cette section, nous allons analyser si la structure du document ISO 15939 rencontre l'ensemble des concepts du cadre de référence.

Il est à noter que dans ISO 15939, les types de mesure (mesures de base, mesures dérivées), les indicateurs et les entités mesurables sont fournis seulement en annexe du document ISO selon le modèle de l'information de mesure. Généralement, les annexes ne font pas partie du contenu prescriptif d'une norme ISO. La norme 15939 consiste principalement à collecter, à analyser et à documenter les données des produits développés et des processus implantés au sein de l'organisation pour supporter

efficacement la gestion des processus et pour démontrer objectivement la qualité des produits.

7.2.2 Modèle du processus de mesure du logiciel de la norme ISO 15939

Le processus de mesure du logiciel définit quatre activités de base. Ces activités sont considérées nécessaires pour obtenir un résultat significatif de mesure :

- établir et supporter un engagement de mesure;
- planifier le processus de mesure;
- exécuter le processus de mesure;
- évaluer l'utilisation de mesure.

Les activités (2) et (3) concernent le « mesureur » tandis que les activités (1) et (4) concernent la base pour le responsable du processus de mesure (Figure 33).

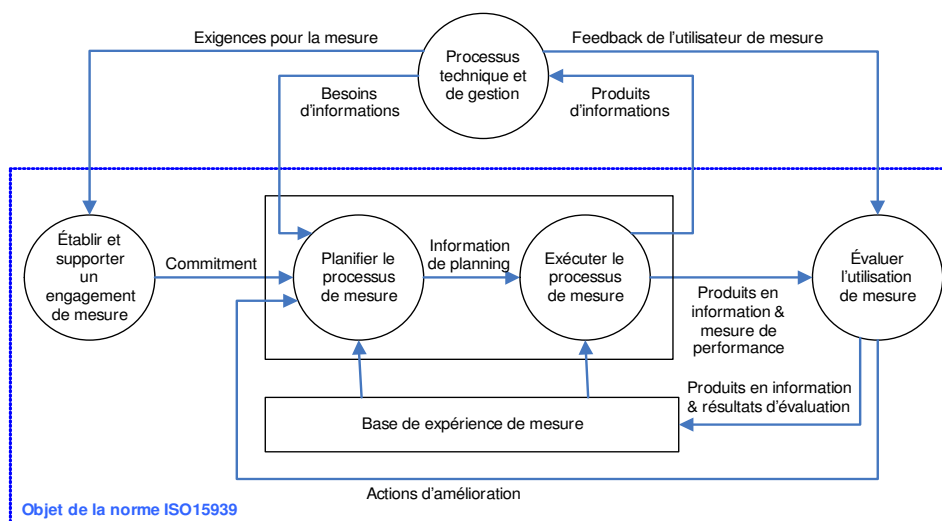


Figure 33 Modèle du processus de mesure du logiciel
(traduit de la norme ISO/IEC 15939 [67])

Il est spécifié que les activités de planification et d'exécution du processus de mesure sont guidées par les besoins d'information de l'organisation. Pour chaque besoin d'information, le processus produit l'information qui sera nécessaire à l'organisation pour la prise de décision. D'autre part, l'évaluation et le feedback sont des composants essentiels du processus de mesure, et qui devraient améliorer continuellement le processus de mesure. Les mesures devraient être évaluées en termes de valeur ajoutée fournie à l'organisation, et déployée seulement si les bénéfices peuvent être identifiés. De plus, il y a un référentiel qui est la « base de l'expérience de mesure » pour permettre de capturer l'information des itérations antérieures du cycle, des évaluations passées de l'information et des évaluations des itérations passées du processus de mesure. Ainsi, tout le processus dépend de la crédibilité des besoins d'informations.

7.2.3 Analyse de la norme ISO 15939

Le Tableau XLII présente le positionnement de l'utilisation des mesures pour un projet logiciel dans la norme ISO 15939 selon les étapes du cadre de référence de mesure. Ainsi, les lignes de ce tableau présentent les activités et les tâches telles que décrites dans le modèle du document ISO 15939, les « X » indiquent le positionnement de chaque ligne (activité) par rapport au cadre de référence dans la colonne respectée.

Tableau XLII

Positionnement de l'utilisation des mesures pour un projet logiciel dans ISO 15939

Cadre de référence Activités et tâches du modèle de la norme 15939	Étape 1 Design	Étape 2 Application	Étape 3 Résultats	Étape 4 Utilisation
Établir et supporter un engagement de mesure :				
- Accepter les exigences pour la mesure				
- Assigner les ressources				

Tableau XLII (Suite)

Cadre de référence Activités et tâches du modèle de la norme 15939	Étape 1 Design	Étape 2 Application	Étape 3 Résultats	Étape 4 Utilisation
Planifier le processus de mesure :				
- Caractériser l'unité organisationnelle		X		
- Identifier les besoins en information				X
- Sélectionner les mesures				X
- Définir la collection, l'analyse et les procédures de rapport des données				X
- Définir les critères pour l'évaluation des produits d'informations et du processus de mesure		X		
- Réviser, approuver et fournir les ressources pour les tâches de mesure		X		
- Acquérir et déployer le support des technologies		X		
Exécuter le processus de mesure :				
- Intégrer les procédures		X		
- Collecter les données		X		
- Analyser les données et développer les produits d'informations				X
- Communiquer les résultats				X
Évaluer l'utilisation de mesure :				
- Évaluer les produits d'informations et le processus de mesure				X
- Identifier les améliorations potentielles				X
Annexe		X		X

Le Tableau XLII montre que le modèle du processus de mesure de la norme ISO 15939 ne traite ni l'aspect du design d'une mesure, ni l'aspect résultat de mesure. En effet, la plupart des activités du processus de mesure (planification, exécution et évaluation de l'utilisation de mesure) traitent quelques éléments de l'application d'une mesure et son utilisation pour la prise de décision. Certaines activités de mesure se basent sur des questions pour identifier les objectifs désirés. À partir de ces questions, il y aura dérivation des attributs à mesurer.

Bien que cette norme utilise une terminologie cohérente pour les concepts fondamentaux de mesure qui est dérivée en partie du document VIM, il manque encore des concepts

pertinents en mesure tels que les éléments de spécification du design et les critères de qualité métrologiques des résultats de mesure.

Dans cette norme ISO 15939, la première activité « **Établir et supporter un engagement de mesure** » n'a pas été adressée dans aucune étape du cadre de référence. Ceci est due au fait que c'est une activité de gestion, et non pas de mesure. Elle se base sur l'acceptation des exigences pour l'utilisation de mesure et l'assignation des ressources pour les objectifs de planification et de contrôle.

La deuxième activité « **Planifier le processus de mesure** » est également une activité qui se compose de plusieurs tâches :

- caractérisation de l'unité organisationnelle relative à la sélection des mesures et l'interprétation des informations fournit un contexte d'utilisation de mesure. Ce contexte contient des contraintes en termes de processus organisationnel, domaines d'application, technologie, interfaces divisions/département et structure de l'organisation (modèle descriptif de gestion du processus);
- les besoins d'informations identifiés pour l'utilisation de mesure émanent des processus techniques et de gestion. Ils peuvent être arrangés et sélectionnés selon un ordre de priorité. Les mesures candidates qui ont un lien avec les besoins d'informations doivent être sélectionnées. Cette sélection se base sur le modèle de l'information de mesure (Figure 34).

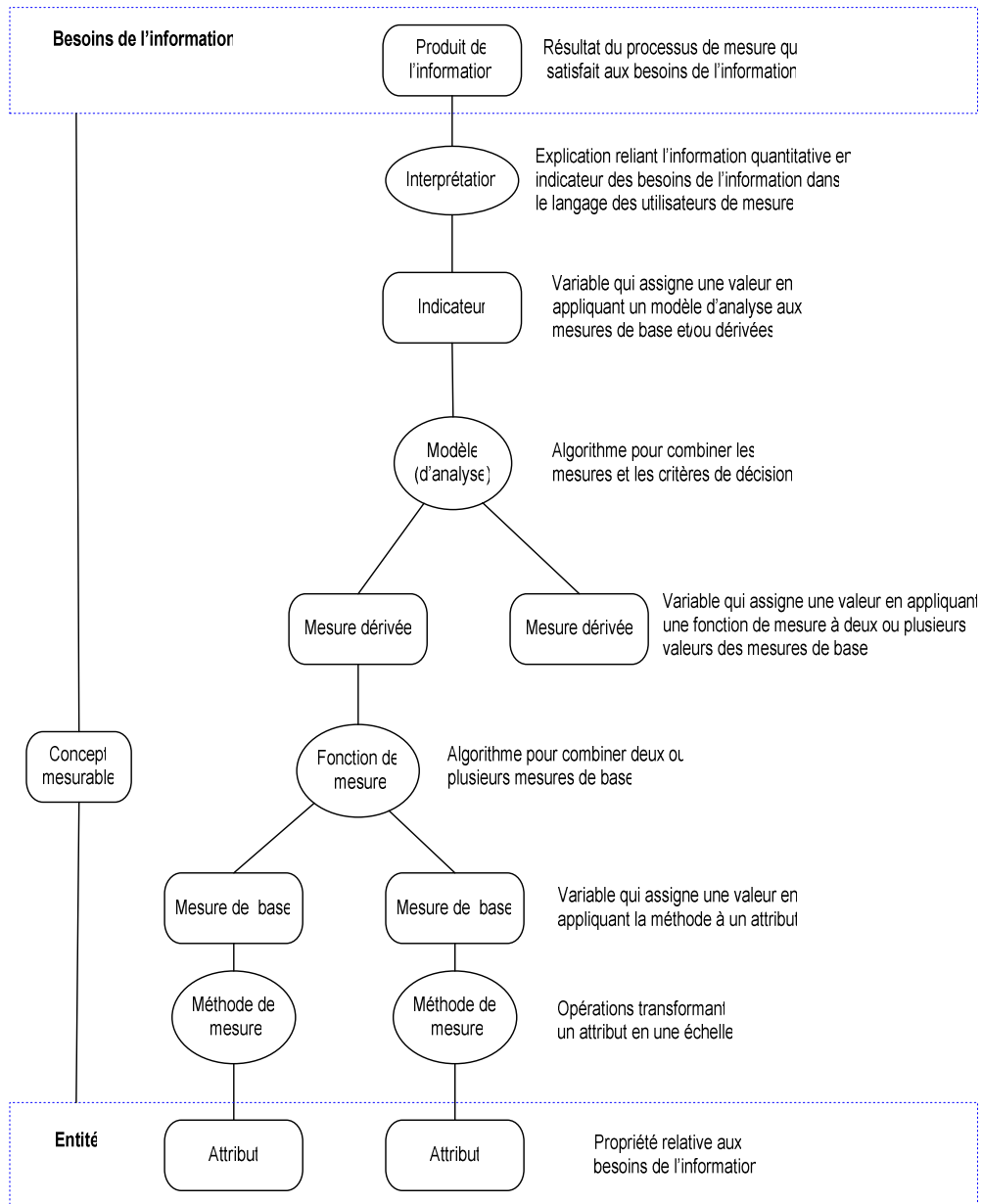


Figure 34 Modèle de l'information de mesure
(traduit de la norme ISO/IEC 15939 [67])

La Figure 34 présente le modèle présenté dans l'annexe du document ISO 15939; en utilisant certains concepts du VIM, le concept mesurable pour fournir un produit d'informations est bien décrit dans le modèle. Cependant, tous les composants du processus de mesure de 15939 [67] se basent sur ce produit d'informations. Il est nécessaire que ces informations soient crédibles pour les utiliser dans un modèle d'analyse. D'autre part, il y a un manque de référence à des étalons. Le mécanisme de prise de décision dans une entreprise doit se baser sur un système de mesure (tel est le cas pour les informations fournies en comptabilité) et qui sont contrôlées pour prendre les bonnes décisions au bon moment. Parmi les mesures candidates, il y a la sélection des mesures :

- la sélection des mesures est bien documentée en spécifiant leur nom, leur unité de mesure, leur définition formelle, la méthode de collection de données et leur lien avec les besoins d'information;
- la définition, la collection, l'analyse et les procédures de rapport de données;
- la définition des critères pour l'évaluation des produits d'informations et du processus de mesure. Les critères pour l'évaluation des produits d'informations sont définis pour s'assurer que les données ont été collectées et analysées rigoureusement pour satisfaire les besoins d'information. Les critères (tels que l'exactitude d'une procédure de mesure, la fiabilité d'une méthode de mesure) sont définis dans un contexte des objectifs techniques et d'affaire de l'unité organisationnelle. Les critères pour l'évaluation du processus de mesure doivent être définis (exemple efficacité du processus de mesure);
- réviser, approuver et fournir les ressources pour les tâches de mesure;
- acquérir et déployer le support des technologies afin de permettre le déploiement et l'opérationnalisation du programme de mesure.

La troisième activité « **Exécuter le processus de mesure** » est une activité de gestion, composée des tâches suivantes :

- intégrer les procédures : Intégration des données dans les processus appropriés. Les procédures de collection des données intégrées vont être communiquées aux fournisseurs de données;
- collecter les données : Les données sont collectées et stockées incluant tout contexte d'informations nécessaire pour vérifier, pour comprendre ou pour évaluer les données;
- analyser les données et développer les produits d'informations : Les données collectées sont analysées, les résultats de l'analyse des données sont interprétées, et les produits d'informations sont révisés;
- communiquer les résultats : Les produits d'informations sont documentés et communiqués aux utilisateurs de mesure.

La quatrième activité « **Évaluer l'utilisation de mesure** » est une activité de gestion, composée des tâches suivantes :

- évaluer les produits d'informations et le processus de mesure : les produits d'informations (mesures de base, dérivées, indicateurs) sont évalués par rapport aux critères d'évaluation spécifiés (objectifs ou subjectifs) et les conclusions sur les forces et faiblesses. Le processus de mesure est évalué par rapport aux critères d'évaluation spécifiés (performance, conformité, maturité) et les conclusions sur les forces et faiblesses; les leçons apprises à partir de l'évaluation sont stockées dans la « base de l'expérience de mesure » pour les rendre disponibles;
- identifier les améliorations potentielles : les améliorations potentielles des produits d'informations et du processus de mesure sont identifiées et communiquées.

Il est à noter que dans ISO 15939 le terme « mesure » est utilisé indifféremment pour désigner à la fois plusieurs concepts distincts tels que : (une méthode de mesure, application d'une méthode de mesure, résultats de l'application de la méthode de

mesure, et utilisation de mesure). Pour éviter toute confusion, le terme « mesure » devrait toujours être utilisé en utilisant une de ces expressions, plus précises et moins ambiguës. La notion d'un système de mesure ou étalon est importante pour toutes les disciplines matures de l'ingénierie. Cette notion ne figure pas encore dans le document de la norme ISO 15939.

Les procédés de normalisation évoluent, et il s'en crée de nouveaux à mesure que des technologies nouvelles sont établies. L'établissement d'étalons permettra de normaliser les indicateurs de qualité, et faire des estimations.

7.3 Interprétation pratique : analyse de COCOMO II

Estimer la taille et le coût est l'un des thèmes les plus importants dans le domaine de la gestion des projets logiciels. L'amélioration continue du marché logiciel a permis d'étendre de nouvelles méthodes pour l'estimation de la taille ou l'effort de développement du produit dans le domaine d'estimation de coût, comme les modèles COCOMO II (*Constructive Cost Model*) [22] et PUTNAM-SLIM (*Software Lifecycle Management*) [95].

Dans cette section, nous allons considérer uniquement le modèle COCOMO II, et ce à titre d'exemple. Ce modèle est la version améliorée de COCOMO'81 pour tenir compte des nouveaux modèles du processus de développement de logiciel et des capacités tels que les processus concurrents, le développement rapide des applications, les initiatives de maturité des logiciels, etc. L'objectif de COCOMO II est l'estimation des projets logiciels. Il est utilisé pour prédire certains attributs des entités telles que l'effort de développement du logiciel, la fiabilité du logiciel et la productivité des programmeurs. Il fournit des sous-modèles : modèle *early design* et modèle « post-architecture ». Ces sous-modèles sont supposés être cohérents avec la granularité de l'information disponible dans le cycle de vie de développement de logiciels afin de déterminer l'estimation des logiciels. Le modèle post-architecture est le plus utilisé dans

l'estimation de l'effort et la durée d'un projet de développement d'un produit logiciel après que l'architecture du cycle de vie du produit logiciel ait été établie. Il permet la prédiction de l'effort de développement en Homme-Mois (PM) d'un projet logiciel selon l'équation suivante :

$$PM = A \times size^E \prod_{i=1}^n EM_i \quad (2)$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j \quad (3)$$

Avec $A= 2.94$ et $B= 0.91$ (selon COCOMO II.2000)

Les intrants de ce modèle sont la taille du logiciel (*size*) mesurée en KISL (Kilo Instructions Sources Livrées), les 5 facteurs d'échelle, SF_j et les 17 multiplicateurs de l'effort EM_i les facteurs affectant le coût (*cost drivers*). Ces intrants sont présentés dans le Tableau XLIII, inspiré de [50].

Dans son modèle, Boehm présentait des valeurs attribuées à A et B de l'équation (2) et (3). Mais, il ne nous dit pas comment il en est arrivé à ces valeurs. Cela ne facilite pas la recherche sur ces questions.

De plus, la détermination de l'effort de développement en PM est basée en particulier sur l'identification des facteurs. Ces facteurs sont mesurés sur la base des données d'autres projets similaires ou sur l'expérience et l'intuition du gestionnaire. Ainsi, les différents facteurs peuvent être ajustés en adoptant un choix qui répond le mieux aux objectifs de coût et de temps.

Tableau XLIII

Facteurs d'échelle et multiplicateurs de l'effort du modèle COCOMO II [22]

Facteurs d'échelle	Multiplicateurs de l'effort
PREC (Precedentedness): FLEX (Development Flexibility): RESL (Architecture/Risk Resolution): TEAM (Team Cohesion): PMAT (Process Maturity):	Attributs Produit RELY (Required Software Reliability): Fiabilité requise DATA (<i>Data Base Size</i>): Taille de la base de données CPLX (<i>Product Complexity</i>): Complexité du logiciel RUSE (<i>Developed for Reusability</i>): développé pour la réutilisabilité DOCU (Documentation Match to Life-Cycle Needs): Attributs Projet TOOL (<i>Use of Software Tools</i>): Utilisation d'outils logiciels SITE (<i>Multisite development</i>): Développement de multi site SCED (<i>Required Development Schedule</i>): Contraintes de planification Attributs Personnel ACAP (<i>Analyst Capability</i>): Compétence des analystes PCAP (<i>Programmer Capability</i>): Compétence des programmeurs PCON (<i>Personnel Continuity</i>): Continuité des personnels APEX (<i>Application Experience</i>): Expérience du domaine d'application PLEX (<i>Platform Experience</i>): Expérience du matériel LTEX (<i>Language and Tool Experience</i>): Expérience du langage et l'outil Attributs Matériel (de plateforme) TIME (<i>Execution Time Constraint</i>): Contrainte du temps d'exécution STOR (<i>Main storage Constraint</i>): Contrainte de la taille mémoire PVOL (<i>Platform Volatility</i>): Plateforme virtuelle

Dans le modèle COCOMO II, les 17 facteurs pris comme étant les principaux facteurs affectant le coût de développement, représentent l'environnement de développement du logiciel. Par exemple, la compétence des analystes, l'expérience dans le langage de programmation utilisé et les contraintes sur les délais de développement. Ils sont regroupés en quatre classes (Tableau XLIII). L'effet sur le coût de chacun des 17 facteurs est exprimé en termes de ratio de productivité, qui est le rapport entre la

productivité (exprimée en KSLOC par Homme-Mois) d'un projet exploitant au mieux le facteur en question et la productivité d'un projet n'exploitant pas ce facteur (Tableau XLIV);

E de l'équation (3), est déterminé par l'équation de l'agrégation de cinq facteurs d'échelle pour capturer les économies et « dés-économies » d'échelle rencontrées dans les projets logiciels de différentes tailles :

- si $E = 1.0$, les économies et « dés-économies » d'échelle sont en balance;
- si $E > 1.0$, le projet montre une « dés-économie » d'échelle. Ceci est causé par exemple par des facteurs principaux comme : la croissance des communications entre le personnel et la croissance de l'intégration dans les grands systèmes;
- si $E < 1.0$, le projet montre des économies d'échelle.

Les 17 multiplicateurs de l'effort sont utilisés dans le modèle pour ajuster l'effort nominal, PM , pour refléter le produit logiciel en développement. Chaque multiplicateur de l'effort est défini par un ensemble de pourcentage des niveaux et un ensemble correspondant des multiplicateurs de l'effort. Le multiplicateur Moyen a la valeur du pourcentage = 1.0.

Les valeurs indiquées dans le Tableau XLIV, et le modèle même a été étalonné sur une base de 165 projets de trois types (médical, etc.). Ce sont les mêmes projets depuis une vingtaine d'années, qui utilisent le nombre de LOC pour mesurer la taille du produit. Mais il faut tenir compte de l'évolution des pratiques en génie logiciel, ce modèle peut ne pas être pratique (hypothèse à vérifier).

Tableau XLIV

Les facteurs du COCOMO II et leurs coefficients

Attribut (facteurs)	Valeur					
	Très bas	Bas	Moyen	Élevé	Très élevé	Extra-élevé
RELY	0,82	0,92	1,00	1,10	1,26	
DATA		0,90	1,00	1,14	1,28	
CPLX	0,73	0,87	1,00	1,17	1,34	1,74
RUSE		0,95	1,00	1,07	1,15	1,24
DOCU	0,81	0,91	1,00	1,11	1,23	
TIME			1,00	1,11	1,29	1,63
STOR			1,00	1,05	1,17	1,46
PVOL		0,87	1,00	1,15	1,30	
ACAP	1,42	1,19	1,00	0,85	0,71	
PCAP	1,34	1,15	1,00	0,88	0,76	
PCON	1,29	1,12	1,00	0,90	0,81	
APEX	1,22	1,10	1,00	0,88	0,81	
PLEX	1,19	1,09	1,00	0,91	0,85	
LTEX	1,20	1,09	1,00	0,91	0,84	
TOOL	1,17	1,09	1,00	0,90	0,78	
SITE	1,22	1,09	1,00	0,93	0,86	0,80
SCED	1,43	1,14	1,00	1,00	1,00	

Dans le modèle COCOMO II [21;22], plusieurs paramètres sont décrits sur une échelle ordinale et leur influence est déterminée par les opinions d'experts plutôt que sur la base d'information d'un référentiel descriptif et prescriptif de l'ingénierie. Le choix des coefficients pour ajustement, bien que justifié et discuté, n'a pas été vérifié et validés dans d'autres environnements à plus large envergure. Le Tableau XLV présente dans quelle étape du cadre de référence les éléments du modèle COCOMO sont adressés. Ce modèle se positionne entièrement dans l'étape 4 du cadre de référence et ne contient aucun élément pertinent aux autres étapes du processus de mesure inclus dans le cadre de référence proposé.

Tableau XLV

Alignement du modèle COCOMO II [22] avec le cadre de référence

Cadre de référence	Étape 1 Design	Étape 2 Application	Étape 3 Résultats	Étape 4 Utilisation
Modèle COCOMO				
Modèle post-architecture				X

Ceci montre que dans le domaine de la gestion des projets logiciels, l'accent est mis sur l'étape 4 du cadre de référence : toutefois, une des conditions pour que les résultats de l'étape 4 soient bons et utiles, il faut que ses entrées qui proviennent des étapes antérieures sont de bonne qualité : par exemple, il faut que les résultats de l'étape 3 soient précise, ce qui requiers que l'étape 2 doit être accomplie correctement ainsi que dans l'étape 1 il faut avoir de bon instrument de mesure. Ainsi, ce modèle COCOMOII a été discuté uniquement de la perspective de l'utilisation de mesure.

7.4 Sommaire

Les activités du processus de mesure de ISO 15939 ne traite ni les concepts de design de la mesure en-soi, ni les résultats obtenus. Il est nécessaire de concevoir des améliorations pour avoir beaucoup plus de précision et moins d'ambiguïté. Cette norme rencontre une partie du VIM, mais il manque les éléments pour vérifier les résultats de mesure avant leurs exploitations, il manque aussi les concepts liés au design de mesure. Les caractéristiques des instruments de mesure et de l'étalon sont requises. Toutes ces notions de base permettent de normaliser les indicateurs de qualité et de mieux construire par exemple les modèles d'estimation.

Dans un modèle prédictif, la valeur d'un ou plusieurs attributs peut être utilisée pour prédire la valeur d'une autre. Par exemple : un modèle de coût utilise souvent la taille pour prédire l'effort. Dans ce type de modèle, l'unité de la variable d'output est différente de celle de la variable d'input. Le modèle COCOMO utilise la taille en termes

de KSLOC comme une variable de l'input et l'effort en terme homme-mois (PM) comme une variable de l'output, dans la formule de l'équation (1).

