

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
ADEL KHELIFI

UN RÉFÉRENTIEL POUR LA MESURE DES LOGICIELS
AVEC LA NORME ISO 19761 (COSMIC-FFP) :UNE ÉTUDE EXPLORATOIRE

MONTREAL, 15 DECEMBRE 2005

(c) droits réservés de Adel Khelifi

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

M. Jean-Marc Desharnais, membre du jury
Département de génie logiciel et des technologies de l'information
à l'École de technologie supérieure

M. Ali Idri, membre du jury
École Nationale Supérieure d'Informatique et d'Analyse des Systèmes de l'Université
Mohammed V Souissi au Maroc

ELLE A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 11 AOUT 2005

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

À mes parents

UN RÉFÉRENTIEL POUR LA MESURE DES LOGICIELS AVEC LA NORME ISO 19761 (COSMIC-FFP) : UNE ÉTUDE EXPLORATOIRE

Adel Khelifi

SOMMAIRE

Sommaire:

La problématique générale de cette thèse est le manque de standards et de références pour le design et pour l'utilisation des mesures en génie logiciel. Cette problématique rend le processus de mesurage des logiciels complexe et difficile. La méthodologie adoptée pour résoudre la problématique indiquée ci-dessus est à la fois exploratoire et qualitative.

L'objectif de cette thèse est la construction d'un référentiel pour la mesure des logiciels. Le référentiel est composé de huit étalons de mesure des logiciels. Dans notre contexte, un étalon est défini comme « une mesure matérialisée, appareil de mesure, matériau de référence destiné à définir, réaliser, conserver ou reproduire une unité ou une ou plusieurs valeurs d'une grandeur pour servir de référence » (ISO, 1993).

Ces étalons de mesure traitent un seul attribut du logiciel, soit la taille fonctionnelle, et ils touchent trois types de logiciels : applications de gestion, systèmes en temps réel et systèmes hybrides. Pour mesurer la taille fonctionnelle, ces étalons utilisent une mesure appelée les points de fonction (en anglais Function Point Analysis - FPA).

«FPA, introduced in 1979 by Allan Albrecht, was proposed to help measure the productivity of software development. Function points have the advantage of well-known norms and well defined detailed procedures that allow a uniform method of counting and the establishment of a reference base for making comparisons, provided, of course, that the rules of calculation and accounting are correctly applied.» (Abran, 1994)

Pour la construction du référentiel, une méthode de mesure des logiciels bien spécifique a été choisie, soit COSMIC-FFP (ISO 19761). COSMIC-FFP est une méthode rigoureuse de mesure des logiciels. Elle rend possible la mesure de la taille des logiciels tels qu'ils sont perçus par leurs utilisateurs, c'est-à-dire à travers les besoins fonctionnels. Le résultat de l'application de la mesure donne une valeur numérique qui quantifie la taille fonctionnelle du logiciel. Les raisons de la sélection de cette mesure spécifique et sa description sont expliquées dans le quatrième chapitre.

La méthode est appliquée sur des spécifications des logiciels appelées encore Fonctionnalités Utilisateurs Requisites (FUR). Selon le manuel de mesure (A. Abran *et*

al., 2003) de la méthode COSMIC-FFP, l'expression « Fonctionnalité Utilisateur Requise » (FUR) réfère à un sous-ensemble des besoins de l'