Dans COCOMO, les projets sont différents avec des domaines différents. La complexité ou la difficulté ne peuvent être jamais quantifiées. Elles peuvent être seulement classées (plus facile, ..., plus difficile).

Le fait que le design d'une mesure n'est pas bien défini (une définition claire d'une mesure veut dire que la caractérisation et le méta-modèle sont bien définis), l'interprétation des résultats obtenus ne va pas être consistante, même si elles ont été bien appliquées par les praticiens. Pour bien exploiter les résultats obtenus de mesure, il faut que les résultats soient corrects. Pour s'assurer que les résultats sont bien corrects, il faut bien appliquer les règles de mesurage. Pour vérifier que les règles de mesurage sont bien appliquées, il faut que le design d'une mesure soit bien défini et bien vérifié.

CHAPITRE 8

ANALYSE DU CADRE PROPOSÉ PAR KITCHENHAM *et al.*

8.1 Introduction

Dans la littérature du génie logiciel, de nombreuses mesures ont été proposées pour plusieurs sujets d'intérêt de l'industrie, telles que les mesures de la complexité du logiciel¹⁰ ou les éléments orientés objets [29;34;47;49;85;113;114]. Ces mesures sont souvent décrites sous forme d'un algorithme avec l'utilisation de termes tels que « processus », « flot de données » sans aucune définition précise pour chacun de ces termes dans l'algorithme proposé. Ce manque de précision explique pourquoi les résultats obtenus par ces mesures sont dépendants du contexte, c'est-à-dire dépendants sur la façon dont les mesureurs comprennent les termes. À cet effet, plusieurs auteurs se sont intéressés sur le thème de « validation de mesure en génie logiciel » pour aider l'industrie à évaluer et choisir parmi les diverses mesures proposées [41;82;100;104].

Qu'est-ce qu'une mesure valide en génie logiciel? Et, comment la valider? Plusieurs auteurs ont tenté de répondre à ces questions, mais toujours selon différentes perspectives (mathématiques, pratiques, etc.). Par exemple, Schneidewind propose une méthodologie de « validation » de mesure basée sur six critères de validité [100]. Par contre, d'autres auteurs postulent qu'une méthode de mesure est valide si et seulement si elle donne une caractérisation numérique propre de certains attributs [41;114]. Néanmoins, cette définition est loin d'être unanimement acceptée, et Fenton remarque qu'une mesure valide doit nécessairement appartenir à un système de prédiction valide [41]. Kitchenham *et al.* [82] mentionnent ce qui manque est une discussion appropriée sur les relations entre les différentes approches. En effet, il est nécessaire de reconnaître

¹⁰ Comme exemple, dans son livre intitulé *Software Complexity: Measures and Methods*, Zuse a dénombré à peu près 100 mesures de la complexité logicielle.

le besoin d'un cadre de vérification qui prend en considération et intègre les différentes perspectives de « vérification » proposées par différents auteurs. Kitchenham *et al.* [82] proposent ainsi un tel cadre de vérification, et encouragent les chercheurs et praticiens de répondre par des critiques. Ce chapitre présente une analyse structurée du cadre de vérification de mesure proposé par Kitchenham *et al.* [82] en utilisant le cadre de référence de mesure tel que défini dans le chapitre 5.

8.2 Cadre de vérification dans [82] et entités types correspondantes

Comme décrit dans la section 3.7 du chapitre 3, le cadre de vérification proposé par Kitchenham *et al.* [82] comporte trois sections. Chacune des trois sections présente divers concepts de mesure. Pour analyser ce cadre, nous utilisons les différentes entités de mesure sujettes aux critères de vérification d'une mesure, comme décrite dans le Tableau XXVIII. Ainsi, nous commençons par la mise en correspondance des concepts de mesure dans [82] avec l'ensemble des concepts du Tableau XXVIII. Cette mise en correspondance est illustrée dans le Tableau XLVI. Pour chaque section, nous présentons l'ensemble des concepts clés de mesure et les critères identifiés par Kitchenham *et al.* [82] (dans chaque ligne) en se basant sur l'ordre séquentiel de leur cadre. Par la suite, ces concepts de mesure vont être examinés et positionnés selon les critères du cadre de référence. Les « X » indiquent ainsi les concepts qui sont couverts par le cadre de référence dans la colonne respective.

Tableau XLVI

Mise en correspondance du cadre de vérification dans [82] avec le cadre de référence

Sections dans [82]	Concepts de mesure discutés dans chaque section du cadre de Kitchenham <i>et al.</i> [82]	Étape 1 Design	Étape 2 Application	Étape 3 Résultat	Étape 4 Utilisation
Structure d'une mesure	Entités, attributs et leurs relations	X			
	Unités, types d'échelle et leurs relations	X			
	Valeurs	X		X	
	Propriétés des valeurs	X			
	Instrument de mesure		X		

Tableau XLVI (Suite)

Sections dans [82]	Concepts de mesure discutés dans chaque section du cadre de Kitchenham <i>et al.</i> [82]	Étape 1 Design	Étape 2 Application	Étape 3 Résultat	Étape 4 Utilisation
Structure d'une mesure	Mesures indirectes				X
	Unités composées				X
	Propriétés des mesures indirectes				X
	Implications pour la validation des mesures	X		X	
Modèles et mesurage	Modèles de définition de l'unité	X			X
	Modèles de l'instrumentation		X		
	Modèles de relation entre attributs				X
	Problèmes pratiques avec les modèles d'attributs	X			X
	Protocoles de mesure		X		
	Différences entre les modèles de définition	X	X		
	Modèles de population de l'entité			X	
Validation de mesures de logiciels	Validation théorique et empirique	X	X	X	X
	Problème de validation théorique de mesure	X	X		
	Corroboration empirique de mesure	X		X	X
	Validation empirique des relations entre attributs				X

Le Tableau XLVI montre que dans chacune des trois sections du cadre de vérification proposé par Kitchenham *et al.* [82], il y a une discussion des concepts de mesure traitant les différents concepts de mesure. Mais, ces concepts chevauchent. En effet, ces concepts de mesure ne sont pas traités d'une façon structurée comparativement à ceux du cadre de référence que nous avons proposé. De plus, une lecture attentive du texte [82] indique que la même « expression » (ou le même terme) peut se référer à deux concepts distincts de mesure (par exemple : le modèle de définition de l'unité qui se réfère à la fois aux concepts liés à l'étape 1 du design d'une méthode de mesure et à l'étape 4 de l'exploitation des résultats de mesure). Ainsi, les différents niveaux des concepts sont mêlés dans [82] : il y a un passage continu d'un niveau à un autre et il est souvent difficile de distinguer entre ces concepts sans avoir recours à notre cadre de référence de mesure. De plus, l'expression « système de relations empiriques » telle que

décrite dans [82] est utilisée dans deux contextes différents, chacun ayant ses exigences particulières :

- le premier contexte est le système de relations empiriques entre le monde réel et le monde mathématique pour un seul attribut;
- le second contexte est le système de relations empiriques pour plusieurs attributs.

Chacun de ces contextes sera discuté en tenant compte des concepts de la métrologie et des éléments (concepts et critères) du processus de mesure.

8.3 Analyse structurée du cadre de vérification proposé dans [82]

Dans cette section, nous proposons de vérifier la couverture des concepts de mesure identifiés dans le cadre proposé par [82] par rapport à ceux définis dans notre proposition de cadre de référence. Par la suite, nous proposons certaines clarifications et améliorations.

8.3.1 Design d'une méthode de mesure : éléments à vérifier

8.3.1.1 Entités de mesure comme objets de vérification dans le design

Pour mesurer, il est nécessaire d'utiliser le design d'une méthode de mesure (si un tel design existe), ou d'en concevoir un (s'il n'en existe pas). Il est mentionné dans [76] que pour construire une méthode de mesure, il est nécessaire de définir les objectifs, de sélectionner le méta-modèle du concept ou de l'objet à mesurer, de caractériser le(s) attribut(s) à mesurer et de définir les règles d'assignation numérique incluant les grandeurs et unités. La question est de vérifier si les éléments de design d'une méthode de mesure sont présents dans [82], c'est-à-dire, si le cadre de Kitchenham *et al.* couvre complètement l'ensemble détaillé des sous-étapes identifiées dans [6;13;76]?

La Figure 35 présente notre modèle hiérarchique des concepts de mesure mentionnés dans [82] comme reliés à l'étape de design d'une méthode de mesure, c'est-à-dire : entités, attributs, unités, types d'échelle et valeurs.

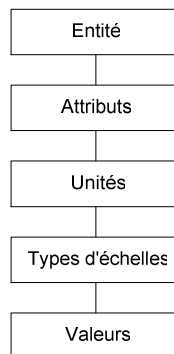


Figure 35 Modèle hiérarchique des concepts de design d'une méthode de mesure, comme discuté dans [82]

En se basant sur les éléments présentés dans la Figure 35 et ceux de la Figure 29, nous remarquons que les concepts reliés à l'entité de mesure et ses attributs sont clairement énoncés dans [82] tandis que les règles d'assignation numérique sont partiellement adressées en terme d'unités, de types d'échelle et de valeurs. Par contre, la référence entre le monde réel et le monde mathématique pour un seul attribut n'est pas assez précise dans [82] et ne spécifie pas les concepts requis pour la vérification de design d'une méthode de mesure, c'est-à-dire les objectifs d'une méthode de mesure, la caractérisation des concepts (à travers les attributs), le méta-modèle de ces attributs dérivés pour rencontrer les objectifs de la méthode ou les règles pour l'assignation numérique des unités et valeurs.

8.3.1.2 Combinaison des attributs comme objet à vérifier

Le second modèle proposé dans [82] consiste à représenter un phénomène décrit par une combinaison d'attributs, qui se réfère comme étant une « mesure indirecte ». Par exemple, le concept de la vitesse est défini par la combinaison de deux autres mesures

(soit la distance et le temps), et ce concept de vitesse est représenté par le ratio « distance parcourue par unité de temps » ou distance/temps, et que les attributs multidimensionnels (tels qu'en physique) peuvent être représentés (définis) par des vecteurs (tels que l'accélération qui est définie par la combinaison de la vitesse et le temps) ou par des scalaires (tels que la vitesse qui est mesurée en termes de distance par unité de temps). Ces exemples présentés dans [82] se réfèrent aux représentations des mesures indirectes, et il y est énoncé que l'équation définissant ces mesures « indirectes » se base sur une association empirique entre des attributs, formalisés par une équation mathématique (l'exemple donné est l'utilisation de la taille d'un programme dans une équation pour prédire l'effort du projet). Cependant, une étude détaillée du texte dans [82] révèle que les éléments présentés comme critères pour la vérification ne sont pas reliés aux concepts illustrés dans l'exemple donné, mais plutôt à d'autres concepts qui se réfèrent à la définition des ratios, des indicateurs ou des formules pour décrire une combinaison d'attributs pour un objet donné. Par conséquent, les auteurs [82] n'expliquent pas la relation entre attributs des entités distinctes. Dans la Figure 36, nous proposons un modèle des concepts d'association entre attributs sur la base de la description du texte de [82] (plutôt que les Figures 2 et 3 dans [82]).

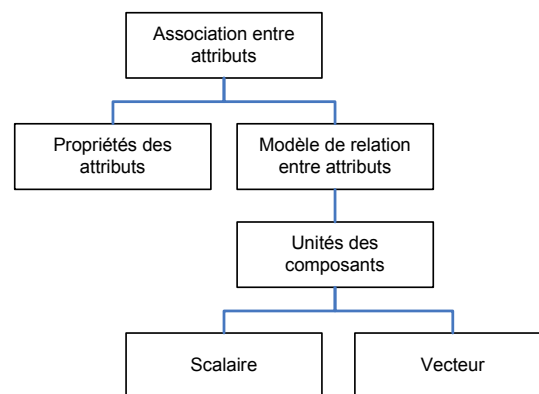


Figure 36 Modèle d'association entre attributs sur la base du texte dans [82]

Le modèle ISO de la qualité d'un produit logiciel décrit dans ISO 9126-1 [59] est un exemple représentatif d'une instanciation de l'association entre attributs, comme illustré dans la Figure 36. Ce modèle de qualité de ISO 9126, comme nous l'avons mentionné dans le chapitre 6, représente la qualité du logiciel à travers six caractéristiques de qualité, chacune des caractéristiques est décomposée en sous-caractéristiques, chaque sous-caractéristique est décrite en termes numériques à travers une sélection de mesures proposées décrites soit par un seul nombre ou une combinaison de ratios et indicateurs (qui devrait être représenté par des scalaires ou vecteurs) ; ISO 9126 propose un modèle des relations entre attributs. En effet, bien que cette norme représente des descriptions détaillées de son méta-modèle, elle n'est pas prescriptive dans le sens qu'elle ne prescrit ni une sélection spécifique des caractéristiques, ni l'ensemble d'assignation numérique spécifique dans le modèle de relation prescrit.

8.3.1.3 Un des systèmes de relations empiriques : la relation entre un attribut et sa mesure

Les attributs sont les propriétés d'une entité, et par définition selon Kitchenham *et al.* [82], une « mesure » transforme les attributs en nombres réels. Par conséquent, pour un attribut donné, il y a une relation dans un monde empirique qui peut être capturée formellement dans un monde mathématique. L'exemple donné dans [82] est celui de deux personnes dont l'un est plus grand que l'autre. Une « mesure » nous permet alors de capturer la relation « est plus grand que », pour la transformer dans un système formel et pour l'explorer mathématiquement. Cependant, bien que cet exemple se base sur deux références (deux personnes différentes), ces références explicites ne sont pas formellement énoncés dans [82]. De plus, il n'est pas reconnu explicitement que cette référence devrait être établie, et qu'elle ne devrait pas changer lors d'une instanciation de mesure. Par ailleurs, l'exemple donné se réfère à des nombres, ces nombres sont de type d'échelle « ordinal », sans aucune référence à l'étalon « échelle ».

Un cadre de vérification de mesure amélioré devrait donc inclure d'autres critères additionnels qui se basent sur la métrologie, telles que les références de mesure pour les conditions de représentation, incluant les références liées à l'étalon de mesure.

Le Tableau XLVII présente maintenant une proposition de critères à utiliser pour vérifier le design d'une méthode de mesure : les règles de modélisation du logiciel pour le méta-modèle, les « étalons » de mesure ou des « études de cas » pour les références de mesure, et les règles d'assignation numérique pour la représentation mathématique.

Tableau XLVII

Design d'une méthode de mesure – critères de vérification

Design d'une méthode de mesure		
Méta-modèle	Référence de mesure	Représentation mathématique
Règles de modélisation du logiciel	Étalon Études de cas	Règles d'assignation numérique

8.3.2 Vérification de l'application d'une méthode de mesure

L'étape 2 dans la Figure 2 du chapitre 2 se réfère à l'application d'une méthode de mesure dans une instantiation spécifique. Dans cette étape, la vérification requiert à la fois la vérification des instruments de mesure et de leurs caractéristiques lors de l'instanciation.

8.3.2.1 Concepts de mesure selon Kitchenham *et al.*

Le type de vérification mentionné ci-dessus est abordé dans le texte de Kitchenham *et al.* [82] dans la deuxième partie du cadre intitulée « modèles et mesure » qui inclut l'ensemble des éléments suivants : modèle de définition de l'unité, modèle de l'instrumentation, modèle de relation entre attributs, les problèmes pratiques avec les modèles d'attributs, les protocoles de mesure, les différences entre les modèles de définition de l'attribut et les modèles de population de l'entité. Les Figure 37 et Figure

38 présentent notre modélisation de l'ensemble des principaux concepts : « modèle de définition de l'unité » et « processus de mesure » tel que présenté et classé textuellement dans [82].

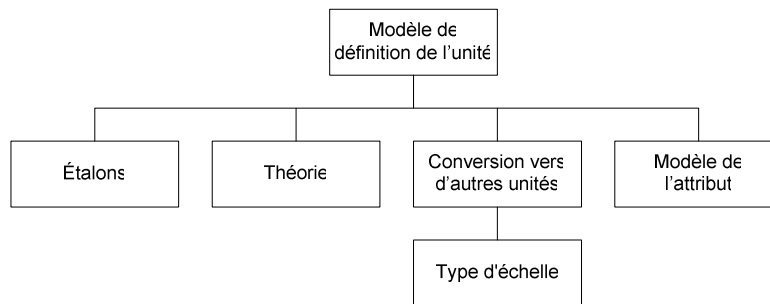


Figure 37 « Modèle de définition de l'unité » sur la base du texte dans [82]

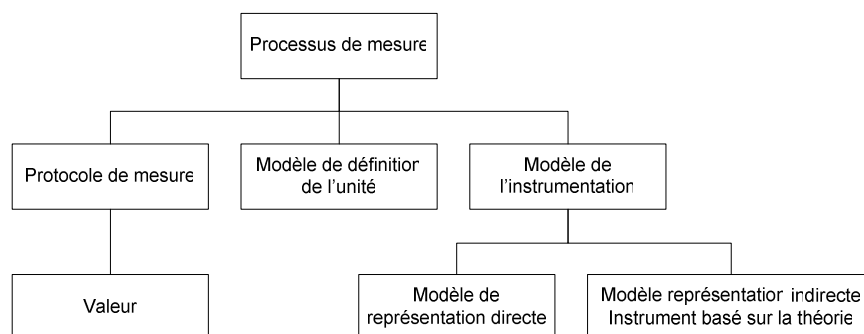


Figure 38 « Processus de mesure » sur la base du texte dans [82]

Selon notre point de vue, le modèle de définition de l'unité de la Figure 37 devrait toutefois appartenir au design de la méthode de mesure plutôt qu'à une instantiation spécifique. Par contre, le processus de mesure (décrit dans [82] et présenté dans la Figure 38) est une perception différente d'un mode opératoire du VIM'93 telle que présentée dans la Figure 9. Ainsi, la vérification de l'application d'une méthode de mesure devrait se référer à la vérification de l'instrument de mesure et les caractéristiques de cet instrument dans une instantiation spécifique.

8.3.2.2 Vérification de l'instrument de mesure

Si l'on considère qu'un instrument de mesure est bien vérifié, alors nécessairement ses caractéristiques devraient être vérifiées dans un environnement donné. La vérification des caractéristiques d'un instrument de mesure devrait tenir compte des aspects d'utilisation et de contrôle (tels que présentés dans le Tableau XIV).

8.3.2.3 Vérification des caractéristiques de l'instrument de mesure

Parmi la vérification des caractéristiques de l'instrument de mesure, il est nécessaire de vérifier si les règles de modélisation du logiciel sont bien définies pour l'instrument pour mesurer ce qu'il est supposé mesurer. Ceci est modélisé dans la Figure 39 où les flèches à double sens indiquent une influence réciproque entre deux éléments. Les flèches simples indiquent une influence causale dans un seul sens.

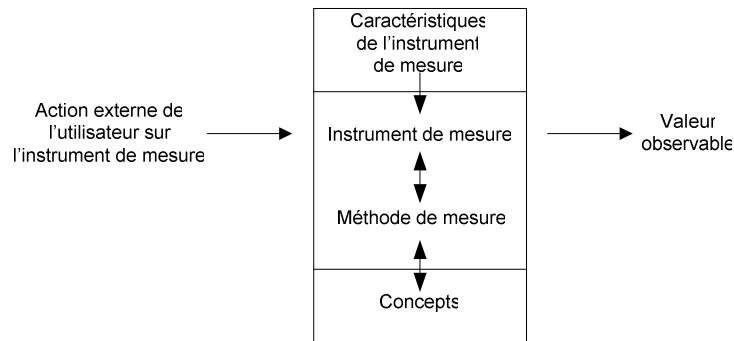


Figure 39 Éléments de vérification d'un instrument de mesure

8.3.3 Éléments de vérification pour les résultats obtenus en mesure

Suite à l'application des règles d'une méthode de mesure, les résultats obtenus peuvent être exploités comme une source de connaissance selon le monde analytique défini par convention, par une méthode de mesure utilisée. Cet aspect est adressé dans chaque section du document de Kitchenham *et al.* [82] pour identifier le besoin pour la vérification des résultats des méthodes de mesure.

8.3.4 Vérification de l'utilisation des résultats de mesure

8.3.4.1 Modèles d'exploitation des résultats de mesure

Une fois que les résultats de mesure ont été obtenus, ils peuvent être utilisés dans des modèles quantitatifs ou qualitatifs pour une utilisation spécifique, tels que le modèle de qualité, le modèle d'évaluation, le modèle descriptif des relations, le modèle de prédiction, etc. Dans [82], ceci se réfère au modèle des « attributs des composants », incluant les divers concepts modélisés dans la Figure 40 sur la base du texte de ces auteurs.

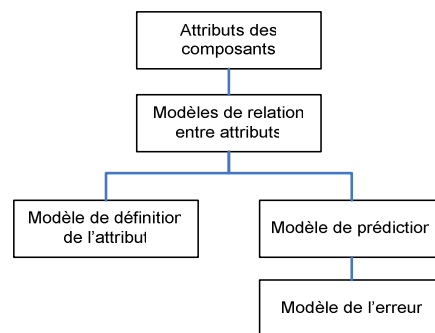


Figure 40 Attributs des composants sur la base du texte de [82]

En génie logiciel, plusieurs auteurs ont abordé le problème de « validation » (vérification) de mesure de logiciels selon différentes perspectives. Il manque encore un cadre complet adressant les différentes parties du processus d'une méthode de mesure. Par exemple, Schneidewind a proposé dans [100] un processus de validation basé principalement sur l'analyse des résultats des méthodes de mesure (étape 3 : obtention des résultats de mesure (Figure 2)). De son côté, Fenton suggère qu'une méthode de mesure valide doit satisfaire aux conditions de représentation de la théorie de mesure [40;41]. En d'autres termes, Fenton propose un processus de validation qui adresse seulement les relations entre deux sous-étapes « caractérisation du concept à mesurer » et « définition des règles d'assignation numériques » de l'étape 1 du processus de mesure

(Figure 29). Cependant, ceci est plutôt une vision exceptionnelle de « vérification » de mesure, car la plupart des concepts discutés par ces auteurs ne se réfère pas à un domaine détaillé de mesure mais plutôt au domaine des études expérimentales ainsi que le domaine des outils d'analyse des données (tels que les modèles statistiques, etc.).

Il est à noter en particulier que les exigences de validation sont plus contraignantes que les exigences de vérification. Par exemple, on peut vérifier la performance ou la « qualité » d'un modèle d'estimation en utilisant plusieurs critères, tel que le coefficient de la détermination de régression (R^2), le coefficient de prédiction – *Pred* (25 %) < dans 75 % des cas. Un processus de vérification fournit des valeurs quantitatives pour évaluer un modèle dans un contexte spécifique. Cependant, ces valeurs ne permettent pas de « valider » un modèle. Par exemple, un modèle de régression est valide selon ses conditions de construction et d'interprétation mais non pas selon des valeurs trouvées dans un contexte particulier.

8.3.4.2 Relations entre multiples attributs

Selon Kitchenham *et al.* [82], la vérification des relations empiriques entre plusieurs attributs nécessite de :

- contrôler si la relation est une association ou une relation causale. En effet, l'observation d'une corrélation statistiquement significative n'est pas suffisante pour confirmer une relation causale. Pour confirmer la causalité, il est nécessaire de contrôler l'attribut qui cause la relation (variable indépendante) et confirmer que le changement prévu est observé dans l'autre attribut (variable dépendante). Cependant, il est souvent difficile de contrôler les attributs logiciels. Dans ce contexte, l'analyse de Kitchenham *et al.* n'est pas basée sur l'ensemble du domaine de connaissance de la mesure (métrologie), mais plutôt sur le domaine des études expérimentales (et leurs méthodologies) en général;
- considérer si la méthode d'investigation d'une relation est susceptible au changement de la relation elle-même. Par exemple, pour vérifier si un modèle de

coût produit des prédictions exactes, il est approprié de comparer l'effort et la durée prévue avec l'effort et la durée réelle;

- considérer la différence entre la qualité d'ajustement et la capacité prédictive.

Il est énoncé dans [82] que pour établir si une « mesure » peut être utilisée pour prédire la valeur d'une autre « mesure », il est nécessaire d'établir s'il y a une relation entre les deux attributs à mesurer. De même, lorsque Schneidewind suggère dans son article *Methodology for Validating Software Metrics* [100] que le contrôle des relations entre les attributs devrait être une méthode fondamentale pour la « validation des mesures de logiciels », notre analyse de ce texte [100] indique que cet auteur n'a pas traité la validation du design d'une mesure pour la représentation numérique d'un attribut, mais plutôt l'utilisation de ces attributs mesurés dans certains contextes (exemple, dans les modèles d'utilisation comme l'évaluation, le contrôle et la prédiction d'autres attributs).

8.4 Résumé des observations et recommandations

L'objectif derrière le cadre de vérification de Kitchenham *et al.* était de proposer une base pour un consensus sur la « vérification » de mesure, car ceci est considéré critique pour le succès des mesures de logiciels. Cependant, comme nous l'avons vu, dans ce domaine de connaissance de mesure de logiciels, il y a encore un bon nombre d'ambiguïtés comme nous l'avons démontré.

En se basant sur les étapes du cadre de référence, nous avons positionné les critères de vérification proposés par les auteurs [82]. Ce positionnement a permis d'identifier la couverture des concepts de mesure. Il a aussi permis d'identifier les contributions potentielles pour un cadre de vérification complet.

Nous avons montré que dans [82], il n'y a pas de distinction entre les concepts et sous-concepts de mesure (i.e. les étapes du cadre de référence) auquel s'applique les critères de vérification proposés. En effet, l'analyse du cadre proposé par Kitchenham (qui est

résumée dans le Tableau XLVIII) montre qu'il ne couvre pas l'ensemble des concepts du cadre de référence. Par exemple, lorsque les entités, les attributs, les unités et les conditions de représentation sont discutés dans [82], ils traitent seulement quelques concepts de design d'une méthode de mesure. En effet les règles d'assignation numérique sont partiellement adressées et le méta-modèle n'est pas abordé. Quelques concepts liés à l'application d'une méthode de mesure sont traités. De plus, l'obtention des résultats et leurs utilisations ont été discutés mais avec différents critères.

De même certains critères (les mêmes critères) peuvent se référer non pas à un seul concept mais à plusieurs concepts distincts, exemple : « différence entre les modèles de définition » qui se réfère à la fois au design d'une méthode de mesure et à son application. Par conséquent, il est nécessaire de préciser l'utilisation de la terminologie de mesure pour éviter toute ambiguïté, de clarifier les entités types à vérifier et de comprendre les propriétés correspondantes qui doivent être adressées dans un cadre de vérification d'une mesure. Par exemple, dans [82], lorsque les termes *measure* et *measurement* sont utilisés pour traiter un objet générique de vérification, nous avons illustré l'utilité d'un vocabulaire plus précis dérivés du domaine de connaissance de la mesure dans le VIM.

Tableau XLVIII

Classification des éléments de vérification des méthodes de mesure du logiciel

Cadre de référence de mesure	Cadre proposé par Kitchenham <i>et al.</i> [82]
Étape 1: Design d'une méthode de mesure	
Définition des objectifs	Introduction
Design ou sélection du métamodèle	Non spécifié
Caractérisation du concept à mesurer	Entités, attributs et leurs relations
Définition des règles d'assignation numérique	Unités, types d'échelles et leurs relations Valeurs Propriétés des valeurs Implications pour la validation des mesures Modèle de définition de l'unité Problèmes pratiques avec les modèles d'attribut Différence entre les modèles de définition Validation théorique et empirique Problèmes de validation théorique de mesure Corroboration empirique de mesure
Étape 2: Application d'une méthode de mesure	
Collection de la documentation du logiciel	Non spécifié
Construction du modèle du logiciel	Non spécifié
Application des règles d'assignation numérique	Instrument de mesure Modèles d'instrumentation Protocoles de mesure Différence entre les modèles de définition Validation théorique et empirique Problèmes de validation théorique de mesure
Étape 3: Résultat de mesure	
Résultat de mesure	Valeurs Implications pour la validation des mesures Modèles de population de l'entité Validation théorique et empirique Corroboration empirique des mesures
Étape 4: Exploitation des résultats de mesure	
Utilisation des résultats de mesure	Mesures indirectes Unités composées Propriétés des mesures indirectes Modèles de définition de l'unité Modèles de relations entre attributs Problèmes pratiques avec les modèles d'attribut Validation théorique et empirique Corroboration empirique des mesures Validation empirique des relations entre les attributs

8.5 Sommaire

Les propositions de vérification pour les modèles de mesure en génie logiciel nécessitent donc d'être consolidées et améliorées en tenant compte des conventions internationales sur la terminologie utilisée et l'introduction des concepts de la métrologie pour apporter des informations plus précises sur plusieurs aspects de mesure, notamment sur les instruments de mesure et les caractéristiques de ces instruments.

L'utilisation du cadre de référence a montré, par exemple, que le design d'une méthode de mesure et l'utilisation des résultats de mesure ont été discutés selon différents critères de mesure. De même, il est à noter dans [82] que l'expression « ensemble de relations empiriques » se réfère non pas à un seul concept mais à deux concepts distincts : condition de représentation d'un seul attribut et relations entre multiples attributs, et même parfois sur des objets de nature distincte (telle que la taille, l'effort dans un modèle causal avec des variables dépendantes et indépendantes). En utilisant à la fois les concepts reliés à la métrologie VIM et au processus des méthodes de mesure, nous avons illustré que, même dans le cadre de la vérification proposée dans [82], il y a certaines ambiguïtés dans la définition des concepts de mesure.

L'impact sur les praticiens en génie logiciel est que la plupart des mesures proposées n'ont pas été conçues à travers un cycle de design ou à travers un contrôle et une vérification de données appropriées, adéquates et intensives, que ce soit à un haut niveau ou à un niveau détaillé de vérification des méthodes de mesure de logiciels.

CHAPITRE 9

ANALYSE DU CORPUS DES CONNAISSANCES DE LA MESURE EN GÉNIE LOGICIEL

9.1 Introduction

L'*IEEE Computer Society*, avec le support des commanditaires industriels, a publié en 2005 le guide au corpus des connaissances en génie logiciel (ISO TR 19759 : SWEBOK¹¹). L'objectif principal de ce guide est de développer un consensus international sur les « connaissances généralement acceptées » dans le domaine du génie logiciel. Dans ce guide, il y a dix domaines de connaissance et trois thèmes généraux : qualité, outils et mesure. Le thème de la mesure y est traité dans tous les domaines de connaissance (dans les éditions 2001 et 2004). Cependant, ce thème ne fait pas partie d'un domaine de connaissance distinct. Dans ce chapitre, nous explorons les raisons pour lesquelles le thème de la mesure n'a pas été reconnu comme étant un domaine de connaissance, et nous proposons des recommandations qui pourront contribuer à la reconnaissance de ce thème en tant qu'un domaine de connaissance distinct.

9.2 Survol du guide SWEBOK

Un des projets conçu pour contribuer spécifiquement à la reconnaissance du génie logiciel comme une discipline *bona fide* de l'ingénierie est le guide SWEBOK (*Software Engineering Body of Knowledge*). Ce guide (2001 *Trial version*) [8] contient dix domaines de connaissance (KA) en génie logiciel et trois thèmes généraux : qualité, outils et mesure. Les descriptions des KA sont conçues pour distinguer divers concepts importants, permettant aux lecteurs de trouver rapidement les sujets qui leurs intéressent. Une fois le sujet trouvé, les lecteurs sont référés à des articles ou des chapitres

¹¹ Le projet SWEBOK a été appuyé par les organisations suivantes : Boeing, Raytheon, MITRE Corporation, National Institute of Standards and Technology (USA), Construx Software, Rational Software, SAP Lab. Canada, NRC, and Canadian Council of Professional Engineers.

sélectionnés de livres présentant succinctement les connaissances. D'autre part, les descriptions des KA du génie logiciel considèrent non seulement ce qui est généralement accepté actuellement mais également ce qui devrait être généralement accepté dans trois ou cinq ans. Les objectifs du guide sont les suivants :

- promouvoir une vue consistante dans le monde du génie logiciel;
- clarifier la place et l'ensemble des limites du génie logiciel en respectant les autres disciplines tels que l'informatique, la gestion des projets, le génie de l'informatique et les mathématiques;
- caractériser le contenu de la discipline du génie logiciel;
- obtenir un accès au corpus de connaissance du génie logiciel;
- obtenir une base pour le développement d'un curriculum normatif et le matériel de certification et de licence individuelle.

Le premier de ces objectifs, une vue consistante dans le monde du génie logiciel, a été élaborée et supportée par un processus de développement qui a inclus la collaboration de près 500 réviseurs provenant de 41 pays¹². De plus, tout récemment, SWEBOK a été accepté comme un rapport technique ISO (ISO/IEC TR 19759 : 2004) [68].

Ce guide SWEBOK a été développé avec une approche à trois phases :

- 1^{ère} phase : *Straw Man version* –1997
- 2^e phase: *Stone Man version* –2001
- 3^e phase : *Iron Man version* – 2004

Le guide SWEBOK documente un consensus sur la structure des connaissances en génie logiciel, qui contient les dix domaines de connaissance (KA) : les cinq premiers représentent le processus primaire selon ISO 12207 [61] et les cinq autres domaines représentent les processus de support et organisationnels (Tableau XLIX). Le guide

¹² De plus amples informations sur le processus de développement se trouve dans la préface du guide SWEBOK sur le site : www.swebok.org

utilise une organisation hiérarchique pour décomposer chaque KA en un ensemble de thèmes avec des labels reconnus. Les divers domaines de connaissance ont une architecture commune, illustrée dans la Figure 41.

Tableau XLIX

Guide SWEBOK : KA [9]

Domaine de connaissance (KA) SWEBOK	ISO 12207 Types du processus
1. Exigences de logiciels	Processus primaire
2. Design de logiciels	
3. Construction de logiciels	
4. Test de logiciels	
5. Maintenance de logiciels	
6. Configuration de logiciels	Processus de support et organisationnels
7. Gestion de l'ingénierie de logiciels	
8. Processus de l'ingénierie de logiciels	
9. Outils et Méthodes de l'ingénierie de logiciels	
10. Qualité de logiciels	

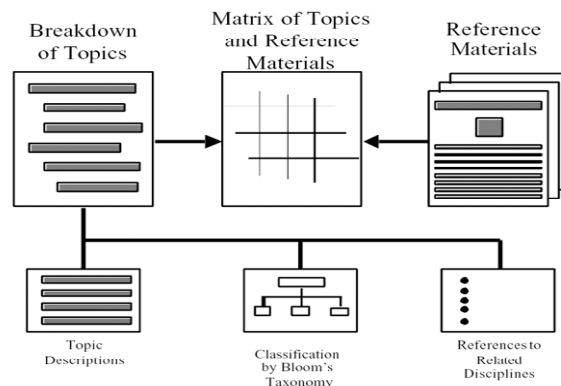


Figure 41 Organisation de la description du domaine de connaissance dans SWEBOK

Dans SWEBOK, il y a également trois thèmes généraux qui sont : qualité, outils et mesure. Toutefois, seulement les thèmes « qualité » et « outils » ont été reconnus comme

étant des domaines de connaissance distincts tandis que le thème de la mesure n'a pas été considéré comme un domaine distinct. Nous allons maintenant explorer les raisons pour lesquelles la mesure n'a pas été reconnue comme un domaine distinct de connaissance.

La mesure est un concept fondamental dans toute discipline de l'ingénierie, et sur cette base, il a été considéré comme un critère pour tous les éditeurs associés aux KA pour en identifier les connaissances généralement acceptées. Dans ce cas, des éditeurs distincts ont développé chacun des 10 KA individuellement, ce qui permet de conduire aux différents niveaux d'utilisation des sous-thèmes tels que la mesure. De plus, les connaissances relatives à la mesure n'ont pas été développées uniformément dans tous les KA dans l'histoire récente du génie logiciel.

Une proposition initiale d'unification des connaissances de mesure en génie logiciel a été proposée par Buglione et Abran [27] en tant qu'un KA distinct de mesure de logiciels, en prenant en compte tous les items relatifs à la mesure des domaines de connaissance (KA) existant dans l'édition 2001 du guide SWEBOK [8] et de les organiser en un item initial.

En 2003-2004, divers contributeurs ont amélioré le domaine de la mesure dans le guide SWEBOK, incluant dans certains KA où le traitement de la mesure était initialement faible, tel que par exemple le KA de la construction. Dans ce cycle de révision, deux réviseurs internationaux ont proposé d'ajouter un nouveau KA, mais ces propositions n'ont pas été acceptées parce que la mesure n'était pas encore devenue généralement acceptée par la communauté du génie logiciel, et ainsi aucune structure détaillée n'avait été validée par la communauté de la mesure en génie logiciel. Il est donc nécessaire d'analyser le domaine de connaissance de la mesure de logiciels d'une perspective de l'ingénierie pour ressortir les recommandations requises à la reconnaissance de ce thème comme étant un domaine de connaissance distinct.

9.3 Méthodologie de l'analyse du domaine de connaissance de la mesure

Comme nous l'avons mentionné dans le chapitre 2, il y a différents outils analytiques qui ont été conçus pour analyser les connaissances en ingénierie. Dans les sections suivantes, nous allons utiliser le corpus de connaissance pour les mesures de qualité de logiciels de Schneidewind [101], la classification des types de connaissance de l'ingénierie de Vincenti [109] ainsi que le cadre de référence de vérification de mesure (incluant les concepts de métrologie et du processus de mesure de Abran *et al.*[6]) pour examiner la maturité de la mesure en génie logiciel de point de vue de l'ingénierie. La Figure 42 présente la méthodologie adoptée :

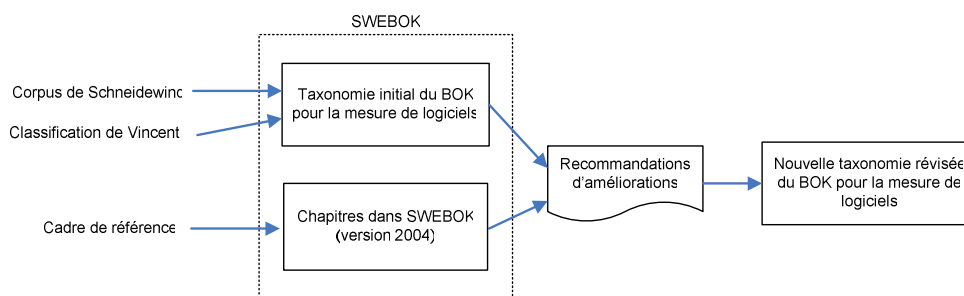


Figure 42 Méthodologie d'analyse

9.3.1 Taxonomie initiale du KA pour les mesures de logiciels dans SWEBOK

Une taxonomie initiale spécifique au domaine de connaissance de la mesure de logiciels avait été proposée par Buglione *et al.* [27] sur la base de la version 2001 du guide SWEBOK est présentée dans la Figure 43. Cette proposition est basée sur l'identification des thèmes de la mesure dans SWEBOK (concepts de mesure, processus de mesure, normes de mesure, mesure par phase du cycle de vie du logiciel, et outils et techniques). Une synthèse des différents éléments est faite ainsi que la vision des auteurs d'une taxonomie appropriée.

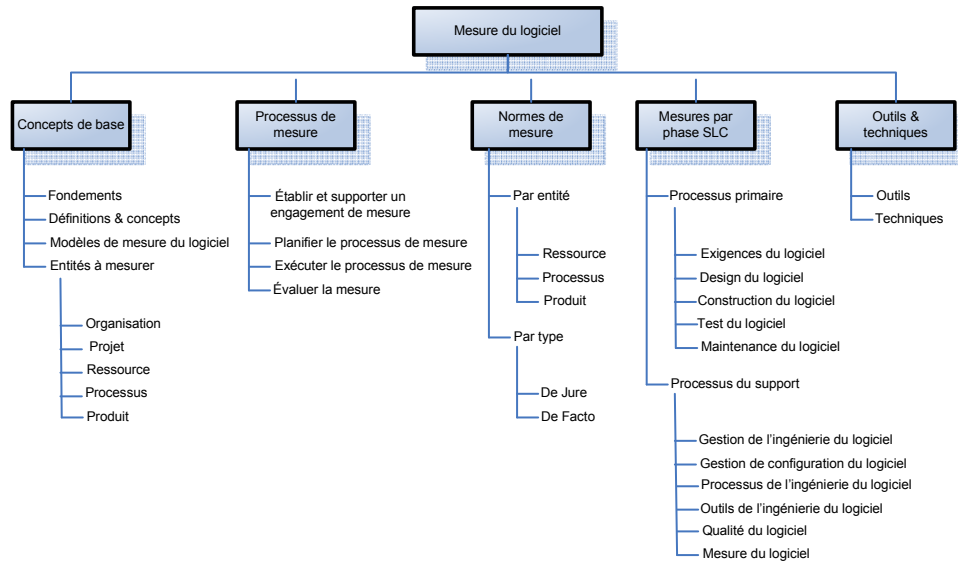


Figure 43 Taxonomie pour KA de mesure de logiciels dans SWEBOK [27]

Nous proposons de comparer cette taxonomie par rapport au corpus de connaissance proposé par Schneidewind [101]. Cette comparaison n'est que tentative, étant donné les différentes bases de structuration des deux corpus de connaissance. Le résumé de cette comparaison est présenté dans le Tableau L. Ce tableau énumère (en ligne) pour chaque thème de mesure proposé par Schneidewind [101] lesquels des concepts de mesure dans la taxonomie proposée [27] sont identifiés (dans la colonne respective). L'analyse de ce Tableau L nous montre par exemple que le thème sur les normes de mesure n'est pas identifié dans [101].

Tandis que le corpus de connaissance proposé par Schneidewind [101] consiste en une liste de séquence de neuf thèmes identifiés en mesure de la qualité d'un produit logiciel, la taxonomie proposée par Buglione *et al.* [27] est plus générale et plus structurée. En effet, dans Schneidewind [101], le thème 1 (objectifs) traite les entités à mesurer, le thème 2 (coûts et risques) et le thème 3 (contexte) dressent quelques éléments des concepts de base et une discussion des mesures de la phase du cycle de vie du logiciel

(SLC). Le thème 4 (profil opérationnel) et le thème 5 (modèles) identifient l'application du processus de mesure et les éléments des concepts de base. Le thème 6 (exigence des données) couvre les éléments d'un processus de mesure. Le thème 7 (type de mesure et granularité) traite les fondements des concepts de base et du processus de mesure. Le thème 8 (test et évaluation de produit et processus) traite certains éléments du processus de mesure et de mesure par SLC, et finalement le thème 9 (prédiction de la qualité de produit et processus) couvre quelques éléments du processus de mesure et des outils et techniques.

Tableau L

Mise en correspondance des thèmes de mesure de Schneidewind par rapport à la taxonomie de mesure du logiciel

Taxonomie Buglione et Abran / Taxonomie Schneidewind Thèmes	Concepts de base	Processus de mesure	Normes de mesure	Mesure par phase du cycle de vie du logiciel (SLC)	Outils & Techniques
Thème 1 : <i>Objectifs</i>	Entités à mesurer				
Thème 2 : Coût et risque	Modèle de mesure logiciel			Processus primaire Processus du support	
Thème 3 : <i>Contexte</i>	Entités à mesurer			Processus du support	
Thème 4 : Profil opérationnel	Entités à mesurer	Établir et supporter un engagement de mesure Planifier le processus de mesure Exécuter le processus de mesure Évaluer la mesure			

Tableau L (Suite)

Taxonomie Buglione et Abran Taxonomie Schneidewind Thèmes	Concepts de base	Processus de mesure	Normes de mesure	Mesure par phase du cycle de vie du logiciel (SLC)	Outils & Techniques
Thème 5: <i>Modèles</i>	Modèle de mesure logiciel Entités à mesurer	Établir et supporter un engagement de mesure Planifier le processus de mesure Exécuter le processus de mesure Évaluer la mesure			
Thème 6 : Exigences des données		Établir et supporter un engagement de mesure Planifier le processus de mesure Exécuter le processus de mesure Évaluer la mesure			
Thème 7 : Types de mesure et granularité	Fondements	Établir et supporter un engagement de mesure Planifier le processus de mesure Exécuter le processus de mesure Évaluer la mesure			

Tableau L (Suite)

Taxonomie Buglione et Abran Taxonomie Schneidewind Thèmes	Concepts de base	Processus de mesure	Normes de mesure	Mesure par phase du cycle de vie du logiciel (SLC)	Outils & Techniques
Thème 8 : Évaluation et test du processus et produit		Établir et supporter un engagement de mesure Planifier le processus de mesure Exécuter le processus de mesure Évaluer la mesure		Processus primaire - Processus du support	
Thème 9 : Prédiction de la qualité du produit et processus		Établir et supporter un engagement de mesure Planifier le processus de mesure Exécuter le processus de mesure Évaluer la mesure			Outils et techniques

9.3.2 Classification préliminaire des concepts du BOK de mesure

Comme décrit dans le chapitre 2, la classification de Vincenti [109] des connaissances de l'ingénierie est supposée être applicable à tout domaine de l'ingénierie. Le logiciel étant considérablement différent par rapport aux autres artefacts physiques, il n'y a pas eu de tentative pour l'application des principes généraux de l'ingénierie au logiciel. Cependant, il a été suggéré aux spécialistes de KA qu'ils utilisent cette classification pour leurs premiers *drafts* de chaque KA dans SWEBOK. Ceci était un défi : vers la fin des années 1990, le domaine du génie logiciel n'est pas encore assez mature et la

classification de Vincenti ne peut être directement appliquée dans la majorité des taxonomies et descriptions des KA. Cependant, il est à noter que nous nous intéressons à un sous-ensemble de l'ensemble du domaine du génie logiciel, c'est-à-dire la mise en correspondance des connaissances de mesure de logiciels aux catégories des types des connaissances de l'ingénierie. En effet, puisque le concept *design* est considéré de façon générale dans la classification de Vincenti, il peut être transposé aux éléments relatifs dans chacun des dix KA qui incluent des processus correspondants (processus des exigences, processus de design, processus de test. Chacun de ces processus nécessite un design pour retenir les généralités et les particularités de n'importe quel projet); de même, des ajustements sont requis pour d'autres thèmes tels que le thème de la mesure.

Ainsi, pour appliquer la classification de Vincenti en tant qu'un outil analytique, il est préférable d'avoir des critères pour classer chaque sous-thème de la mesure dans chaque catégorie, une liste de tels critères était identifié dans [96]. Un ajustement plus raffiné des concepts de mesure est présenté dans la colonne de droite du Tableau LI. Par exemple, dans le contexte d'une analyse spécifique de mesure, le *design* peut se référer au design d'une méthode de mesure spécifique, au design de la combinaison des différents composants de mesure dans des mesures cumulées ou au design des modèles relatifs au mesure pour l'utilisation des résultats de mesure, tels que dans les modèles de qualité, les modèles de productivité, les modèles d'estimation, etc.

Tableau LI

Classification de Vincenti des connaissances de l'ingénierie et des critères associés

#	Catégories de Vincenti	Critères dans [96]	Ajustement aux concepts de mesure
1.	Concepts fondamentaux du design (FD)	Principes opérationnels Comment ses parties de caractéristiques rencontrent leurs fonctions spécifiques en combinaison à une opération complète qui atteint l'objectif ?	Principes de mesure Ensemble des concepts de la métrologie (voir [57])

Tableau LI (suite)

#	Catégories de Vincenti	Critères dans [96]	Ajustement aux concepts de mesure
2.	Critères et spécifications (CS)	Exigences spécifiques (des principes opérationnels) Limites (à travers une technologie complète)	Caractéristiques des instruments de mesure (voir [57])
3.	Outils théoriques (OT)	Concepts liés au « design » Outil intellectuel pour réfléchir sur le « design »	Théorie de la mesure Modèle du processus de mesure
4.	Données quantitative (DQ)	Représentées dans des tableaux et des graphes	Références de mesure Données expérimentales codifiées
5.	Considérations pratiques (CP)	La théorie est souvent insuffisante – considérations à partir de l’expérience et de la pratique Compromis qui sont les résultats des connaissances générales sur le dispositif, ses utilisations et contextes	Approches expérimentales avec leurs forces et faiblesses
6.	Design de l’instrumentation (DI)	Connaissance professionnelle (<i>Knowing How</i>) – façons de penser Connaissance procédurale	Procédures de mesure Techniques d’analyse de mesure Mesure – leçons apprises

Dans cette analyse, la classification de Vincenti [109] est utilisée pour reconnaître et identifier les types des connaissances de l’ingénierie impliqués dans le domaine de connaissance de la mesure du logiciel. Le Tableau LII présente une mise en correspondance des thèmes du BOK de mesure de [27] selon la classification de Vincenti. Les « X » indiquent ainsi la couverture des éléments du BOK par rapport à cette classification. Ceci nous permet d’identifier les forces et faiblesses de mesure de logiciels en tant qu’un concept et outil fondamental de l’ingénierie.

Tableau LII

Classification préliminaire des concepts du BOK de la mesure [27] et ceux de Vincenti [109]

BoK Buglione et Abran [27] \ Vincenti [109]	Concepts fondamentaux de design	Critères et spécifications	Outils théoriques	Données quantitatives	Considérations pratiques	Design de l'instrument
Concepts de base :						
Fondements	X					
Définitions & concepts	X					
Modèles de mesure du logiciel			X			
Entités à mesurer		X	X			
Processus de mesure :	X					
Établir et supporter un engagement de mesure	X				X	
Planifier le processus de mesure	X				X	
Exécuter le processus de mesure	X				X	
Évaluer la mesure	X				X	
Normes de mesure		X				
Par entité		X				X
Par type		X				X
Mesure par phase SLC :		X				
Processus primaire		X				
Processus du support		X				
Outils & techniques :						
Outils					X	X
Techniques					X	X

Le Tableau LIII ci-après présente la combinaison des thèmes et des sous-thèmes du BOK de la mesure dans chaque catégorie de la classification des connaissances de l'ingénierie.

Tableau LIII

Sommaire préliminaire des thèmes de mesure de Vincenti et BOK

#	Catégories de Vincenti	Thèmes –Corpus de connaissance de mesure	Sous-thèmes –Corpus de connaissance de mesure
1.	Concepts fondamentaux de design (FD)	Concepts de base	Fondements Définitions & concepts Norme ISO sur le processus de mesure (15939)
2.	Critères et spécification (CS)	Concepts de base	Entités à mesurer
		Mesure par phase du SLC	Processus primaire Processus organisationnel et de support
3.	Outils théoriques (TT)	Concepts de base	Design des mesures de logiciels Entités à mesurer Modèles pour l'utilisation des résultats de mesure
4.	Données quantitatives (QD)		(N'est pas inclus directement dans SWEBOK mais peut être inclus dans les références)
5.	Considérations pratiques (PC)	Concepts de base	Modèles de <i>l'utilisation</i> de mesure du logiciel
		Processus de mesure	Établir et supporter un engagement de mesure (Directives) Planifier le processus de mesure (Directives) Accomplir le processus de mesure (Directives) Évaluer la mesure (Directives)
		Outils et techniques	Outils de mesure Techniques d'analyse de mesure
6.	Instruments de design (DI)	Concepts de base	Modèles pour l'utilisation des résultats de mesure
		Normes de mesure	Normes de mesure par entité Normes de mesure par type
		Outils & techniques	Outils de mesure Techniques d'analyse de mesure

Cette analyse a permis d'identifier le niveau auquel les recherches subséquentes de mesure de logiciels sont nécessaires. Les améliorations requises d'une perspective de l'ingénierie peuvent être identifiées :

- le corpus de connaissance pour les mesures de logiciels proposé par Buglione *et al.*[27] couvre seulement cinq des six catégories de Vincenti [109] à l'exception

des données quantitatives. Dans les autres disciplines de l'ingénierie, les méthodes et les instruments de mesure établis doivent rencontrer les concepts du VIM [57], incluant les critères de qualité des résultats de mesure ainsi que l'utilisation des références de mesure;

- les méthodes de mesure ainsi que les instruments de mesure et les modèles validés expérimentalement pour l'évaluation et l'estimation ne sont pas encore matures dans le domaine du génie logiciel. Ce qui est requis est une définition précise des mesures de logiciels, et qui devraient être basées sur des approches vérifiables.

L'application des concepts de la métrologie aux mesures de logiciels peut contribuer à l'amélioration des méthodes de mesure et des instruments de mesure reliés à l'application de ces méthodes ainsi qu'à l'amélioration du domaine de connaissance de la mesure de logiciels.

9.3.3 Mise en correspondance de la mesure dans SWEBOK avec le cadre de référence

En utilisant l'inventaire détaillé des thèmes de mesure apparaissant dans les dix chapitres de SWEBOK, lesquels thèmes ont été analysés en termes de concepts de mesure, il est possible maintenant de les mettre en correspondance par rapport à l'ensemble des concepts du cadre de référence de vérification d'une mesure (Tableau LIV). Ce tableau montre la couverture des concepts de la métrologie, indiquée par un « X » pour chacun des dix chapitres de SWEBOK. La plupart des concepts de mesure mentionnés dans SWEBOK concernent l'utilisation des résultats de mesure. En effet, les chapitres du BOK tels que « exigences de l'ingénierie du logiciel », « test de l'ingénierie du logiciel », « maintenance de l'ingénierie du logiciel » et « gestion de configuration du logiciel » adressent uniquement les concepts liés à l'utilisation des résultats de mesure de l'étape 4. Par contre, un seul chapitre du BOK (« qualité de l'ingénierie du logiciel ») adresse un seul aspect de fondements de mesure à travers un sous-ensemble de design

d'une mesure (étape 1), et d'autres aspects liés à l'étape 3 et 4. Quelques éléments de chapitre du BOK (« gestion de l'ingénierie du logiciel » et « processus de l'ingénierie du logiciel ») adressent l'étape 2 de l'application d'une mesure à travers la collection de la documentation et l'application des règles d'assignation numérique. Finalement, le design, la gestion et la qualité de l'ingénierie du logiciel adressent la qualité des résultats de mesure directe (avant leur utilisation dans des modèles analytiques quantitatifs, tels que ceux pour l'évaluation ou la prédiction).

Tableau LIV

Classification du thème de la mesure dans SWEBOK avec le cadre de référence

Cadre de référence		Étape 1	Étape 2	Étape 3	Étape 4
SWEBOK	BOK	Design	Application	Résultats	Utilisation
Thème					
Exigences de l'ingénierie du logiciel	Support et gestion du processus				X
	Négociation des exigences				X
	Qualité des documents				X
	Tests d'acceptation				X
	Trace des exigences				X
Design de l'ingénierie du logiciel	Mesures			X	
Test de l'ingénierie du logiciel	Évaluation des programmes sous test				X
	Évaluation des tests accomplis				X
Maintenance de l'ingénierie du logiciel	Mesure de la maintenance du logiciel				X
Gestion de configuration du logiciel	Surveillance de la gestion de configuration du logiciel				X

Tableau LIV (Suite)

Cadre de référence		Étape 1	Étape 2	Étape 3	Étape 4
SWEBOK BOK (2001)	Thème	Design	Application	Résultats	Utilisation
Gestion de l'ingénierie du logiciel	Objectifs				X
	Sélection de mesure				X
	Mesure du logiciel et de son développement				X
	Collection des données		X		
	Modèles de mesure de logiciels			X	
Processus de l'ingénierie du logiciel	Méthodologie dans la mesure du processus		X		
	Paradigmes de la mesure du processus				X
Qualité de l'ingénierie du logiciel	Mesurer la valeur de la qualité				X
	Fondements de mesure	X			
	Mesures			X	
	Techniques d'analyse de mesure				X
	Caractérisation des défauts				X
	Utilisation additionnelle des données de SQA et V&V				X

Cette analyse illustre les raisons pour lesquelles les éditeurs et réviseurs du guide SWEBOK n'ont pu accepter la proposition d'un chapitre séparé sur les mesures parce que les connaissances sur les concepts de mesure en génie logiciel ne rencontraient pas les critères du « généralement accepté » tels que définis par le *Project Management Institute* – PMI. En effet, en génie logiciel les fondements de mesure sont encore relativement faibles dans la plupart des thèmes de mesure : par conséquent, il n'y a pas encore de consensus sur leur valeur, leur efficacité et leur force de généralisation en dehors du contexte initial de leur définition.

Une nouvelle structure du BOK de la mesure du logiciel est maintenant proposée dans la Figure 44, incluant une branche additionnelle sur les données quantitatives.

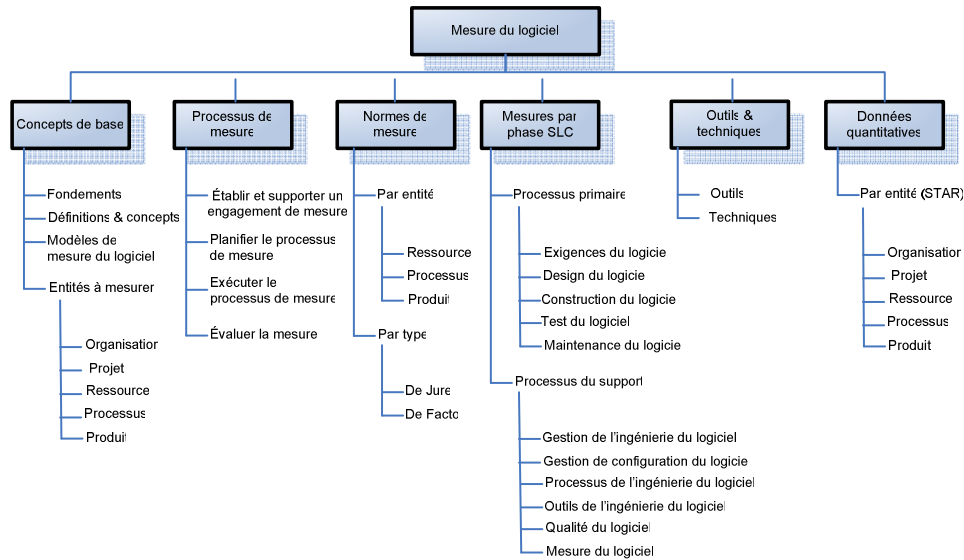


Figure 44 Nouvelle taxonomie pour un corpus de connaissance de mesure du logiciel [2]

Cette nouvelle taxonomie est subdivisée en six branches en utilisant la taxonomie d'entité STAR (*Software Taxonomy Revised*) [26] où les critères du « généralement accepté » pour les références des données quantitatives devraient être choisis pour renforcer les méthodes de validation expérimentales. En particulier par les catégories « observationnelles », à cause de leurs impacts directs sur les projets à faible coût, par exemple *Project monitoring and field studies* selon la classification de Zelkowitz & Wallace [112].

Quelques exemples : dans la section « processus », un type de références possibles à inclure pourrait être les rapports biannuels CBA-IPI/SCAMPI par le *Software Engineering Institute*¹³, tandis que pour la section « produit », il pourrait être possible

¹³ Disponible dans la page Web SEMA : <http://www.sei.cmu.edu/cmml/appraisals/appraisals.html#appraisal-results>

d'inclure des études sur les expériences de mesures MTF validées par le groupe COSMIC.

9.4 Sommaire

L'intérêt pour le domaine de la mesure de logiciels s'accroît au sein de la communauté du génie logiciel et beaucoup des connaissances reliées à ce domaine ont été incluses dans le guide SWEBOK. Bien que le thème de la mesure soit réparti à travers les 10 KA composant le corpus de connaissances SWEBOK, il n'a pas été reconnu initialement comme un KA à part.

Pour vérifier et examiner si la mesure de logiciels est une discipline assez mature de l'ingénierie, elle doit avoir une place spécifique dans les prochaines révisions du guide SWEBOK. Une première proposition pour un nouveau domaine de connaissances sur la mesure a été réalisée par Buglione *et al.* [27]. Dans le but d'améliorer cette proposition, nous avons réalisé ces étapes :

- une comparaison par rapport à une étude antérieure sur un BOK pour la mesure de la qualité du logiciel de Schneidewind [101] a révélé la pertinence des normes internationales comme une base pour couvrir le critère « généralement accepté » et mettre à jour l'analyse de la proposition initiale pour un nouveau KA de mesure de logiciels dans SWEBOK;
- l'utilisation de la classification de Vincenti [109] des types de connaissance de l'ingénierie comme un outil analytique pour analyser la proposition initiale. Après la classification des concepts du BOK de la mesure, nous avons remarqué que tous les critères ont été couverts, à l'exception des données quantitatives;
- l'utilisation du cadre de référence comme un outil d'analyse du thème de la mesure dans SWEBOK affirme : (i) le manque de consensus sur la mesure de logiciels, (ii) le manque des données quantitatives valides et largement reconnues pour supporter la qualité des mesures de point de vue de l'ingénierie, et (iii) le manque des méthodes de mesure de logiciels avec assez de force métrologiques

comme instruments de mesure et la correspondance avec les critères de qualité des résultats de mesure dans le domaine de la métrologie ainsi que l'utilisation des références de mesure.

Cette analyse a montré également qu'il y avait un manque des références reconnues des autres concepts de la métrologie VIM. En résumé, la mesure en génie logiciel est loin d'être mature d'une perspective de métrologie, et elle représente une faiblesse des fondements de l'ingénierie pour la discipline du génie logiciel.

9.5 Recommandations

Après avoir réalisé cette analyse et vérification de différentes perspectives, une nouvelle taxonomie révisée et améliorée a été proposée, avec le critère de sélection des références « généralement acceptées » pour une nouvelle branche de « données quantitatives ».

Les mesures de logiciels ne sont pas matures dans leur design selon les concepts du cadre de référence, et ainsi elles sont insuffisantes pour supporter une expérimentation adéquate en génie logiciel. De plus, les étalons de mesure devraient être requis pour supporter le génie logiciel comme une discipline *bona fide* de l'ingénierie.

CHAPITRE 10

PROPOSITION D'UN CADRE INTÉGRÉ DE VÉRIFICATION D'UNE MESURE

10.1 Introduction

L'objectif de notre proposition d'un cadre intégré de vérification est d'établir les critères pour un processus de vérification des méthodes de mesure en génie logiciel. Il est nécessaire de segmenter le concept de mesure pour mettre en relief les différentes étapes impliquées dans ce processus.

10.2 Cadre intégré de vérification de mesure

Le cadre intégré de vérification de mesure vise à identifier un inventaire plus exhaustif des critères de qualité pour les mesures proposées en génie logiciel. Ce cadre intègre principalement trois approches avec chacune des perspectives différentes tout en étant complémentaire (Tableau LV).

Tableau LV

Cadre intégré de vérification de mesure

Étape 1 Design de la méthode de mesure	Étape 2 Application de la méthode de mesure	Étape 3 Vérification des résultats de mesure	Étape 4 Utilisation des résultats de mesure
Définition du contexte de design d'une mesure Identification des concepts d'intrant au processus de mesurage	Collection des artéfacts du logiciel Construction d'un modèle pour l'entité logicielle à mesurer Application des règles d'assignation numérique	Vérification des résultats obtenus de mesure selon les critères du VIM	Utilisation des résultats de mesure dans des modèles d'analyse

Tableau LV (Suite)

Étape 1 Design de la méthode de mesure	Étape 2 Application de la méthode de mesure	Étape 3 Résultats de mesure	Étape 4 Utilisation des résultats de mesure
Modélisation du monde empirique Définition du monde numérique	Utilisation de l'instrument de mesure : utilisation de l'étalon Identification des caractéristiques de l'instrument de mesure		

Les sections suivantes vont maintenant présenter un aperçu de chacun des éléments de vérification dans les trois premières étapes de mesure. La quatrième étape portant exclusivement sur l'utilisation des résultats dans des modèles d'analyse, ce thème relève plus des méthodes expérimentales que de la mesure directement et ne sera donc pas inclus dans le cadre de vérification proposé.

10.3 Étape 1 : Design d'une méthode de mesure

10.3.1 Définition du contexte de design d'une mesure

La définition du contexte de design d'une méthode de mesure est requise pour identifier les objectifs et pour décrire la structure d'une mesure incluant les concepts de mesure et leurs relations ainsi que pour déterminer les conditions de mesurage :

- existence des instruments de mesure;
- possibilité d'expérimentation;
- existence d'étalons;
- existence d'un ensemble représentatif de données historiques, etc.

10.3.2 Identification des concepts d'intrant au processus de mesurage

10.3.2.1 Grandeurs et unités

Pour appliquer le concept de « grandeur (mesurable) » au génie logiciel, il doit y avoir un attribut spécifique et distinct à mesurer. Autrement dit, on doit spécifier la propriété d'un attribut de l'entité logiciel (exemple : propriété « taille fonctionnelle » d'un logiciel) et auquel une expression quantitative peut être assignée. Cependant, certaines grandeurs ne peuvent être clairement identifiées (exemple : une mesure de la complexité qui dépend des opinions et des intuitions des mesureurs); il est alors difficile d'appliquer les concepts de traçabilité et d'étalon. Les descriptions purement qualitatives des attributs (nommées « mesures subjectives » dans ISO 25021 [70]) sont généralement ambiguës. Pour être sujet au mesurage, un attribut doit être susceptible d'être distingué qualitativement et déterminé quantitativement. Ainsi, un attribut à mesurer d'une entité d'un logiciel doit être tout d'abord bien défini pour en avoir une caractérisation non ambiguë. D'autre part, certaines grandeurs peuvent être classées les unes par rapport aux autres selon un ordre croissant ou décroissant : ce sont les grandeurs de même nature telles que par exemple toutes les grandeurs de longueur, incluant l'épaisseur, la circonférence, la longueur d'onde, etc. Ce classement permet de simplifier plusieurs relations à partir d'un traitement quantitatif.

La détermination des grandeurs de même nature d'un logiciel n'est pas encore claire. De plus, dans un système de grandeurs, l'identification des grandeurs qui sont fonctionnellement indépendantes les unes des autres (grandeurs de base) et des grandeurs qui dépendent des grandeurs de base (grandeurs dérivées) est nécessaire, car généralement les grandeurs de base ayant la même unité sont additives alors que les grandeurs dérivées ne sont pas toujours additives [28] (exemple : quand deux quantités égales d'eau sont combinées à une température donnée, la quantité résultante d'eau n'aura pas une température qui est égale à la somme des températures des différentes quantités).

Une fois que l'attribut à mesurer a été bien défini, il est possible d'assigner des nombres et d'obtenir une valeur.

10.3.2.2 Valeurs des grandeurs de l'intrant

L'assignation des nombres à une grandeur n'est pas suffisante, il faut que les nombres soient significatifs pour permettre une communication claire, précise et sans ambiguïté. Pour être significatifs, il faut des références, des unités de mesure et des échelles spécifiant les multiples et sous-multiples d'unités. Cependant, il y a la possibilité que certaines grandeurs ne soient pas exprimées sous forme d'une unité multipliée par un nombre. Dans ce cas, elles peuvent être exprimées par référence à une échelle de repérage et/ou à un mode opératoire. Sinon, il est nécessaire d'avoir un échantillon représentatif pour obtenir des valeurs significatives des grandeurs de l'intrant [102].

10.3.2.3 Références de mesure : design de l'étalon

Le mesurage constitue la comparaison d'un objet à la réalisation physique de l'unité. Cette réalisation est obtenue par un étalon primaire. Dans le cas où il y a absence d'un étalon, des études de cas bien documentées peuvent être reconnues comme un substitut de référence. Définir le design d'un étalon de mesure du logiciel est un objectif ambitieux qui nécessite un consensus de la communauté scientifique et de recherche. Dans ce travail, nous avons considéré certains concepts nécessaires pour le design d'un étalon, en spécifiant le type d'étalons et ses propriétés, son mode de conservation et de matérialisation (voir tableau XXVI).

10.3.2.4 Valeurs des grandeurs de l'extrant

Comment déterminer la valeur de l'extrant en fonction des valeurs de l'intrant? Selon Kitchenham *et al.* [82], la mesure d'un attribut se fait par l'application d'une unité de mesure spécifique à une entité particulière pour obtenir une valeur, généralement une valeur numérique. La vérification d'une telle valeur requiert la définition d'un ensemble

approprié de valeurs permmissibles, qui peut être fini ou infini, discret ou continue, etc. En cas d'estimation, la valeur de l'intrant n'est pas connue mais peut être déduite à partir de la valeur de l'extrant selon une fonction mathématique de mesure et en utilisant un étalon de référence.

10.3.3 Modélisation du monde empirique

10.3.3.1 Définition du méta-modèle de l'attribut

Construire un méta-modèle consiste à décrire l'ensemble des caractéristiques sélectionnées pour représenter le logiciel, et fournir les règles de modélisation du logiciel permettant l'identification des entités utilisées lors de cette modélisation. Exemple : plusieurs attributs d'un produit logiciel peuvent être définis à travers un modèle conceptuel correspondant au produit logiciel en utilisant la modélisation conceptuelle tels que les diagrammes UML [99]. Il y a aussi, par exemple, l'ensemble des règles décrites pour mettre en relief l'attribut de taille fonctionnelle selon le méta-modèle de COSMIC-FFP [69].

10.3.3.2 Caractérisation de l'attribut à mesurer

Une fois que le méta-modèle a été défini et les règles de modélisation du logiciel ont été choisies, une définition opérationnelle de l'attribut mesuré doit être construite. L'opération de caractérisation de l'attribut peut être vue comme étant la définition d'un ensemble empirique relationnel d'un point de vue mathématique.

10.3.4 Définition du monde numérique (mathématique)

10.3.4.1 Définition des échelles de mesure

Selon Kitchenham *et al.* une mesure transforme un attribut empirique dans un monde mathématique [82]. Cet attribut peut être mesuré en utilisant différentes unités. L'unité

de mesure permet de comprendre les différents types d'échelle. Les échelles les plus utilisées en logiciel sont de type nominal, ordinal, intervalle, ratio et absolue.

Il y a aussi d'autres types d'échelle qui peuvent être définis tels que l'échelle logarithmique (terme d'origine grec qui signifie « mesure des rapports »). Cette échelle est une alternative à l'échelle linéaire. Elle dilate les valeurs faibles et rapproche les valeurs fortes.

10.3.4.2 Définition des règles d'assignation numérique

Cette partie définit les règles à utiliser pour assigner des valeurs numériques aux attributs d'une entité empirique donnée. Ces règles permettent de définir un ensemble relationnel numérique d'un point de vue mathématique (c'est-à-dire, de trouver l'ensemble des valeurs d'un attribut associé à une entité empirique donnée). Par exemple, pour un comptage, les règles déterminent comment distinguer les entités à compter, comment les compter, etc. Pour un mesurage à réaliser à partir d'autres valeurs, donc des mesures dérivées, les règles déterminent les équations, l'incertitude de mesure, etc. Pour un mesurage réalisé par l'intermédiaire d'un instrument de mesure, les règles doivent tenir compte des caractéristiques de l'instrument, l'étalonnage, etc.

10.4 Étape 2 : Application d'une méthode de mesure

Cette étape consiste à mettre en correspondance les artefacts du logiciel à mesurer avec les règles de la méthode de mesure pour faciliter le mesurage. Le déroulement de mesurage peut être manuel ou automatique. Dans tous les cas, il est nécessaire de passer par les étapes suivantes :

Collection des artefacts du logiciel : c'est une étape importante dans le domaine du logiciel du fait que ce qu'on mesure n'est pas l'entité physique de logiciel mais plutôt une abstraction telle qu'un produit logiciel ou un processus logiciel. Par exemple, les

documents de description des « cas d'utilisation » permettent de collecter l'information nécessaire pour la mesure de la taille du logiciel.

Construction d'un modèle pour le logiciel à mesurer : sur la base des documents collectés, le logiciel en question est modélisé en utilisant des règles et les attributs à mesurer. Cette modélisation n'implique pas uniquement l'application des règles mais également les choix de modélisation. Par exemple, la modélisation des exigences du logiciel en tant que collection de « processus fonctionnels » selon la méthode COSMIC-FFP [69] peut nécessiter différents choix pour reconnaître ce que devrait être un « processus fonctionnel ».

Application des règles d'assignation numérique : cette étape consiste à appliquer les règles définies dans l'étape précédente sur une instance de l'entité modélisée.

10.4.1 Utilisation de l'instrument de mesure

L'utilisation d'un instrument de mesure permet la réalisation des mesurages et ainsi de fournir une valeur mesurée d'un attribut dans une unité particulière. Plusieurs opérations sont possibles lors de l'utilisation d'un instrument de mesure, telles que le calibrage, l'ajustage et le réglage. D'autre part, l'étalonnage devrait permettre d'améliorer la précision d'un instrument de mesure. Le processus d'étalonnage permet de s'assurer que les valeurs de la grandeur indiquées par l'instrument de mesure correspondent aux valeurs de la grandeur réalisées par des étalons. Ainsi l'utilisation d'un instrument de mesure nécessite l'utilisation de l'étalon.

10.4.2 Identification des caractéristiques de l'instrument de mesure

Les caractéristiques de l'instrument de mesure sont classées en caractéristiques d'utilisation et caractéristiques de contrôle (les détails de ces caractéristiques sont décrits dans le Tableau XIV du chapitre 4).

10.5 Étape 3 : Vérification des résultats de mesure

Comment exprimer un résultat obtenu de mesure? L'application d'une méthode de mesure à une entité possédant une grandeur donnée fournit un résultat, qui est la valeur mesurée. Une des caractéristiques d'une valeur mesurée doit être sa traçabilité. Pour appliquer la traçabilité au mesurage des produits logiciels, il est nécessaire de mettre en place un système de mesure cohérent.

La traçabilité est définie par la propriété du résultat d'un mesurage (ou d'un étalon) tel qu'il puisse être relié à des références déterminées, généralement des étalons nationaux ou internationaux, par l'intermédiaire d'une chaîne ininterrompue de comparaisons ayant toutes des incertitudes déterminées. En se basant sur la définition de la traçabilité dans le VIM, il est nécessaire d'évaluer l'incertitude de mesure. Cependant, en génie logiciel, il est rarement question de l'incertitude d'un mesurage, et lorsqu'il en est question, c'est souvent déterminé par jugement d'experts plutôt que par l'analyse d'un mesurage.

L'étalonnage signifie qu'il y a une relation entre les valeurs d'une grandeur indiquées par un instrument de mesure et les valeurs correspondantes de la grandeur réalisée par des étalons, qui sont généralement des étalons primaires. La relation est appelée traçabilité. L'étalonnage et la traçabilité assurent que toutes les mesures sont réalisées avec les mêmes unités de mesure, à un niveau approprié d'incertitude. Ainsi, les résultats peuvent être comparativement robustes en tout temps et lieu. Or en génie logiciel, il a été rarement question à date de ce concept d'étalonnage qui est pourtant fondamental en métrologie.

D'autre part, les méthodes statistiques de traitement de l'exactitude et de répétabilité sont rarement utilisées en mesure du logiciel. Un résultat de mesure peut être une estimation de la valeur de l'extrait d'une grandeur associée nécessairement avec l'incertitude de mesure.

Quelles sont les sources de l'incertitude?

- expérimentale (erreur de mesure);
- fondamentale (définition de la grandeur à mesurer elle-même ainsi que le type d'échelle utilisé);
- notion de la mesure elle-même et de l'exactitude de mesure (concepts probabilistes ouvrant une large part au hasard et à l'imprévisible). Il faut enfin admettre que le hasard joue un rôle dans l'obtention des résultats de mesure.

10.6 Identification des critères du cadre intégré de vérification d'une mesure

Tout d'abord, il faut clarifier les termes vérification et étalonnage. L'étalonnage est l'opération qui établit la relation entre les valeurs de la grandeur indiquées par un étalon et les indications correspondantes d'un système de mesure, réalisées sous des conditions spécifiées et incluant l'évaluation de l'incertitude de mesure (VIM'04).

L'étalonnage est l'opération établissant la relation, obtenue par référence à un ou plusieurs étalons, qui existe sous des conditions spécifiées, entre l'indication d'un système de mesure et le résultat de mesure qui devrait être obtenu en utilisant le système de mesure selon la définition d'étalonnage dans (IEC 60050-300, item 311-01-09) [51]. Tandis que la vérification est la confirmation à travers l'examen d'un item donné et la preuve d'une évidence objective qu'elle satisfait les exigences spécifiées.

L'élaboration d'un cadre intégré de vérification de mesure est requise pour comprendre, structurer, comparer et analyser les différentes approches de mesure proposées en génie logiciel, ainsi que pour contrôler les fondements de mesures que les praticiens appliquent dans l'industrie du logiciel.

En se basant sur l'analyse, la caractérisation du cadre de vérification passe par trois étapes : vérification du design de la méthode, vérification de l'application d'une méthode de mesure, et vérification des résultats obtenus.

10.6.1 Vérification du design de la méthode de mesure

La première étape du cadre intégré proposé consiste à constituer une équipe de vérification pour vérifier le design de la méthode :

- vérifier la présence (et la qualité) d'une définition des objectifs de la méthode de mesure et des attributs à mesurer;
- vérifier la définition des concepts d'intrant au processus de mesurage : identification des grandeurs et unités, vérification de la signification des grandeurs de l'intrant, identification des références de mesure et l'étalon, et vérification des valeurs des grandeurs de l'extrait;
- vérifier la modélisation du monde empirique : vérification empirique, il faut s'assurer que la modélisation empirique représente effectivement le concept à modéliser, et que le méta-modèle et caractérisation de l'attribut à mesurer sont identifiés;
- vérifier la définition du monde numérique : vérification théorique, s'assurer la mise en correspondance entre le monde empirique et le monde numérique (mathématique) avec l'utilisation d'une échelle adéquate pour vérifier si la mesure capture exactement l'attribut qui est supposé être mesuré.

10.6.2 Vérification de l'application de la méthode de mesure

La deuxième étape du cadre intégré proposé consiste à assembler les intrants de vérification à l'application de la méthode de mesure :

- vérifier les documents collectés en se basant sur les critères de la norme IEEE-830 [53] (correct, non ambigu, complet, consistant, rangé pour l'importance et/ou la stabilité, vérifiable, modifiable et traçable);
- vérifier le méta-modèle construit;
- vérifier les règles d'assignation numérique appliquée;
- vérifier le mesurage manuel ou à l'aide d'un instrument de mesure (vérification expérimentale);

- vérifier les caractéristiques de contrôle et d'utilisation des instruments de mesure.

10.6.3 Vérification des résultats obtenus

La troisième étape consiste à vérifier les résultats obtenus :

- vérifier la répétabilité (par exemple, si deux personnes utilisent la même méthode dans les mêmes conditions de répétabilité, arrivent-elles au même résultat?), la reproductibilité (si l'on fait varier les conditions de mesure, arrive-t-on au même résultat de mesure?), et l'exactitude des résultats de mesure;
- vérifier les sources de l'incertitude;
- identifier et documenter le potentiel d'incertitude.

10.7 Conclusion

Nous avons développé dans ce chapitre un cadre intégré de vérification et de contrôle qui va de la mise au point d'une méthode de mesure jusqu'à l'obtention des résultats qu'elle peut produire, et qui a pour but de retenir les meilleures explications des faits connus dans la littérature de mesure en génie et génie logiciel pour déterminer les critères pertinents de vérification.

CHAPITRE 11

CONCLUSION

11.1 Bilan du travail

En génie logiciel, le domaine de la mesure a été abordé en autres dans un contexte d'amélioration des produits et des processus logiciel [5;29;42;114]. La plupart de ces mesures ont été conçues en se basant sur les intuitions des chercheurs et/ou sur des fondements empiriques, et sont caractérisées par la facilité du comptage de certaines entités du processus de développement (mesures de la complexité, mesures orientées objet).

Dans leur analyse de ces entités, la plupart des chercheurs ont souvent utilisé les concepts de la « théorie de la mesure » comme une base pour leurs recherches analytiques. En se basant sur les concepts de cette « théorie de la mesure », les chercheurs en mesure du logiciel ont traité principalement les conditions de représentation, les propriétés mathématiques de manipulation des nombres et les conditions propres de telles manipulations. Toutefois, la théorie de la mesure, telle que référencée par ces auteurs, porte seulement sur un sous-ensemble de l'ensemble des concepts classiques de mesure.

Notre revue de la littérature sur les mesures de logiciels n'a pas permis d'identifier parmi ces travaux des références portant sur les concepts fondamentaux de la métrologie, et ce dans toutes les recherches dans la qualité des méthodes de mesures proposées dans cette littérature du génie logiciel.

De plus, il est difficile d'analyser la qualité de ces méthodes de mesures à cause d'un manque de consensus sur un cadre de vérification de mesure. Pour cette raison, nous avons développé un cadre intégré de vérification des méthodes de mesure, intégrant

différentes perspectives, et qui pourra être utilisé non seulement pour analyser les mesures de logiciels mais pour toutes les mesures proposées.

Au cours de notre recherche, nous avons développé un cadre de référence de vérification des méthodes de mesure. Un tel cadre nous a permis de comprendre comment vérifier une mesure en-soi (au niveau de son design), comment évaluer les différentes mesures proposées en génie logiciel et comment vérifier le design des instruments de mesure reliés à l'application pratique d'une méthode de mesure donnée.

L'utilisation de ce cadre de référence nous a permis de développer un cadre intégré de vérification de mesure. Le développement de ce cadre intégré a passé par trois volets : le premier volet théorique a consisté à identifier les outils analytiques de mesure et les critères (concepts) qui pourraient venir de la métrologie VIM pour vérifier les propositions de mesure en génie logiciel. Le second volet plus pratique a consisté à illustrer l'application pratique des concepts de mesure dans l'analyse des normes ISO sur les mesures de logiciels. Le troisième volet intégrateur a consisté à développer un cadre intégré de mesure. Un sommaire de ces trois volets est maintenant présenté.

Volet théorique : Dans cette thèse, nous avons commencé par identifier les principaux concepts des mesures des grandeurs physiques du VIM de telle façon qu'ils puissent être introduits dans les mesures de logiciels.

La modélisation des concepts du VIM nous a permis de comprendre convenablement les connaissances de la métrologie. Elle nous a permis également de rendre plus visibles et compréhensibles les concepts du VIM et de préciser les niveaux d'abstraction et de relations entre ces concepts. En effet, nous avons proposé 15 modèles représentant les topologies de haut niveau et détaillées des 120 termes du VIM. Ainsi, un modèle de haut niveau de la classification du VIM a été développé, suivi par des modèles détaillés illustrant comment les termes individuels sont inclus dans ce modèle de haut niveau. Par la suite, nous avons développé 24 modèles additionnels en utilisant d'autres explications

provenant soit des « Note » et « Exemples » du VIM, soit à partir de la littérature de métrologie. Ces modèles nous ont permis de clarifier l'ensemble des termes associés avec les différentes catégories du VIM et de préciser les concepts de mesure aussi variés et complexes que leurs relations.

Après avoir modélisé les concepts de mesure du VIM, nous avons proposé un cadre de référence de vérification pour les méthodes de mesure basé sur les concepts du VIM et du processus de mesure d'Abram *et al.* Ce cadre inclut les forces des deux modèles.

Volet pratique : Nous avons illustré l'utilisation pratique de ce cadre initial de référence en nous référant aux concepts de mesure à la fois en génie logiciel et dans les autres domaines de l'ingénierie. En effet, les mesures de logiciels devraient être consistantes avec les mesures dans les autres disciplines de l'ingénierie. Nous avons appliqué notre proposition initiale d'un cadre de référence à plusieurs mesures utilisées dans le domaine du logiciel. Ceci nous a montré que certaines mesures sont bien vérifiées dans des contextes bien définis alors que d'autres mesures ne rencontrent pas les critères de vérification.

Lors de l'analyse de ISO TR 9126-2 à 4, nous avons observé que la plupart des mesures sont décrites par une seule formule algébrique, sans référence à des méthodes de mesure, et encore moins à des méthodes qui permettent d'obtenir des résultats de mesure qui répondent aux critères du VIM. La plupart des catégories de mesure dans cette série de documents ISO sont toutes documentées en utilisant la même structure de documentation. Nous avons choisi à titre illustratif les mesures de la productivité et l'efficacité de la caractéristique « fonctionnalité » de l'ISO TR 9126-4. Nous avons remarqué que la dimension d'une mesure n'est pas spécifiée : il y a confusion entre mesure de base et mesure dérivée. Cela veut dire que la structure des éléments de mesure dans les rapports 9126-2 à 4 n'est assez complète. Par contre, les forces de ces rapports sont : la description informelle par le positionnement d'une mesure à l'intérieur de la structure caractéristique/sous-caractéristiques d'un des niveaux de mesure (interne,

externe et en utilisation), les formules de « comptage » et des unités implicites ainsi que l'identification des données selon les références ISO 12207 et spécifie une formule algébrique pour un type de mesure donné et définit un méta-modèle des caractéristiques de la qualité pour 9126-1.

L'absence des éléments de design d'une mesure tels les grandeurs et unités, le méta-modèle, les méthodes de mesure, les étalons, la réalisation des références et la traçabilité par une chaîne d'étalonnage montre que l'application de telle mesure conduit à des résultats erronés. Nous recommandons d'introduire tous ces concepts pour obtenir un résultat de qualité en termes métrologiques. ISO 9126-2 à 4 doivent être décrites d'une façon plus précise comme les mesures de la taille fonctionnelle pour être reconnues comme des normes.

Au cours de l'analyse de ISO 14143-1 à 5, nous avons constaté que cette série traite d'un seul type de mesure MTF, et elle le traite de façon très détaillée et plus en profondeur que dans la série ISO 9126-2 à 4 sur la mesure de la qualité du logiciel. Cette série de documents ISO 14143-2 à 5 peut être utile dans l'évaluation d'une méthode de mesure particulière selon les critères de « qualité » des méthodes de mesure. Cependant, elle rencontre seulement un sous-ensemble de l'ensemble des caractéristiques des résultats de mesure selon le VIM (Tableau XI). Il est suggéré d'introduire les autres concepts du VIM particulièrement ceux reliés aux instruments de mesure et aux différents types d'étalon.

Le cadre de référence de mesure a donc été utile pour analyser différents types de mesure : mesures des attributs de qualité dans ISO 9126-2 à 4 et mesures de la taille fonctionnelle dans ISO 14143-2 à 5. Il nous a permis d'identifier des problèmes de mesure à résoudre. Les « métriques » pour mesurer les attributs de mesure de la qualité dans 9126-2 à 4 ne sont donc pas complètement définis. Les mesures MTF peuvent être enrichies par les concepts du VIM pour atteindre la précision autant que les autres mesures des sciences physiques et de l'ingénierie.

La revue de la littérature montre également qu'il n'y a pas d'autres types de mesure, qui ont été aussi bien décrites et documentées que dans le document ISO 14143-1 à 5 et en utilisant autant de concepts pertinents à la métrologie. Ce document pourrait donc être utilisé comme exemple pour définir et documenter d'autres types de mesure du logiciel.

L'utilisation du modèle du VIM nous a permis d'analyser non seulement la méthode de mesure COSMIC-FFP : ISO 19761 mais aussi un prototype d'automatisation dans l'environnement RUP-Rational Rose pour découvrir si oui ou non cette mesure rencontre beaucoup des concepts du VIM. Cette analyse nous a montré que le design de COSMIC-FFP couvre la plupart des concepts du VIM relatifs au design des méthodes de mesure; d'autre part, des études de cas à plus grande échelle seraient requises pour l'étude des caractéristiques des instruments de mesure telles qu'identifiées dans le VIM. En résumé, le système de grandeurs ne comprend que la seule grandeur de base définie par la taille en Cfsu. Il n'y a qu'une seule unité Cfsu. Le système d'unités ne comprend ni unités dérivées ni unités hors système. Les sous-multiples d'unités ne sont pas encore formellement définis. De même, il n'y a qu'un seul niveau de granularité. Nous recommandons de déterminer ces éléments notamment les sous-multiples d'unités, pour que cette méthode devient plus précise.

Il en est de même pour l'utilité d'un étalon pour la mesure COSMIC-FFP, lequel étalon est en cours de design et de réalisation dans un autre projet de doctorat. D'autre part pour obtenir des résultats statistiquement significatifs sur les concepts des résultats de mesure du prototype COSMIC/RUP, il est nécessaire de mettre en place de plus larges études de cas. De plus, au niveau des concepts des caractéristiques des instruments de mesure, il est requis de mettre l'accent sur l'expérimentation et de développer des instruments de mesure qui permettent d'explorer tous ces concepts de façon plus adéquate et exhaustive.

Le cas pratique illustre comment le modèle du VIM pourrait être utilisé non seulement pour analyser les autres méthodes MTF, mais également pour toutes les autres mesures

de logiciels proposées à l'industrie. Nous suggérons donc d'utiliser l'ensemble des concepts de métrologie documentés dans le VIM comme des critères pour analyser les forces et les faiblesses des autres méthodes MTF ainsi que toute autre mesure du logiciel.

L'analyse de ISO 15939 a montré qu'il y a des faiblesses au niveau des processus de mesure pour les activités de gestion de logiciels. Les éléments liés au design et aux résultats de mesure ne sont pas couverts. L'introduction des concepts du VIM pourra permettre la normalisation des indicateurs de qualité, et pour de construire des modèles d'estimation basés sur des fondements vérifiables. L'analyse de COCOMO a révélé que les données en input de modèle ne sont pas vérifiables.

D'une façon générale, l'utilisation du cadre de référence nous a permis d'identifier les forces et les faiblesses des normes ISO de mesure de logiciels. Nous avons pu remarquer que la majorité des documents ISO sur la mesure du logiciel ne sont pas encore des normes. Ces divers documents ISO (i.e. rapports techniques) sont basés sur des approches qui n'ont pas été encore rigoureusement démontrées. Plusieurs de ces documents ISO sont basées sur des exigences « subjectives ». Les organisations de ces normes devront réviser leurs objectifs et réviser les documents/ rapports techniques qui adressent des objectifs d'une façon beaucoup plus objective. La plupart des documents ISO ont été développés en isolation par le sous-comité du génie logiciel. Ces documents ne font pas référence au corpus des connaissances en métrologie.

Plus généralement, on trouve qu'on doit apprendre beaucoup des normes des autres disciplines de l'ingénierie. Les normes de mesure de logiciels manquent des critères solides d'évaluation.

L'introduction des concepts fondamentaux de métrologie au domaine de la mesure du logiciel a permis d'apporter de nouvelles possibilités d'interprétation de mesure dans le domaine du génie logiciel et même de contribuer éventuellement à l'évolution et à la

maturation des mesures de logiciels. En particulier, certains des modèles détaillés qui ont été proposés dans notre travail de recherche peuvent suggérer comment de tels concepts de métrologie peuvent être utilisés pour concevoir de nouveaux instruments de mesure ainsi que la vérification (et la validation) de l'ensemble courant des mesures de logiciels proposés. Le modèle du VIM est applicable tant pour les mesures en général que pour les mesures du logiciel en particulier.

En conclusion, la majorité des caractéristiques métrologiques ne sont pas abordées dans ISO 9126, ISO 14143 et ISO 15939. Le domaine de connaissance de mesure de VIM qui devrait être considéré lors de design des mesures n'est pas totalement couvert. Les designs des mesures en génie logiciel n'ont pas rencontré ce domaine et par conséquent leur exploitation passe par des difficultés de validation.

En se basant sur les résultats des analyses mentionnés dans le volet pratique, il semble nécessaire de développer un cadre intégré comme une base pour unifier les différentes perspectives et vérifier toute proposition de mesure. Les divers volets de ce modèle intégré sont maintenant présentés.

Volet intégrateur : L'analyse du cadre de vérification proposé par Kitchenham *et al.* [82] a révélé qu'il y a une ambiguïté dans le domaine de la mesure en génie logiciel. En effet, dans [82] il n'y a pas de distinction entre les concepts et les sous-concepts de mesure (i.e. les étapes du cadre de référence) auxquels s'appliquent les critères de vérification proposés. Par exemple, la référence entre le monde réel et le monde mathématique pour un seul attribut n'est pas assez précise dans [82] et ne spécifie pas tous les concepts requis pour la vérification de design d'une méthode de mesure (étape 1 du cadre de référence). Nous avons remarqué que les concepts reliés à l'entité de mesure et ses attributs sont clairement énoncés ; par contre les règles d'assignation numérique sont partiellement adressées en termes d'unités, types d'échelle et valeurs, et le méta-modèle n'est pas abordé. Nous avons remarqué aussi que Kitchenham *et al.* [82] n'expliquent pas la relation entre attributs des entités distinctes de mesure. En effet, les

éléments présentés dans le texte de [82] comme critères pour la vérification ne se réfèrent pas aux concepts liés aux « mesures indirectes » (association empirique entre plusieurs attributs qui est formalisée par une équation mathématique), mais plutôt à des concepts liés à la définition des ratios, indicateurs ou formules pour décrire la combinaison d'attributs. D'autre part, dans [82] il n'est pas reconnu explicitement que des références de mesure doivent être établies, et que lors d'une instanciation de mesure ces références ne doivent pas changer. De plus, l'exemple donné se réfère à des nombres appartenant à l'échelle de type ratio, sans aucune référence à l'étalon « échelle ». Nous avons proposé ainsi d'introduire d'autres critères additionnels au modèle de [82] qui se basent sur la métrologie, tels que les références de mesure pour les conditions de représentation, incluant les références liées à l'étalon de mesure, et que le modèle de définition de l'unité doit appartenir au design d'une méthode de mesure plutôt qu'à une instanciation spécifique.

De même, certains critères (les mêmes critères) dans [82] peuvent se référer non pas à un seul concept mais à plusieurs concepts distincts, exemple : « différence entre les modèles de définition » qui se réfère à la fois au design d'une méthode de mesure et à son application. Par conséquent, nous avons proposé de préciser l'utilisation de la terminologie de mesure pour éviter toute ambiguïté, de clarifier les entités types à vérifier et de comprendre les propriétés correspondantes qui doivent être adressées dans un cadre de vérification d'une mesure. Il est à noter également que l'expression « système de relations empiriques » telle que décrite dans [82] se réfère non pas à un seul concept mais à deux concepts distincts : condition de représentation d'un seul attribut, et relation entre plusieurs attributs.

Nous avons remarqué également que l'analyse effectuée par Kitchenham *et al.* se base sur le domaine des études expérimentales plutôt que sur le domaine de connaissance de la métrologie ; l'accent est mis sur l'exploitation des résultats de mesure comme une source de connaissance d'un monde empirique. La plupart des mesures proposées n'ont

pas été conçues à travers un cycle de design ou à travers une vérification de données appropriées, adéquates et exhaustives, que ce soit à un haut niveau ou à un niveau détaillé de vérification des méthodes de mesure de logiciels. Nous recommandons d'utiliser les concepts de métrologie pour apporter des informations plus précises sur plusieurs aspects de mesure notamment sur les instruments de mesure et les caractéristiques de ces instruments.

L'analyse de SWEBOK a confirmé qu'en génie logiciel, les fondements de mesure sont encore relativement faibles dans la plupart des thèmes de mesure :

- la comparaison entre le corpus de connaissance de Schneidewind et celui de Buglione *et al.* nous a confirmé l'importance des normes;
- l'alignement du corpus de connaissance de Buglione *et al.* [27] avec la classification de Vincenti [109] a montré le manque de données quantitatives du point de vue de l'ingénierie;
- l'utilisation du cadre de référence nous a montré l'importance des méthodes et instruments de mesure qui doivent rencontrer les concepts du VIM [57], incluant les critères de qualité des résultats de mesure ainsi que l'utilisation des références de mesure. Et, que les méthodes de mesure ainsi que les instruments de mesure et les modèles validés expérimentalement pour l'évaluation et l'estimation ne sont pas encore matures dans le domaine du génie logiciel. Ce qui est requis, c'est une définition précise des mesures de logiciels, et qui devraient être basées sur des approches vérifiables.

Cette analyse a permis d'identifier le niveau auquel les recherches subséquentes de mesure de logiciels sont nécessaires. Les améliorations requises d'une perspective de l'ingénierie peuvent être identifiées : il y a un manque des méthodes de mesure de logiciels avec assez de force comme les instruments de mesure et la correspondance avec les critères de qualité des résultats de mesure dans le domaine de la métrologie. Il y a aussi un manque des données quantitatives valides et largement reconnues pour

supporter encore la qualité prévue de point de vue de l'ingénierie pour les thèmes décrits du génie logiciel. Cette analyse indique également un manque de références aux autres concepts de la métrologie VIM. En résumé, la mesure en génie logiciel est loin d'être mature d'une perspective de métrologie, et elle représente une faiblesse dans les fondements de l'ingénierie pour la discipline du génie logiciel.

En se basant sur le cadre de référence, les outils analytiques et les analyses des normes ISO, nous avons développé un cadre intégré de mesure pour la comparaison, l'évaluation et la définition des mesures en génie logiciel. Ce cadre converge avec les connaissances existantes de mesure et avec ce que les autres chercheurs ont trouvé sur la vérification des mesures de logiciels (telle que le cadre de Kitchenham *et al.*, et celui d'Abran *et al.*), mais il se veut plus exhaustif et intègre la perspective de métrologie aux mesures de logiciels existantes. Ce cadre identifie également des critères explicites de vérification des mesures pour comprendre, structurer, comparer et analyser les différentes approches de mesure présentées en génie logiciel.

Positionnement par rapport à d'autres travaux de recherche (*Related Work*) : la plupart des autres écoles de pensées (travaux précédents) sur la vérification des mesures (Kitchenham *et al.* [82], Lopez *et al.* [83], Schneidewind [100], Zuse [115], etc.) se basent sur la formulation des mesures de logiciels en tant qu'évaluation. Le cadre que nous avons proposé, constitue le prolongement de ces travaux en introduisant un autre aspect important de mesure qui est l'aspect métrologique. Ce cadre présente ainsi une nouvelle perspective qui a un impact significatif tant pour l'industrie que pour la communauté de recherche de logiciels.

Pour l'industrie, nos résultats de recherche sont pertinents pour les responsables des mesures de logiciels et les groupes d'amélioration des processus lorsqu'ils ont à sélectionner des mesures et à juger de leurs forces et faiblesses, ou encore pour concevoir de nouvelles mesures pour les besoins de leurs organisations.

Pour la communauté de recherche en génie logiciel, l'impact est sur les chercheurs qui ont à structurer le corpus de connaissance des mesures en génie logiciel et à analyser les modèles existants de mesure, et pour les scientifiques qui désirent suivre les méthodes de recherches basées sur des études de cas et des méthodologies de recherche.

L'originalité de cette thèse porte sur l'application de la métrologie au domaine de vérification des mesures de logiciels. Ceci produit une nouvelle approche qui inclut tous les éléments pertinents de vérification d'une mesure allant du design d'une méthode de mesure jusqu'à l'utilisation des résultats qu'elle peut fournir. Cette nouvelle approche converge avec ce que Dubé [35] et Tichy [108] ont proposé dans leurs recherches sur les études empiriques. En particulier, dans son article « *Hints for Reviewing Empirical Work in Software Engineering* » sur la compréhension et la révision des études empiriques en génie logiciel, Tichy [108] a mentionné que les concepts liés uniquement aux statistiques (théorie de la mesure) ne sont pas représentatives des études empiriques. En effet, le jugement d'un travail empirique par des standards des tests statistiques conduit à l'élimination de plusieurs résultats intéressants. La répétabilité dans les résultats obtenus est une nécessité.

Les résultats que nous avons trouvé concordent avec les connaissances existantes du domaine de la métrologie et permettent de comprendre l'étendue entre la théorie et la pratique dans les recherches empiriques actuelles en mesure en génie logiciel.

11.2 Contribution dans d'autres travaux de recherche

Au cours de notre recherche nous avons fait une analyse des mesures en génie logiciel. Nous pensons qu'une telle analyse a permis d'identifier plus clairement certaines lacunes du génie logiciel.

À partir des résultats de nos travaux de recherche, nous avons donc proposé des rapports techniques (*white papers*) pour sensibiliser le comité industriel représentant le

Canada auprès du sous-comité ISO en génie logiciel, et afin d'arrimer au domaine du logiciel tant les modèles que les concepts de la métrologie pour les prochaines révisions des normes en mesure en génie logiciel. Ainsi, nous recommandons que les normes en génie logiciel soient révisées et améliorées.

Le cadre que nous avons proposé est général, une fois que la mesure du logiciel atteigne le niveau de maturité de la même façon que dans les autres disciplines du génie, il est convenable de mettre en place des cadres spécialisés pour différents objectifs de mesure. Il est important de définir clairement l'attribut à mesurer et l'entité correspondante pour concevoir correctement les mesures de logiciels.

Une faiblesse de plusieurs méthodes de mesure de logiciels est la subjectivité dans l'identification des entités. Ceci survient lorsque les règles d'assignation numérique sont ambiguës ou lorsque les règles d'identification des composants des méta-modèles sont ambiguës.

En génie logiciel, les concepts de mesure manquent de précision et de clarté et il y a un manque d'un consensus détaillé sur les caractéristiques des concepts à mesurer. L'application des concepts de la métrologie aux mesures de logiciels peut contribuer à l'amélioration des méthodes de mesure et les instruments de mesure reliés ainsi qu'à l'amélioration du domaine de connaissance de la mesure de logiciels.

11.3 Perspectives d'avenir

Cette thèse nous a permis de comprendre le processus des méthodes de mesure en science et en génie logiciel, notre travail pourrait donner naissance à plusieurs autres projets de recherche sur les thèmes de la mesure et des principes fondamentaux du génie logiciel. En effet, les résultats obtenus des analyses pourraient être approfondis.

Le cadre intégré que nous avons proposé pourrait servir comme un document de base pour le développement d'une méta-norme de vérification de mesure. Étant donné que ce cadre fournit les détails sur les différentes étapes de vérification, incluant les leçons apprises, les études de cas, etc., il fournit les critères de vérification de mesures en utilisant principalement la terminologie du VIM. Ces critères peuvent devenir des exigences normalisées pour évaluer les caractéristiques métrologiques d'une méthode ou d'un instrument de mesure d'une façon uniforme et claire. Les différentes perspectives ont été coordonnées et intégrées pour fournir un cadre consistant de vérification.

Pour obtenir des résultats de mesure répondant aux caractéristiques métrologiques, on doit utiliser des méthodes appropriées ou encore des instruments de mesure appropriés. Les instruments de mesure en particulier permettent d'élargir nos connaissances empiriques et ainsi contribuer à de nouveau progrès de la discipline du génie logiciel. Mais les instruments eux-mêmes ont des limites (erreurs et incertitudes de mesure). La réalisation des expériences avec la formulation des hypothèses pourront permettre de remédier à ces limites.

Ceci pourra permettre de suggérer de nouvelles idées, comme par exemple le développement d'un étalon spécifique à un type de mesure logiciel, ce qui pourra ouvrir la voie à de nouvelles recherches :

- produire des étalons de vérification en déterminant la valeur de l'erreur maximale tolérée, le maximum de l'erreur de répétabilité, etc.
- développer un cadre de vérification et validation plus détaillé : une approche plus robuste devrait connaître l'importance des échelles de mesure et tenir compte de l'avantage de construire une hiérarchie de classification. Il est important de reconnaître que les unités de mesure et les échelles de mesure sont étroitement liées;

- quand les systèmes d'unités deviennent matures, et quand ils sont à appliquer, la notion d'un système de grandeurs devient importante car elle met en relief les conversions entre unités;
- validation des principes fondamentaux du génie logiciel en utilisant par exemple la classification de Vincenti pour déterminer clairement la place de la mesure en génie logiciel;
- appliquer le cadre intégré pour résoudre les problèmes de mesure orientées-objet particulièrement les mesures de couplage.

Nous espérons que cette recherche sur le processus de vérification des mesures de logiciels aura contribué à mieux comprendre les problèmes de mesure en génie logiciel. Nous espérons également qu'elle permettra d'améliorer le niveau de maturité du génie logiciel, si les recherches subséquentes identifiées sont effectuées.

ANNEXE A

Liste des publications

Résumé des publications :

Abran, A. et A. Sellami (2002). “Initial Modeling of the Measurement Concepts in the ISO Vocabulary of Terms in Metrology”, présenté aux Software Measurement and Estimation - Proceedings of the 12th International Workshop on Software Measurement - IWSM, Magdeburg (Allemagne).

Abstract:

The field of software metrics is usually discussed from the perspective referred to as “measurement theory”. However, in other disciplines, the domain of knowledge referred to as “metrology” is the foundation for the development and use of measurement instruments and measurement processes. This paper presents an initial modelling of the sets of measurement concepts documented in the ISO International Vocabulary of Basic and General Terms in Metrology. In particular, this modelling illustrates both the different levels of abstraction of the concepts as well as the relationships across related concepts and sub-concepts. We refer to this representation type as the topology of the concepts within this ISO Vocabulary. These models will provide the basis for analysing the current status of the field of “software metrics” and to suggest improvements along the classic path of the field of metrology.

Abran, A. et A. Sellami (2002). “Measurement and Metrology Requirements for Empirical Studies in Software Engineering”, présenté au *IEEE Software Technology and Engineering Practice Workshop (STEP)*, Montréal.

Abstract:

“Software metrics” are most often proposed as the measurement tools of choice in empirical studies in software engineering, and the field of “software metrics” is most often discussed from the perspective referred to as “measurement theory”. However, in

other disciplines, it is the domain of knowledge referred to as “metrology” that is the foundation for the development and use of measuring instruments and measurement processes. In this paper, our initial modeling of the sets of measurement concepts documented in the ISO International Vocabulary of Basic and General Terms in Metrology is used to investigate and position the measurement concepts referred to in the Guide to the Software Engineering Body of Knowledge. This structured analysis reveals that much work remains to be done to introduce the full set of measurement and metrology concepts as fundamental tools for empirical studies in software engineering.

Abran, A., A. Sellami et W. Suryn (2003). “Metrology, Measurement and Metrics in Software Engineering” présenté au Software Metrics Symposium. Proceedings of the Ninth International Software Metrics Symposium (METRICS'03) IEEE, Sydney (Australie).

Abstract:

Up until recently software “metrics” have been most often proposed as the quantitative tools of choice in software engineering, and the analysis of these had been most often discussed from the perspective referred to as “measurement theory”. However, in other disciplines, it is the domain of knowledge referred to as “metrology” that is the foundation for the development and use of measurement instruments and measurement processes.

The IEEE-Computer Society, with the support of a consortium of industrial sponsors, has recently published a Guide to the Software Engineering Body of Knowledge (SWEBOK) and, throughout this guide; measurement is pervasive as a fundamental engineering tool. In this paper, we use our initial modelling of the sets of measurement concepts documented in the ISO International Vocabulary of Basic and General Terms in Metrology to investigate and position the measurement concepts within this body of

knowledge, and to identify gaps where further research on software measurement is required.

Sellami, A. et A. Abran (2003). « The Contribution of Metrology Concepts to Understanding and Clarifying a Proposed Framework for Software Measurement Validation », présenté à l'Investigations in Software Measurement - Proceedings of the 13th International Workshop on Software Measurement - IWSM, Montréal.

Abstract:

In the software engineering literature, numerous practitioners and researchers have proposed hundreds of “software measures”, or “software metrics”. To help industry assess the quality of these proposed measures, researchers have proposed various approaches to software measurement validation, none of which has yet been widely used by either designers or users of software measures. To tackle this diversity of validation approaches, Kitchenham *et al.* had proposed a framework for software measurement validation and suggested a critical review of their proposed framework. This paper performs such a review using a key ISO document on measurement, that is, the ISO Vocabulary on Metrology, as well as a measurement process model derived from an analysis of the individual validation proposals. The metrology concepts in particular have facilitated greater understanding of the set of measurement sub-concepts that must be included in each of the steps from the design of a measurement method to the use of the measurement results.

Abran, A. et A. Sellami (2004). « Analysis of Software Measures Using Metrology Concepts - ISO 19761 Case Study », présenté au Software Audit and Metrics, Proceedings of the 1st International Workshop on Software Audit and Metrics - SAM 2004 -in conjunction with ICEIS 2004, Porto (Portugal)

Abstract:

To help identify the strengths of proposed software measurement methods, this paper proposes an analytical approach based on metrology concepts documented in the ISO International Vocabulary of Basic and General Terms in Metrology. This approach is illustrated with a case study using one specific functional size measurement method recognized as an ISO standard: COSMIC-FFP (ISO 19761). The case study documents the metrology concepts addressed in this ISO standard, either in the design of this measurement method or in some of its practical uses. It illustrates, for instance, that the design of COSMIC-FFP encompasses a large number of related metrology concepts. It is suggested that such a review using metrology criteria be used to analyze other software functional size measurement methods, as well as other software measures suggested to industry.

Abran, A., L. Buglione et A. Sellami (2004). « Software Measurement Body of Knowledge - Initial Validation Using Vincenti's Classification of Engineering Knowledge Types», présenté au Software Measurement - Research and Application - Proceedings of the International Workshop on Software Metrics and DASMA - IWSM/Metrikon 2004, Königs Wusterhausen (Allemagne).

Abstract:

The “Guide to the SWEBOK” (2001 Trial version) currently contains ten distinct software engineering Knowledge Areas (KA) and three common themes: Quality, Tools and Measurement. The Measurement topic is pervasive throughout all the KAs (in both the 2001 and 2004 editions). An initial taxonomy for a new specific KA on Software Measurement had been proposed in 2003. To improve this initial proposal, the Vincenti’s classification of engineering knowledge types was used as an analytical tool to study the initial proposal breakdown. This paper presents a revised breakdown for a KA on Software Measurement.

Bourque, P., S. Wolff, R. Dupuis, A. Sellami et A. Abran (2004). « Lack of Consensus on Measurement in Software Engineering: Investigation of Related Issues », présenté au Software Measurement - Research and Application - Proceedings of the International Workshop on Software Metrics and DASMA - IWSM/ Metrikon 2004, Königs Wusterhausen (Allemagne).

Abstract:

Even though measurement is considered an essential concept in recognized engineering disciplines, the use of measures in software engineering is still far from being widely used and would certainly not qualify as generally accepted. To figure out why software measurement has not yet gained enough peer recognition, this paper presents a set of issues that still have to be addressed adequately by the software measurement community. These issues were derived from the analysis of comments obtained during two Delphi-studies and a web-based survey conducted to identify and reach a consensus on the fundamental principles of the discipline within the software engineering international community. The paper also discusses the application of metrology concepts as a research direction to address some of the measurement issues identified.

Sellami, A. et A. Abran (2005). « Analyse de COSMIC-FFP selon la perspective de la métrologie », Génie Électrique et Informatique -GEI'2005, p. 19-25.

Résumé :

Dans le but d'aider à identifier les forces des méthodes proposées pour mesurer la taille fonctionnelle du logiciel, cet article présente une approche analytique basée sur les concepts de la métrologie. Cette approche est illustrée par une étude de cas en utilisant la norme ISO 19761 : COSMIC-FFP pour la mesure du logiciel. Cette étude de cas

documente les concepts de la métrologie qui sont traités dans cette norme, tant le design de la méthode de mesure que ses utilisations pratiques. Le résultat de cette analyse montre entre autres que le design de la méthode COSMIC-FFP rencontre la plupart des concepts de la métrologie.

Abran, A. et A. Sellami (2005). « Modelling the Measurement Concepts in the ISO Vocabulary of Metrology Terms » À soumettre dans *Journal of NIST*, 2005

Abstract:

In the scientific fields, including engineering, as well as in other fields such as business administration and a significant number of the social sciences, measurement is one of a number of analytical tools. In these fields, it is based on a large body of knowledge, often built up over centuries or even millennia, commonly referred to as the field of “metrology”.

The ISO document that constitutes the official national and international consensus on the metrology terminology is the ISO “Vocabulary of Basic and General Terms on Metrology” – VIM [12]. The VIM follows some of the concepts of traditional presentations of vocabularies, with terms defined individually using textual information, in a single level of classification. This mode of representation is as challenging in terms of assembling the full set of specialized and interrelated terms, as it is in understanding and remembering such a large number of them, all with their own precise and distinct meaning. To improve both the presentation and the understanding of this complex set of interrelated concepts which, up to now, has been referred to as “metrology”, the authors have used the software engineering technique of “modeling” to facilitate the presentation of so many concepts and the relationships between them.

This paper presents a modelling of the sets of 120 measurement definitions documented in the VIM, as well as those additional measurement concepts embedded within these definitions, but not explicitly defined themselves. This modelling illustrates the various levels of abstraction of the concepts, as well as the relationships that exist across related concepts and sub-concepts. We refer to this representation type as the topology of the concepts within the VIM.

Abran, A. et A. Sellami. « Analyse d'une mesure de la taille fonctionnelle du logiciel selon la perspective de métrologie : étude du cas COSMIC-FFP », accepté dans la revue du génie logiciel (Le Magazine de l'Ingénierie du Logiciel et des Systèmes –N° 74 Septembre 2005)

Résumé :

Dans le but d'aider à identifier les forces des méthodes proposées pour mesurer la taille fonctionnelle du logiciel, cet article présente une approche analytique basée sur les concepts de la métrologie. Cette approche est illustrée par une étude de cas en utilisant la norme ISO 19761 : COSMIC-FFP pour la mesure du logiciel. Cette étude de cas documente les concepts de la métrologie qui sont traités dans cette norme, tant le design de la méthode de mesure que ses utilisations pratiques. Le résultat de cette analyse montre entre autres que le design de la méthode COSMIC-FFP rencontre la plupart des concepts de la métrologie. Enfin, il est suggéré que cette approche soit utilisée non seulement pour analyser les autres méthodes de mesure de la taille fonctionnelle, mais également pour toutes les autres mesures du logiciel proposées à l'industrie.

Budgen, D., G. Hoffnagle, M. Müller, F. Robert, A. Sellami et S. Tilley (2002). "Empirical Software Engineering: A Roadmap", présenté à la Software Technology and Engineering Practice, IEEE Proceedings. 10th International Workshop, Montréal.

Abstract:

The growing awareness of the need to employ evidence based arguments to support the practices of software engineering, rather than arguments based upon advocacy, has led to a growing interest in the related area of empirical practices. This workshop, held at STEP 2002, set out to address two questions. What techniques do we need for Empirical Software Engineering? What are the “grand challenges” that Empirical Software Engineering should be addressing as a matter of priority?

For practical reasons, these were addressed in reverse order, and we began by identifying what each participant saw as being the challenges facing the wider acceptance and use of empirical techniques in software engineering.

We then classified and grouped the challenges so identified under the three headings of resource, technical, and strategic. The next step involved examining how the particular challenges could be addressed, what proposals for action we would put forward, and how these proposals might be followed up. A core element we identified was the need to provide some form of centralised resource (such as a web site) that could support both research and teaching. In this paper we outline the role that such a site could have, and also identify some of the elements that address this topic.

ANNEXE B

Travaux ISO en cours sur la métrologie

VIM Préliminaire 2004-2005

Métrologie en général

Le VIM constitue un consensus international sur la terminologie commune et cohérente de l'ensemble des concepts de la métrologie. La 3^e édition (2005) du VIM [74] présentera une nouvelle approche pour les concepts d'incertitude, qui est basée sur le concept d'incertitude décrit dans GUM [58]. Il est à noter que ce document n'a pas reçu encore une autorisation finale d'ISO, et que son contenu n'est donc pas encore définitif. Ce document pourrait donc ne pas être publié en 2005, mais plutôt dans les années subséquentes. Ce concept caractérise l'étendue au quel une valeur inconnue du mesurande devient connue après le mesurage selon l'information donnée du mesurage. D'autres concepts de base relatifs à la traçabilité de mesure sont introduits dans le VIM'05.

Nous proposons de contribuer à la progression (évolution) du domaine des connaissances de la mesure (Métrologie) par une restructuration conceptuelle, qui peut fournir les éléments nécessaires pour faciliter l'introduction de la métrologie en génie logiciel. D'autre part, certaines représentations mathématiques doivent être introduites pour conserver la définition propre de certains concepts.

Bien que la 3^e édition du VIM représente une terminologie commune de l'ensemble des concepts du domaine de connaissance de la mesure, il est plutôt un document avancé. Nous proposons de présenter dans ce contexte les concepts susceptibles d'être appliqués en génie logiciel. Nous proposons également des suggestions utiles dans ce domaine.

Dans la prochaine édition du VIM, les concepts de mesure seront classés sous un ensemble de 5 catégories organisés par chapitre (Tableau LVI)

Tableau LVI

Catégories des termes en Métrologie

Categories (VIM 2004)	Number of terms = 122	Categories (VIM 1993)	Number of terms = 120
Quantities and units	26	Quantities and units	22
Measurement	43	Measurements	9
		Measurement results	16
Devices for measurement	12	Measuring instruments	31
Characteristics of measuring systems	23	Characteristics of the measuring instruments	28
Measurement standards – Etalons	18	Measurement standards – Etalons	14

Le modèle de haut niveau des catégories (VIM 2004) de Tableau LVI est présenté dans la Figure 45. Ce modèle initial correspond à notre compréhension de la topologie intégrée dans le vocabulaire du domaine de connaissance de mesure relié à la métrologie.

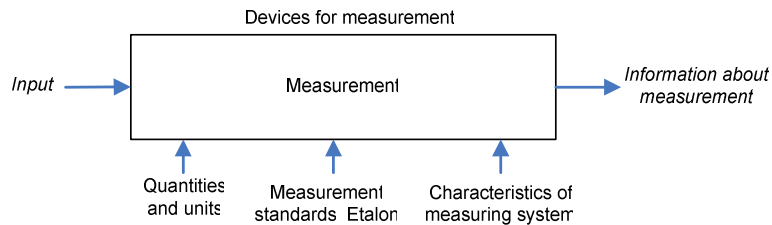


Figure 45 Modèle du processus des categories des termes VIM'2004

La Figure 45 présente les concepts de base du domaine de connaissance de la mesure (métrologie). L'objectif principal de toute mesure est d'établir une description quantitative des choses (phénomène, corps, substance,...). Autrement dit, c'est d'établir une valeur ou des tendances de certaines variables (grandeurs à mesurer). Ceci est basé sur les valeurs ou les tendances fournies par les dispositifs de mesures (*Devices for measurement*) qui doivent avoir certaines caractéristiques (*Characteristics of measuring systems*). Pour interpréter l'information obtenue, des étalons de mesure sont nécessaires.

Toutes ces considérations sont importantes et seront traitées en détail dans les sections suivantes.

Grandeurs et unités

Une grandeur est une propriété d'un phénomène, d'un corps ou d'une substance au quelle on peut attribuer une expression quantitative. Pour faire des mesures, il est nécessaire de déterminer quantitativement une grandeur; autrement dit, une grandeur est un concept nécessaire, mais pour faire des mesures, on doit attribuer des nombres à une grandeur. Cependant, ce concept peut être subdivisé en deux niveaux : général et individuel. En tant qu'un concept général, une grandeur peut appartenir ou non à un système de grandeurs et on peut classer certaines grandeurs en grandeurs de même nature. Les grandeurs acceptées par convention dans un système de grandeurs peuvent être des grandeurs de base et/ou des grandeurs dérivées. Mais il n'existe aucune équation qui relie les grandeurs de base dans un système de grandeurs (elles sont indépendantes dans leur fonctionnalité). Par contre, les grandeurs dérivées sont définies comme une fonction des grandeurs de base (elles sont dépendantes dans leur fonctionnalité). La structure d'une grandeur en tant qu'un concept individuel est basée sur le concept général de grandeur. Ce sont des concepts créés par convention qui s'ajoutent au concept général pour mieux les exprimer (l'expression mathématique d'une grandeur Q est généralement comme suit $Q = \{Q\} \cdot [Q]$ avec $\{Q\}$ la valeur et $[Q]$ l'unité). Mais une grandeur peut ne pas être exprimée comme une unité multipliée par un nombre. (Il y a aussi la possibilité qu'une grandeur soit exprimée par référence à une échelle de repérage). Elle peut être mesurée directement ou indirectement (voir section 2. *Measurement*). Ainsi, les grandeurs générales peuvent être caractérisées de différentes façons. Selon Emerson (*Metrologia, One as a 'unit' in expressing the magnitudes of quantities, 2004*), il y a quatre types de grandeurs qui peuvent être reconnues :

- grandeurs pour lesquelles chaque expression d'une valeur nécessite une référence à une procédure au quelle la grandeur est mesurée. Exemple : nombre d'octanes

pour les carburants d'un moteur. (Ça demande une échelle conventionnelle ordinale, sans manipulation algébrique).

- grandeurs pour lesquelles chaque expression d'une *magnitude* (expression quantitative) nécessite d'être comparée avec une autre, grandeur de même nature et une *magnitude* bien connue et reproductible : une unité. Exemple : masse comparée avec l'étalon kilogramme (le kilogramme est l'unité de la grandeur masse). Une valeur de ce type de grandeur est exprimée sous la forme d'une unité multipliée par un nombre. Sans l'unité, le nombre est non-significatif en tant qu'une expression quantitative de la grandeur.
- grandeurs qui sont des nombres purs (absolus), exemple toute grandeur relative, telle qu'une longueur relative ou une grandeur sans dimension. Exemple : nombre de Prandtl. Pour ce type de grandeur, les grandeurs sans dimension est un nombre sans unité mais qui est significatif.
- grandeurs, chacune d'elles est évaluée en comptant ses éléments discrets et reconnus. Exemple : la taille d'une population. Ce type de grandeur est relatif au comptage.

Le quatrième type de grandeur peut représenter un cas particulier du deuxième type (le comptage peut être considéré comme un cas simple du mesurage). Le Tableau LVII résume la hiérarchie des concepts de la catégorie « *quantities and units* » de la nouvelle version VIM'2004.

Tableau LVII

Topologie détaillée de l'ensemble des concepts « grandeurs et unités »

General concept	Quantity	System of quantities	Quantity dimension	Quantity value	Quantity calculus
	Quantities of the same kind Base quantity Derived quantity Ordinal quantity	International system of quantities, ISQ	Quantity of dimension one Base dimension Derived dimension	Conventional quantity value Numerical quantity value	Quantity equation
Individual concept	Quantity	System of quantities	Quantity dimension	Quantity value	Quantity calculus
	Measurement unit	System of units			Rules for use of units
	Base unit Derived unit Coherent derived unit Multiple of a unit Submultiple of a unit	Coherent system of units International system of units, SI			Unit equation Numerical value equation Conversion factor between units

Quels sont les représentations possibles des grandeurs? En tenant compte des définitions du VIM'2004, le concept « grandeur » peut être subdivisé en deux niveaux (Figure 46) :

- général, qui est commun à plusieurs concepts individuels, ou applicable à plusieurs concepts individuels et non singulier;
- individuel, qui est propre à un seul objet (phénomène, corps ou substance).

Ainsi, une grandeur Q exprimée comme étant « la propriété d'un phénomène, d'un corps ou d'une substance au quelle on peut lui attribuer une expression quantitative » est un concept général, tel que par exemple la longueur L , le rayon r , la longueur d'onde λ , etc. Le rayon r par exemple est général en ce qu'il représente toutes les longueurs L dans

ce qu'ils ont de commun, au delà des différences entre les objets ou entre les divers sous-groupes (grandeurs de même nature) comme la longueur d'onde λ . Ce concept est aussi individuel : il représente des propriétés communes séparément et des éléments propres tels que : le rayon d'un cercle A , $r(A)$.

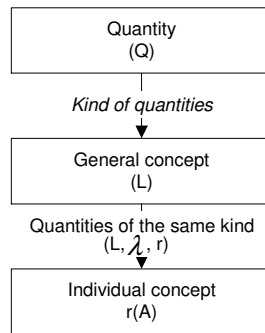


Figure 46 Niveaux des grandeurs

Nous décrivons les sous-concepts de la catégorie « grandeurs et unités » selon trois aspects : empirique, conventionnel et mathématique (Figure 47). Une grandeur représente une propriété nominale qui peut être de différentes natures, basée sur des raisons expérimentales exemple « grandeur ordinal », ou par convention et appartenant à un système de grandeurs. Dans un système de grandeurs, certaines grandeurs peuvent être subdivisées en « grandeurs de même nature ». À une grandeur, on associe une dimension et une valeur. Plusieurs sous-concepts d'une grandeur sont impliqués dans un système de grandeurs, tels que les grandeurs de base Mais d'autres concepts tels que les grandeurs dérivées peuvent ne pas être impliquées comme elles peuvent être impliquées. Mais, l'essentiel c'est qu'il y a des équations non contradictoires reliant les grandeurs (comme un concept primitif) dans un système de grandeurs. D'autre part, une unité (grandeur scalaire) est un concept spécifique d'une grandeur. Mais qui est général pour d'autres concepts tels que les multiples d'une unité et les sous-multiples d'unité, les unités de base et les unités dérivées ainsi que les unités dérivées cohérentes. Au système international de grandeurs (ISQ) est associé le système international d'unités (SI), qui

comporte toutes les caractéristiques d'un système cohérent d'unités et par conséquent, toutes les caractéristiques d'un système d'unités. Lors de l'utilisation d'un système d'unités, la notion de système de grandeurs est importante. Elle nous fournit les conversions entre les unités. Les unités dérivées sont définies en décrivant la mesure d'une grandeur, exemple : la vitesse est le ratio de la distance et l'intervalle de temps pour parcourir cette distance. L'unité de la vitesse est l'unité de la longueur par rapport à l'unité de temps.

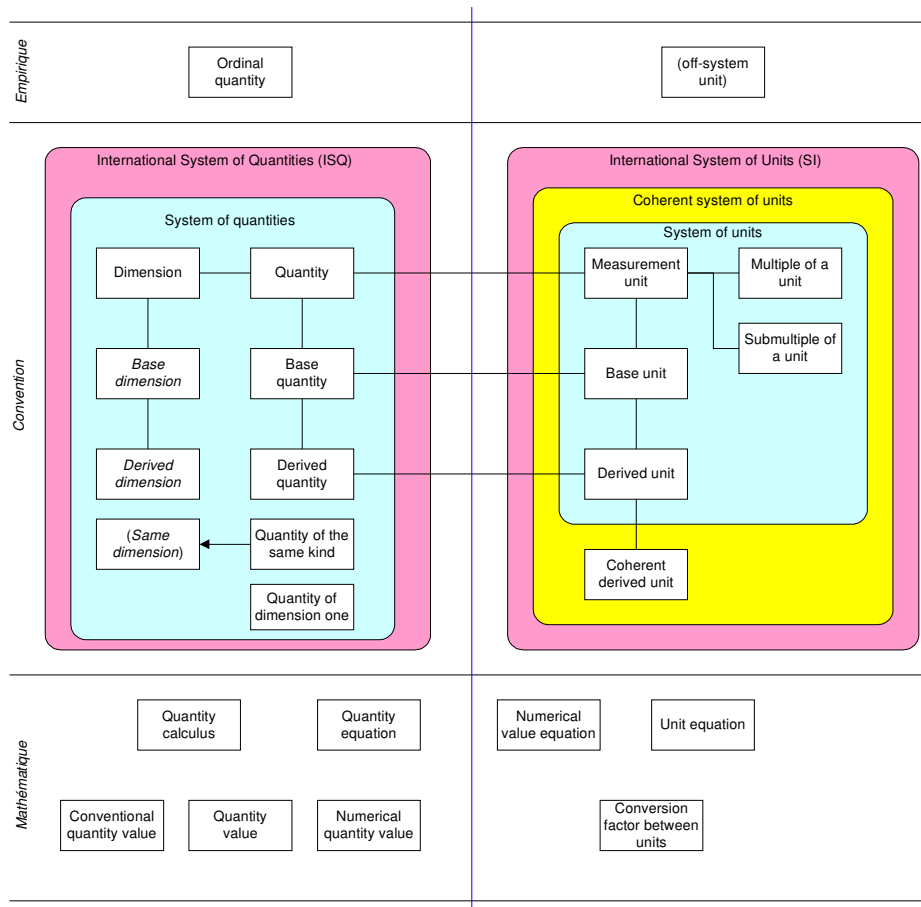


Figure 47 Modèle structurel de la catégorie « grandeurs et unités »

Manipulation algébrique (mathématique)

Quantity calculus : une grandeur doit être liée à un modèle. Une grandeur physique a deux composantes (telles que présentées dans le Tableau LVIII) : une valeur g et une unité de mesure $[Q]$. Ainsi, une grandeur Q peut être représentée comme suit : $Q = \{Q\} \cdot [Q]$. Exemple : $m = 2Kg$

Tableau LVIII

Exemples du VIM

	Name and Symbols			Equation	Example
Quantity	Q	Q_1	Q_2	Quantity equation $Q = \zeta Q_1 Q_2$ with ζ numerical factor	$T = \frac{1}{2}mv^2$
Unit	$[Q]$	$[Q_1]$	$[Q_2]$	$[Q] = [Q_1][Q_2]$ for quantity equation with $[Q], [Q_1], [Q_2]$ are coherent	$J = Kg \cdot m^2 \cdot s^{-2}$
Numerical Value	$\{Q\}$	$\{Q_1\}$	$\{Q_2\}$	$\{Q\} = \zeta \{Q_1\} \{Q_2\}$ for quantity equation with $[Q], [Q_1], [Q_2]$ are coherent	If $m = 2Kg$; $v = 3ms^{-1}$; $T = 9J$ $9 = (\frac{1}{2}) \times 2 \times 3^2$

Les opérations arithmétiques nécessitent certaines conditions; les opérations algébriques telles que l'addition et la soustraction nécessitent que les grandeurs ont la même unité. Pour la multiplication et la division, l'unité et les valeurs sont traitées séparément pour les grandeurs dérivées. Supposons par exemple la grandeur : $E = mc^2$

Selon Hall (*Software support for physical quantities, 2002*) : Si $Q = Q_1 Q_2$ et si les unités de Q sont reliées aux unités de Q_1 et Q_2 par un facteur de proportionnalité numérique α

C'est-à-dire : $[Q] = \alpha [Q_1][Q_2]$

$Q = Q_1 Q_2 = \{Q_1\} [Q_1] \cdot \{Q_2\} [Q_2]$

$\{Q\} [Q] = \{Q_1\} \{Q_2\} [Q_1][Q_2]$

Alors la valeur de Q est : $\{Q\} = \frac{1}{\alpha} \{Q_1\} \{Q_2\}$

Donc : $\alpha > 0$;

Ceci n'est applicable que pour les mesurages appartenant au type d'échelle ratio (*Conventional reference measurement scales*). Ainsi, les unités de mesure et les échelles de mesure sont étroitement liées. (La même grandeur peut être mesurée avec des unités appartenant à différents types échelles). Dans un système cohérent d'unités, le facteur α est toujours =1. Ainsi, les *Numerical value equation* ont exactement la même forme que les équations entre les grandeurs *quantity equation*. Selon (ISO 31-0 :1992), les équations entre les grandeurs ont l'avantage par rapport aux équations entre les valeurs numériques d'être indépendantes des unités.

Mesurages

Le mesurage est le processus permettant d'obtenir de l'information concernant la valeur d'une grandeur (Figure 48).

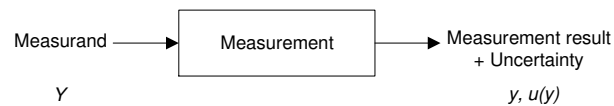


Figure 48 Topologie de haut-niveau du processus de mesure

Le mesurande est généralement noté Y et est déterminé à partir de N autres grandeurs X_1, X_2, \dots, X_N à travers une fonction de relation f tel que : $Y = f(X_1, X_2, \dots, X_N)$

Un estimé du mesurande Y noté y est obtenu à partir des estimés de l'input x_1, x_2, \dots, x_N pour les valeurs des N grandeurs de l'input X_1, X_2, \dots, X_N

Ainsi, l'estimé de l'output y qui est le résultat de mesure est donné par :

$$y = f(x_1, x_2, \dots, x_N)$$

L'incertitude d'un résultat de mesure $u(y)$ peut nécessiter l'utilisation de tous les composants de l'incertitude de mesure y compris l'incertitude standard, incertitude standard combinée, incertitude étendue, etc. Le terme incertitude implique que l'incertitude d'un résultat de mesure est un estimé et qui n'a pas une limite bien-définie. Le domaine de connaissance de la mesure (la métrologie) inclut tous les concepts théoriques et pratiques de mesurage (Tableau LIX).

Tableau LIX

Measurement Foundations Concepts

Metrology		
Knowledge Concepts	Theoretical aspect of measurement	Practical aspect of measurement + GUM
Measurement	Measurement procedure	(Measuring system)
	Measurement principle	Étalon
	Measurement method	Measurement result Measurement uncertainty
	Primary procedure	(Primary standard)
	Measurement scale	Conventional reference scale

Composants de l'incertitude de mesure

Généralement, un résultat de mesure représente une estimation de la valeur du mesurande. Ce résultat n'est complet que s'il est accompagné par son incertitude. L'incertitude de mesure est un paramètre caractérisant la dispersion des valeurs attribuées au mesurande. La dispersion est due à l'incertitude définitionnelle du mesurande et l'effet systématique et de probabilité en mesurage. Le paramètre peut être une déviation standard ou un étalon d'incertitude de mesure. L'incertitude d'un résultat de mesure consiste en plusieurs composantes. Certaines composantes peuvent être groupées en deux catégories selon la méthode utilisée (évaluation type A ou type B) pour estimer leurs valeurs numériques :

- celles qui sont évaluées par des méthodes statistiques
- celles qui sont évaluées par d'autres moyens (autres que l'analyse statistique).

L'incertitude de mesure étendue d'un résultat de mesure peut être exprimée par le facteur de couverture en donnant l'intervalle de couverture contenant la valeur du mesurande avec une probabilité élevée ou contenant un intervalle large des valeurs dispersées qui devraient être attribuées au mesurande. Le Tableau LX montre les divers concepts en relation avec le résultat de mesure selon la liste des composants de l'incertitude de mesure, la description opérationnelle des résultats de mesure et certaines conditions.

Tableau LX

Concepts en relation avec le « résultat de mesure »

Components of measurement uncertainty	Operational outline of measurement result	Conditions
Definitional uncertainty	Calibration of a measuring system	Calibration hierarchy
Type A evaluation	Metrological traceability	Traceability chain
Type B evaluation	Metrological traceability to a unit	
Standard uncertainty	Verification	
Combined standard uncertainty	Validation	
Coverage factor	Comparability	
Expanded uncertainty	Compatibility	
Coverage interval	Influence quantity	
Coverage probability	Correction	Input quantity to a measurement function
Target uncertainty	Measurement function	Repeatability condition
Expanded uncertainty= standard uncertainty * coverage factor	Precision	
	Repeatability	Intermediate precision condition
	Intermediate precision	
	Reproducibility	Reproducibility condition
	Selectivity	
	Specificity	

Évaluation vs Estimation

En mesure, il y a deux cas qui se posent; le mesurande Y peut être mesuré directement ou indirectement.

Cas où Y n'est pas mesuré directement, donc il peut être déterminé à partir de N autres grandeurs X_1, X_2, \dots, X_N à travers une fonction f

$$Y = f(X_1, X_2, \dots, X_n) \quad (1)$$

f peut être déterminée expérimentalement ou selon un algorithme qui doit être évalué numériquement. Les grandeurs d'entrée X_1, X_2, \dots, X_N dont dépend la grandeur de sortie Y peuvent être envisagées comme mesurandes ou peuvent dépendre d'autres grandeurs (tels que les effets systématiques).

Estimation

Soit y une estimation du mesurande Y , obtenu à partir de (1).

En utilisant les estimations d'entrée x_1, x_2, \dots, x_N pour les valeurs des N grandeurs X_1, X_2, \dots, X_N

$$y = f(x_1, x_2, \dots, x_N) \quad (2)$$

Lors d'un mesurage la valeur de Y n'est pas connue mais elle est déduite à partir de y (valeur de l'output) à partir d'un système de mesure. Cependant, cette valeur de y peut être influencée par plusieurs variables. Des descriptions opérationnelles expérimentales telles que par exemple les tests de contrôle sont nécessaires. Seulement dans le cas d'étalonnage la valeur Y est connue et elle nous fournit l'information pour développer la relation entre l'input et l'output qui va être utile pour d'autres mesurages subséquents pour pouvoir interpréter la valeur de l'output.

Concepts qualitatifs de mesurage

La précision est définie par « l'étroitesse de l'accord entre les valeurs obtenues par des mesurages dupliqués d'une grandeur dans des conditions spécifiques ». En fait, ce concept de « précision » englobe à la fois les concepts de répétabilité et de reproductibilité puisque la répétabilité est définie comme étant « la précision sous les

conditions de répétabilité » et la reproductibilité comme « la précision sous les conditions de reproductibilité ». Ces concepts sont qualitatifs et généralement exprimés par la déviation standard et l'incertitude standard.

Dispositifs de mesure

Les dispositifs de mesure sont utilisés pour quantifier la variable grandeur. Cette quantification consiste à assigner une valeur spécifique à cette variable qui devient la variable mesurée. La figure 5 illustre la succession des éléments d'une chaîne de mesure. En effet, la grandeur à mesurer $Y(t)$ se traduit par un signal $M(t)$ à l'entrée de la chaîne de mesure (caractérisée par une fonction de transfert $T(t)$ et fournit un signal de sortie $S(t)$, image de $Y(t)$. Une chaîne de mesure est l'ensemble des éléments d'un système de mesure constituant un seul chemin du signal de mesure (Figure 49).

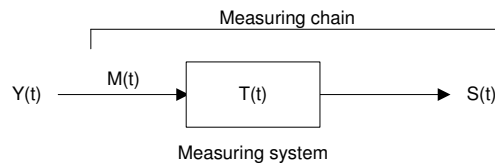


Figure 49 Éléments d'une chaîne de mesure

La Figure 50 présente une vue plus détaillée d'un système de mesure : suite à un stimulus ou un signal d'entrée, le détecteur (*detector/sensor*) détecte la présence d'un signal. S'il existe un signal, l'instrument de mesure va indiquer une valeur d'une grandeur qui lui est associé. Suite à une lecture, le transducteur de mesure fournit une grandeur de sortie ayant une relation déterminée avec la grandeur d'entrée.

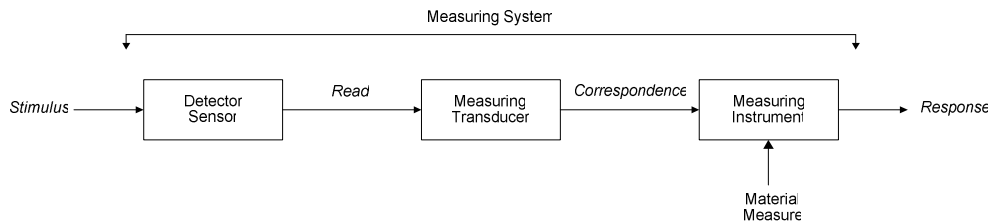


Figure 50 Système de mesure

Les dispositifs pour mesurage comportent aussi certains composants tels que : dispositifs d'affichage, échelle d'un dispositif d'affichage et indicateur d'un instrument de mesure et nécessitent des conditions opérationnelles comme l'ajustement et le zéro d'ajustement d'un système de mesure (Tableau LXI).

Tableau LXI

Composants des dispositifs pour mesurage et conditions opérationnelles des systèmes de mesure

Components of devices for measurement	Operational conditions of measuring systems
Displaying device	Adjustment of a measuring system
Scale of a displaying device	Zero adjustment of a measuring system
Indicating measuring instrument	

Caractéristiques des systèmes de mesure

Nous considérons les caractéristiques des systèmes de mesure selon deux aspects présentés dans le Tableau LXII : caractéristiques d'utilisation et caractéristiques de contrôle.

Tableau LXII

Topologie détaillée des « caractéristiques d'un système de mesure »

Characteristics of measuring Systems	
<i>Characteristics of use</i>	
<i>Specific characteristics (limits of operation)</i>	Nominal indication interval/ Nominal interval Span of a nominal interval/ Span Nominal quantity value/ Nominal value Measuring interval/ Working interval
<i>Operating characteristics (quantitative)</i>	Steady state condition for a measuring system/ Steady state condition Rated operating condition for a measuring system/ Rated operating condition Limiting condition for a measuring system/ Limiting condition Reference condition for a measuring system/ Reference condition
<i>Displaying characteristics (related to indication)</i>	Indication of a measuring system/ Indication Indication interval Resolution of a measuring system Resolution of a displaying device
<i>Dynamic characteristics (related to time)</i>	Dead band of a measuring system/ dead band Stability of a measuring system/ stability Drift of a measuring system/ drift Variation due to an influence quantity Step-change response time of a measuring system/ Step-change response time
<i>Control characteristics</i>	
<i>Qualitative characteristics (static)</i>	Optimum condition for a measuring system/ optimum condition Sensitivity of a measuring system/ sensitivity Discrimination threshold Repeatability of a measuring system/ repeatability
<i>Control and estimation characteristics</i>	Instrumental uncertainty Accuracy class

Toute prise de décision à partir d'une mesure nécessite une bonne interprétation de cette mesure et il est nécessaire de connaître en particulier l'incertitude associée, le système de référence et les étalons utilisés.

Étalons

Les étalons sont requis pour établir l'unité de mesure d'une ou plusieurs valeurs d'une grandeur à utiliser comme référence. Les sous-concepts de la catégorie « étalon » sont présentés dans le Tableau LXIII, et sont groupés en types d'étalon utilisés pour la comparaison, la vérification et/ou l'étalonnage.

Tableau LXIII
Topologie détaillée des « Étalons »

Measurement Standard/ Etalon	
<i>Type of standard (Etalon)</i>	<i>Etalon for comparison, verification and/or calibration</i>
Measurement standard/ Standard/ Etalon	Transfer measurement device/ Transfer device
National measurement standard/ National standard	Intrinsic measurement standard/ Intrinsic standard
Primary measurement standard/ Primary standard	Check measurement device/ Check device
Secondary measurement standard/ Secondary standard	Conservation of a measurement standard
Reference measurement standard/ Reference standard	Calibrator
Working measurement standard/ Working standard	Reference material/ RM
Travelling measurement standard/ Travelling standard	Certified reference material/ CRM
	Commutability of a reference material
	Reference data
	Standard reference data
	Reference quantity value/ Reference value

D'autres commentaires sur les étalons et l'étalonnage

Quand un système de mesure est étalonné, il est comparé avec un étalon, une valeur qui est supposée connue. L'étalon en question peut être une pièce d'équipement ou une propriété physique bien définie destinée à être utilisée pour un objectif de comparaison ou encore une technique acceptée pour fournir une valeur confiante. Ces étalons sont la base de toute mesure. Les étalons primaires sont nécessaires vu que la valeur attribuée à

l'unité est arbitraire [8]. Que ce soit le mètre étalon, la longueur du bras d'un roi ou la distance de la lumière parcourue en une fraction de seconde, ça dépend toujours de la façon dont on veut la définir.

D'autre part, pour éviter toute ambiguïté, les unités sont définies par consensus et par un accord international selon l'utilisation des étalons primaires. Une fois le consensus établi, l'étalon primaire établit la définition exacte de l'unité jusqu'à un changement par des consensus subséquents. Selon [8], les caractéristiques principales d'un étalon comportent :

- la disponibilité globale;
- la fiabilité continue;
- la stabilité, avec un minimum de sensibilité à des sources environnementales externes.

Pour cela, il est nécessaire d'introduire dans le design d'un système de mesure une méthode pour un étalonnage reproductible afin de simuler le type de mesurande qui devra être l'input du système lors d'un mesurage réel. Dans un processus de vérification des mesures, il est nécessaire de :

- identifier les grandeurs à mesurer;
- déterminer la relation entre l'input et l'output permettant d'atteindre l'objectif de mesure;
- identifier la méthode de mesure la plus appropriée;
- identifier le dispositif pour le mesurage;
- déterminer l'incertitude de mesure;
- s'assurer que le processus de mesure prend en considération les caractéristiques du système de mesure;
- respecter les exigences d'étalonnage en utilisant les étalons de mesure.

Dans le cas des mesures en génie logiciel, il est difficile de connaître ce que recouvre l'expression « étalons nationaux de mesure » ni comment l'interpréter : elle devrait probablement recouvrir à la fois les aptitudes en matière de mesurage des laboratoires nationaux de métrologie (et leurs incertitudes) et les mécanismes de transfert de ces compétences à d'autres laboratoires (par exemple, au moyen d'un système de référence, de protocole de mesure...). Un système national de mesure doit inclure aussi bien des étalons de mesure (qui peuvent être des systèmes de mesure ou des matériaux de référence primaires), des aptitudes en matière de mesurage (inspirés des programmes d'accréditation) et un mécanisme de publication.

Il est important d'effectuer des comparaisons dans l'établissement d'un système de traçabilité et d'équivalence des mesures (problème de conversion), de choisir une méthode permettant de déterminer les valeurs de référence et d'établir le contexte et le domaine d'applicabilité de ces comparaisons. Selon Taylor, les méthodes primaires permettent d'assurer la traçabilité d'un système de mesure :

- l'utilisation d'une méthode primaire directe permet de mesurer une valeur d'une grandeur sans référence à un étalon de la même grandeur;
- l'utilisation d'une méthode primaire de mesure de rapports permet de mesurer la valeur du rapport entre deux valeurs d'une même grandeur sans référence à un étalon de la même grandeur.

Dans les deux cas, le résultat obtenu doit être accompagné de l'incertitude.

ANNEXE C

Termes (et concepts relatifs) de métrologie documentés dans VIM'93

Nr.	Concepts dans le VIM	Notes
1	Grandeurs et unités	
1.1	grandeur (mesurable)	Grandeur dans le sens général Grandeur particulière Grandeurs de même nature Catégories de grandeurs Symboles de grandeurs
1.2	système de grandeurs	
1.3	grandeur de base	
1.4	grandeur dérivée	
1.5	dimension d'une grandeur	(ISO 31-0)
1.6	grandeur de dimension un, grandeur sans dimension	
1.7	unité (de mesure)	
1.8	symbole d'une unité (de mesure)	
1.9	système d'unités (de mesure)	
1.10	unité (de mesure) (dérivée) cohérente	
1.11	système cohérent d'unités (de mesure)	
1.12	Système international d'unités, SI	(SI fondé sur les 7 unités de base)
1.13	unité (de mesure) de base	
1.14	unité (de mesure) dérivée	
1.15	unité (de mesure) hors système	
1.16	multiple d'une unité (de mesure)	
1.17	sous-multiple d'une unité (de mesure)	
1.18	valeur (d'une grandeur)	
1.19	valeur vraie (d'une grandeur)	
1.20	valeur conventionnellement vraie (d'une grandeur)	Valeur assignée Meilleure estimation de la valeur Valeur convenue Valeur de référence (à ne pas confondre avec Note 5.7)
1.21	valeur numérique (d'une grandeur)	
1.22	échelle de repérage	
2	Mesurages	
2.1	mesurage	
2.2	métrologie	
2.3	principe de mesure	
2.4	méthode de mesure	Méthode de substitution Méthode différentielle Méthode de zéro
2.5	mode opératoire (de mesure)	
2.6	mesurande	
2.7	grandeur d'influence	
2.8	signal de mesure	Signal d'entrée: stimulus

		Signal de sortie: réponse
2.9	valeur transformée (d'un mesurande)	
3	Résultats de mesure	
3.1	résultat d'un mesurage	
3.2	indication (d'un instrument de mesure)	Indication directe
3.3	résultat brut	
3.4	résultat corrigé	
3.5	exactitude de mesure	(“Précision” à ne pas utiliser pour “exactitude”)
3.6	répétabilité (des résultats de mesurage)	Conditions de répétabilité
3.7	reproductibilité (des résultats de mesurage)	(Conditions à faire varier)
3.8	écart-type expérimental	Écart-type expérimental de la moyenne Erreur de la moyenne
3.9	incertitude de mesure	GUM, “Guide pour l’expression de l’incertitude de mesure”
3.10	erreur (de mesure)	Erreur absolue de la mesure (à ne pas confondre avec la “valeur absolue de l’erreur”)
3.11	écart	
3.12	erreur relative	
3.13	erreur aléatoire	
3.14	erreur systématique	pour un instrument de mesure voir “erreur de justesse” Note 5.25
3.15	correction	
3.16	facteur de correction	
4	Instruments de mesure	
4.1	instrument de mesure, appareil de mesure	
4.2	mesure matérialisée	Grandeur fournie
4.3	transducteur de mesure	
4.4	chaîne de mesure	
4.5	système de mesure	Installation de mesure
4.6	appareil (de mesure) afficheur, appareil (de mesure) indicateur	Indication analogique ou numérique
4.7	appareil (de mesure) enregistreur	Enregistrement (affichage) analogique ou numérique
4.8	appareil (de mesure) totalisateur	
4.9	appareil (de mesure) intégrateur	
4.10	appareil de mesure (à affichage) analogique	
4.11	appareil de mesure (à affichage) numérique	
4.12	dispositif d'affichage, dispositif	Affichage analogique

	indicateur	Affichage numérique Affichage semi-numérique
4.13	dispositif enregistreur	
4.14	capteur	(“détecteur” est utilisé pour “ capteur ”)
4.15	détecteur	Seuil de détection (“capteur” est utilisé pour “ détecteur ”)
4.16	index	
4.17	échelle (d'un appareil de mesure)	
4.18	longueur d'échelle	
4.19	étendue des indications	Étendue d'échelle
4.20	division	
4.21	longueur d'une division (d'échelle)	
4.22	échelon, valeur d'une division (d'échelle)	
4.23	échelle linéaire	Échelle régulière
4.24	échelle non-linéaire	Échelle logarithmique Échelle quadratique
4.25	échelle à zéro décalé	
4.26	échelle dilatée	
4.27	cadran	
4.28	chiffraison d'une échelle	
4.29	calibrage (d'un instrument de mesure)	
4.30	ajustage (d'un instrument de mesure)	
4.31	règlage (d'un instrument de mesure)	
5	Caractéristiques des instruments de mesure	
5.1	calibre	(Voir Note 5.2)
5.2	intervalle de mesure	Étendue
5.3	valeur nominale	
5.4	étendue de mesure	(Voir Note 5.2)
5.5	conditions assignées de fonctionnement	(Valeurs assignées)
5.6	conditions limites	
5.7	conditions de référence	Valeurs de référence Étendues de référence
5.8	constante (d'un instrument)	
5.9	caractéristique de transfert	
5.10	sensibilité	
5.11	(seuil de) mobilité	
5.12	résolution (d'un dispositif afficheur)	
5.13	zone morte	
5.14	constance	
5.15	discrétion	
5.16	dérive	
5.17	temps de réponse	
5.18	exactitude d'un instrument de mesure	

5.19	classe d'exactitude	Indice de classe
5.20	erreur (d'indication) d'un instrument de mesure	
5.21	erreurs maximales tolérées (d'un instrument de mesure), limites d'erreur tolérées (d'un instrument de mesure)	
5.22	erreur au point de contrôle (d'un instrument de mesure)	
5.23	erreur à zéro (d'un instrument de mesure)	
5.24	erreur intrinsèque (d'un instrument de mesure)	
5.25	erreur de justesse (d'un instrument de mesure)	
5.26	justesse (d'un instrument de mesure)	
5.27	fidélité (d'un instrument de mesure)	
5.28	erreur réduite conventionnelle (d'un instrument de mesure)	Valeur conventionnelle
6	Étalons	
6.1	étalon	Étalon collectif Série d'étalons
6.2	étalon international	
6.3	étalon national	
6.4	étalon primaire	
6.5	étalon secondaire	
6.6	étalon de référence	
6.7	étalon de travail	Étalon de contrôle
6.8	étalon de transfert	Dispositif de transfert
6.9	étalon voyageur	
6.10	traçabilité	Chaîne de raccordement aux étalons Chaîne d'étalonnage
6.11	étalonnage	Certificat d'étalonnage Rapport d'étalonnage
6.12	conservation d'un étalon	
6.13	matériau de référence (MR)	(ISO 30: 1992)
6.14	matériau de référence certifié (MRC)	“Certificat de matériau de référence” (ISO 30: 1992)
Total	120 termes	~ 52 termes ajoutés

RÉFÉRENCES

- [1] Abran, A. (1998). *Software Metrics Need to Mature into Software Metrology*. Paper presented at the NIST Workshop on Advancing Measurements and Testing for Information Technology (IT), Gaithersburg (Maryland), October 26-27, <http://www.lrgl.uqam.ca/publications/laboratory.jsp?custom1=All+Publications+by+date>.
- [2] Abran, A., Buglione, L., & Sellami, A. (2004). *Software Measurement Body of Knowledge - Initial Validation using Vincenti's Classification of Engineering Knowledge types*. Paper presented at the Software Measurement - Research and Application - Proceedings of the International Workshop on Software Metrics and DASMA - IWSM/ Metrikon 2004, Königs Wusterhausen, Germany, 255-270, 2-5 November,
- [3] Abran, A., Desharnais, J. M., Oligny, S., St-Pierre, D., & Symons, C. (2001). *COSMIC FFP Measurement Manual*, <http://www.lrgl.uqam.ca/publications/laboratory.jsp?custom1=All+Publications+by+date>.
- [4] Abran, A., Desharnais, J. M., Oligny, S., St-Pierre, D., & Symons, C. (2003). *COSMIC FFP Measurement Manual - Version 2.2, The COSMIC Implementation Guide for ISO/IEC 19761: 2003*. Montreal (Canada): École de technologie supérieure – ETS, <http://www.lrgl.uqam.ca/cosmic-ffp/>.
- [5] Abran, A., & Dumke, R. (1997). *A Framework of the Software Measurement Domain: Current Situation and Future Directions*. Montreal (Canada), pp.
- [6] Abran, A., & Jacquet, J. P. (1999). *A Structured Analysis of the New ISO Standard on Functional Size Measurement-Definition of Concepts on: "Functional Size Measurement - Definition of Concepts" (ISO/IEC 14143-1)*. *Fourth IEEE International Software Engineering Standards Symposium*, Curitiba, Brazil, 230-241.
- [7] Abran, A., Lopez, M., & Habra, N. (2004). *An analysis of the McCabe Cyclomatic complexity number*. Paper presented at the Software Measurement - Research and Application, Königs Wusterhausen, Germany,
- [8] Abran, A., Moore, J., Bourque, P., Dupuis, R., & Tripp, L. (2001). *Guide to the Software Engineering Body of Knowledge - SWEBOK Trial Version*. Los Alamitos (USA): IEEE-Computer Society Press.
- [9] Abran, A., Moore, J., Bourque, P., Dupuis, R., & Tripp, L. (2004). *Guide to the Software Engineering Body of Knowledge - SWEBOK Trial Version*. Los Alamitos (USA): IEEE-Computer Society Press.

- [10] Abran, A., & Robillard, P. N. (1994). Function Points: A Study of Their Measurement Processes and Scale Transformations. *Journal of Systems and Software*, 25(2), 171.
- [11] Abran, A., & Robillard, P. N. (1996). Function points analysis: an empirical study of its measurement processes. *IEEE Transactions on Software Engineering*, 22(12), 895-910.
- [12] Abran, A., & Sellami, A. (2002). Initial Modeling of the Measurement Concepts in the ISO Vocabulary of Terms in Metrology. *12th International Workshop on Software Measurement - IWSM*, Magdeburg, Germany, 315, ISBN 3-8322-0765-1.
- [13] Abran, A., & Sellami, A. (2002). Measurement and Metrology Requirements for Empirical Studies in Software Engineering. *IEEE Software Technology and Engineering Practice Workshop (STEP)*, Montreal (Canada).
- [14] Abran, A., Sellami, A., & Suryn, W. (2003). Metrology, measurement and metrics in software engineering. *Ninth International Software Metrics Symposium (METRICS'03) IEEE*, Sydney (Australia), Sept. 3-5.
- [15] Albrecht, A. (1979). *Measuring Application Development Productivity*, 83-92,
- [16] Azzouz, S., & Abran, A. (2004). A proposed measurement role in the Rational Unified Process (RUP) and its implementation with ISO 19761: COSMIC FFP. *Software Measurement European Forum 2004*, Rome, Italy.
- [17] Basili, V. R., & Rombach, H. D. (1988). The TAME project: Towards improvement-oriented software environments. *IEEE Transactions on Software Engineering*, 14(6), 758-773.
- [18] Basili, V. R., & Weiss, D. (1984). A Methodology for Collecting Valid Software Engineering Data. *IEEE Transactions on Software Engineering*, 10(6), 728-738.
- [19] Berg, K. G. V. d., & Broek, P. M. V. d. (1996). Axiomatic Validation in the Software Metric Development Process. In A. Melton (Ed.), *Software Measurement* (pp. 157): International Thomson Publishing Company.
- [20] Bieman, J. M., & Schultz, J. (1992). An Empirical Evaluation (and Specification) of the all-du-paths. *Software Engineering Journal*, 7(1), 43-51.
- [21] Boehm, B. W. (1981). *Software Engineering Economics*: New York, Prentice Hall.
- [22] Boehm, B. W. (2000). *Software Cost Estimation with COCOMO II*: Prentice Hall.
- [23] Bourque, P., Dupuis, R., Abran, A., Moore, J.-W., Tripp, L., & Wolff, S. (2002). Fundamental Principles of Software Engineering - A Journey. *The Journal of Systems and Software*, www.elsevier.com/locate/jss.

- [24] Bourque, P., Wolff, S., Dupuis, R., Sellami, A., & Abran, A. (2004). Lack of Consensus on Measurement in Software Engineering: Investigation of Related Issues. *14th International Workshop on Software Measurement (IWSM) IWSM-Metrikon 2004*, Königs Wusterhausen, Magdeburg, Germany, 321-333, 3-8322-3383-0.
- [25] Briand, L., El-Emam, K., & Morasca, S. (1995). *On the Application of Measurement Theory in Software Engineering*: International Software Engineering Research Network, Technical report,
- [26] Buglione, L., & Abran, A. (2002). *ICEBERG: a different look at Software Project Management*. Paper presented at the 12th International Workshop on Software Measurement (IWSM2002) - Software Measurement and Estimation, Magdeburg (Germany), 153-167, October 7-9, <http://www.lrgl.uqam.ca/publications/pdf/757.pdf>.
- [27] Buglione, L., & Abran, A. (2004). The Software Measurement Body of Knowledge. *Software Measurement European Forum, SMEF 2004*, Rome, Italy, 84-98, 88-86674-33-3.
- [28] Carnahan, L., Carver, G., Gray, M., Hogan, M., Hopp, T., Horlick, J., et al. (1997). Metrology for Information Technology. *StandardView*, 5(3), 103-109, ISSN:1067-9936.
- [29] Chidamber, S. R., & Kemerer, C. F. (1994). A Metrics Suite for Object Oriented Design. *IEEE Transactions on Software Engineering*, 20(6), 476-493.
- [30] Cloarec. (2003). *Métrologie: gérer et maîtriser les processus et les équipements de mesure*: AFNOR.
- [31] Cloarec, J.-Y. (2003). *Métrologie: Gérer et Maîtriser les Processus et les Équipements de Mesure*: Recueil Normes & Réglementation -Qualité, AFNOR.
- [32] CMMI Development Team. (2002). *Capability Maturity Model - Integrated Systems/Software Engineering*: Software Engineering Institute.
- [33] Curtis, B. (1980). Measurement and Experimentation in Software Engineering. *Proceedings of the IEEE*, 68(9), 1144-1157.
- [34] Dimitrov, E., Dumke, R., Foltin, E., & Wipprecht, M. (1999). Conception and Experience of Metrics Based Software Reuse in Practice. Lac Supérieur, Québec, pp.
- [35] Dubé, L., & Paré, G. (2003). Rigor in Information Systems Positivist Case Research: Current Practices, Trends, and Recommendations. *MIS Quarterly*, 27(4), 597-635.
- [36] Dumke, R., Foltin, E., & Bourque, P. (1996). Issues when Evaluating Software Measurement Tools Based on Existing Standards. *Forum on Software Engineering Standards (SES'96)*, USA.

- [37] Dumke, R., Lothar, M., & Abran, A. (2001). An Approach for Integrated Software Measurement Processes in the IT Area. *4th European Conference on Software Measurement and ICT Control in cooperation with DASMA*, Heidelberg, Germany.
- [38] El-Emam, K. (2000). *A Methodology for Validating Software Product Metrics*: National Research Council of Canada,
- [39] Fenton, N. E. (1994). Software Measurement: A Necessary Scientific Basis. *IEEE Transactions on Software Engineering*, 20(3), 28-37.
- [40] Fenton, N. E., & Kitchenham, B. (1991). Validating Software Measures. *Journal of Software Technology*, 1(2), 27-42.
- [41] Fenton, N. E., & Pfleeger, S. L. (1997). *Software Metrics: A Rigorous and Practical Approach*: second edition, PWS Publishing Co.
- [42] Fetcke, T., Abran, A., & Dumke, R. (2001). A Generalized Representation for Selected Functional Size Measurement Methods. *11th International Workshop on Software Measurement*, Montréal, 1-25.
- [43] Figliola, R. S., & Beasley, D. E. (1995). *Theory and Design for Mechanical Measurements*: John Wiley & Sons, Inc.
- [44] Gagnon, M., & Hébert, D. (2000). *En Quête de Science: Introduction à l'Épistémologie*. Canada: Éditions Fides.
- [45] Gile, A. E., & Daich, G. T. (1995). Metric Tools. *Crosstalk*, 8(2).
- [46] Halstead, M. H. (1977). *Element of Software Science*: New York, Elsevier North-Holland.
- [47] Henderson-Sellers, B. (1996). *Object-Oriented Metrics - Measures of Complexity*. New Jersey: Prentice Hall.
- [48] Himbert, M. <http://www.metrodiff.org/activit/rehseis/m.himbert.htm>
- [49] Hitz, M., & Montazeri, B. (1996). Chidamber and Kemerer's Metrics Suite: A Measurement Theory Perspective. *IEEE Transactions on Software Engineering*, 22(4), 267 -271.
- [50] Idri, A. (2003). *Un modèle intelligent d'estimation des coûts de développement de logiciels*. Unpublished PhD, UQAM, Montréal.
- [51] IEC Publication 60050. (2001). *International Electrotechnical Vocabulary - Chapters 300, Parts 311 and 314*.
- [52] IEEE IEEE 610.12: 1991. (1991). *IEEE Standard Glossary for Software Engineering Terminology*. USA: IEEE Computer Society.
- [53] IEEE IEEE Std 830: 1998. (1998). *IEEE Recommended Practice for Software Requirements Specifications*: IEEE Computer Society.

- [54] IFPUG. (1994). *Function Point Counting Practices Manual*. Westerville, Ohio: International Function Point Users Group (IFPUG), Version 4.0.
- [55] IFPUG. (1999). *Function Point Counting Practices Manual*. Westerville, Ohio: International Function Point Users Group (IFPUG), Version 4.1.
- [56] ISO 31. (1992). *Quantities and units -- Part 0: General principles*: International Organization for Standardization - ISO.
- [57] ISO. (1993). *International Vocabulary of Basic and General Terms in Metrology* (2nd ed.). Switzerland: International Organization for Standardization - ISO.
- [58] ISO GUM. (1993). *Guide to the Expression of Uncertainty in Measurement (1993 amended 1995)*: published by ISO in the name of BIPM, IEC, IFCC, IUPAC, IUPAP and OIML.
- [59] ISO ISO/IEC 9126-1 to 3: 2001. (2001 to 2003). *Information Technology - Software Quality Characteristics and Metrics; Part 1: Quality model; Part 2: External metrics; Part 3: Internal metrics*. Geneva: International Organization for Standardization - ISO.
- [60] ISO ISO/IEC 9126-1: 2001. (2001). *Software Engineering - Product Quality - Part 1: Quality model*. Geneva: International Organization for Standardization - ISO.
- [61] ISO ISO/IEC 12207: 1995. (1995). *International Standard 12207 - Information Technology: Software Life Cycle Processes*. Geneva: International Organization for Standardization - ISO.
- [62] ISO ISO/IEC 14143-1: 1998. (1998). *Information technology -- Software measurement -- Functional size measurement -- Part 1: Definition of concepts*. Geneva: International Organization for Standardization - ISO.
- [63] ISO ISO/IEC 14143-2: 1999. (1999). *Software Engineering - Functional Size Measurement - Part3: Verification of Functional Size Measurement Methods*. Geneva: International Organization for Standardization - ISO.
- [64] ISO ISO/IEC 14143-3: 2003. (2003). *Software Engineering- Functional Size Measurement- Part 3: Verification of Functional Size Measurement Methods*. Geneva: International Organization for Standardization - ISO.
- [65] ISO ISO/IEC 14143-4: 2000. (2000). *Software Engineering - Functional Size Measurement - Part 4: Reference Model*. Geneva: International Organization for Standardization - ISO.
- [66] ISO ISO/IEC 14143-5: 2003. (2003). *Software Engineering - Functional Size Measurement- Part 5: Determination of Functional Domains for Use with Functional Size Measurement*. Geneva: International Organization for Standardization - ISO.

- [67] ISO ISO/IEC 15939: 2002. (2002). *Information Technology - Software Engineering - Software Measurement Process*. Geneva: International Organization for Standardization - ISO.
- [68] ISO ISO/IEC 19759: 2003. (2003). *Guide to the Software Engineering Body of Knowledge - SWEBOOK*. Geneva: International Organization for Standardization - ISO.
- [69] ISO ISO/IEC 19761: 2003. (2003). *Software Engineering -- COSMIC-FFP -- A Functional Size Measurement Method*. Geneva: International Organization for Standardization - ISO.
- [70] ISO ISO/IEC PDTR 25021. (2004). *Software Engineering: Software Product Quality Requirements and Evaluation (SQuaRE) - Measurement Primitives*. Geneva: International Organization for Standardization - ISO.
- [71] ISO ISO/IEC TR 9126-2: 2003. (2003). *Software Engineering - Product Quality - Part 2: External Metrics*. Geneva: International Organization for Standardization - ISO.
- [72] ISO ISO/IEC TR 9126-3: 2003. (2003). *Software Engineering - Product Quality - Part 3: Internal Metrics*. Geneva: International Organization for Standardization - ISO.
- [73] ISO ISO/IEC TR 9126-4: 2004. (2004). *Software Engineering - Product Quality - Part 4: Quality in Use Metrics*. Geneva: International Organization for Standardization - ISO.
- [74] ISO VIM. (2004). *International Vocabulary of Basic and General Terms in Metrology (Draft)*: International Organization for Standardization - ISO.
- [75] Jabir, & Moore, J. W. (1998). *A Search for Fundamental Principles of Software Engineering*. Paper presented at the Computer Standards & Interfaces -The International Journal on the Development and application of Standards for Computers, 1998. Data Communications and Interfaces. Report of a workshop conducted at the forum on Software Engineering Standards Issues, 19 North-Holland, Elsevier, 2, 155-160,
- [76] Jacquet, J. P., & Abran, A. (1997). From Software Metrics to Software Measurement Methods: A Process Model. *International Software Engineering Standards Symposium and Forum, ISESS 97*, 128-135.
- [77] Jacquet, J. P., & Abran, A. (1999). Metric Validation Proposals: A Structured Analysis. In *Software Measurement - Current Trends in Research and Practice, Proceedings of the 8th International Workshop on Software Measurement - IWSM 98* (pp. 43-59): Deutscher Universität Verlag.
- [78] Jacquet, J. P., Abran, A., & Dupuis, R. (1997). Une analyse structurée des méthodes de validation de métriques.

- [79] Kelvin, L. (1889). *Popular Lectures and Addresses. I.*
- [80] Khelifi, A., Abran, A., & Buglione, L. (2004). *A System of References for Software Measurements with ISO 19761 (COSMIC-FFP)*. Paper presented at the Software Measurement - Research and Application - Proceedings of the International Workshop on Software Metrics and DASMA - IWSM/ Metrikon 2004, Königs Wusterhausen, Germany,
- [81] Kirby, R.-S., Withington, S., Darling, A.-B., & Kilgour, F.-G. (1990). *Engineering in History*. New York: Dover Publications Inc.
- [82] Kitchenham, B., Pfleeger, S. L., & Fenton, N. E. (1995). Towards a Framework for Software Measurement Validation. *IEEE Transactions on Software Engineering*, 21(12), 929 -944.
- [83] Lopez, M., Paulus, V., & Habra, N. (2003). *Integrated Validation Process of Software Measure*. Paper presented at the International Workshop on Software Measurement (IWSM),
- [84] McCabe, T. J. (1976). A Complexity Measure. *IEEE Transactions on Software Engineering*, 2(4), 308-320.
- [85] McCabe, T. J., & Butler, C. W. (1989). Design Complexity Measurement and Testing. *Communications of the ACM*, 32(12).
- [86] Melton, A. (1996). *Software Measurement*: Thomson Computer Press.
- [87] MK II. (1998). *Function Point Analysis: Counting Practices Manual*: United Kingdom Software Metrics Association.
- [88] Morris, P. M., & Desharnais, J.-M. (1996). Validation of Function Points - An Experimental Perspective IFPUG. Atlanta, Georgia.
- [89] NESMA. (1997). *Definitions and Counting Guidelines for the Application of Function Point Analysis: A Practical Manual*: version 2.0.
- [90] Oman, P., & Pfleeger, S. L. (1997). *Applying Software Metrics*: IEEE Computer Society Press.
- [91] Pérard, A., & Terrien, J. (1962). *Mesures Physiques*: Presses universitaires de France.
- [92] Pfleeger, S. L. (1997). *Software Engineering: Theory and Practice*: Prentice Hall.
- [93] Placko, M. (2000). *Mesure et Instrumentation*. Paris: Hermès Science Publications.
- [94] Prather, R. E. (1984). An Axiomatic Theory of Software Complexity Metrics. *The Computer Journal*, 27, 42-45.
- [95] Putnam, L. H. (1978). A General Empirical Solution to the Macro Software Sizing and Estimation Problem. *Transactions on Software Engineering*, 4(4).

- [96] Robert, F., Abran, A., & Bourque, P. (2002). A Technical Review of the Software Construction Knowledge Area in the SWEBOK Guide. *10th International Workshop on Software Technology and Engineering Practice, STEP 2002*, Montréal, Québec, Canada.
- [97] Rombach, D. (1991). Practical Benefits of Goal-Oriented Measurement. In *Fenton B.; Littlewood, B. Software Reliability and Metrics*: Elsevier Applied Science.
- [98] Rubey, R. J., & Hartwick, R. D. (1968). *Quantitative Measurement Program Quality*. Paper presented at the ACM, National Computer Conference, 671-677,
- [99] Rumbaugh, J., Jacobson, I., & Booch, G. (2004). *The Unified Modeling Language Reference Manual*: Hardcover.
- [100] Schneidewind, N. E. (1992). Methodology for Validating Software Metrics. *IEEE Transactions on Software Engineering*, 18(5), 410-422.
- [101] Schneidewind, N. E. (2002). Body of Knowledge for Software Quality Measurement. *IEEE Transactions on Software Engineering*, 35(2), 77-83.
- [102] Sellami, A., & Abran, A. (2003). The Contribution of Metrology Concepts to Understanding and Clarifying a Proposed Framework for Software Measurement Validation. *13th International Workshop on Software Measurement - IWSM*, Montreal (Canada), 326, ISBN: 3-8322-1880-7.
- [103] Shaw, M. (1990). Prospects for an Engineering Discipline of Software. *IEEE Transactions on Software Engineering*, 930-940.
- [104] Shepperd, M., & Ince, D. (1993). *Derivation and Validation of Software Metrics*. Oxford: Science Publications.
- [105] St-Pierre, D., Maya, M., Abran, A., Desharnais, J.-M., & Bourque, P. (1997). *Full Function Points: Counting Practices Manual*. Montreal (Canada): Software Engineering Management Research Laboratory, UQAM,
- [106] Stevens, S. S. (1946). On the Theory of Scales and Measurement. *Science Journal*, 103, 677-680.
- [107] Symons, C. R. (1988). Function Point Analysis: Difficulties and Improvements. *IEEE Transactions on Software Engineering*, 14(1), 2-11.
- [108] Tichy, W. F. (2000). Hints for Reviewing Empirical Work in Software Engineering. *Empirical Software Engineering*, 5, Boston, 309-312.
- [109] Vincenti, G. W. (1990). *What Engineers Know and How They Know It - Analytical Studies from Aeronautical History*: Johns Hopkins University Press.
- [110] Watson, A. H., & McCabe, T. J. (1996). Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric.

- [111] Wolff, S. (1999). *La Place de la Mesure au Sein des Principes Fondamentaux du Génie Logiciel* (Activité de Synthèse). Montreal: UQAM,
- [112] Zelkowitz, M. V., & Wallace, D. R. (1998). Experimental Models for Validating Technology. *IEEE Computer Society Press*, 31(5), 23 - 31.
- [113] Zuse, H. (1991). *Software Complexity Measures and Methods*. Germany, Berlin: Walter de Gruyter.
- [114] Zuse, H. (1997). *A Framework for Software Measurement*. Walter de Gruyter, Germany, Berlin.
- [115] Zuse, H. (1999). *Validation of Measures and Prediction Models*. Paper presented at the International Workshop on Software Measurement (IWSM),

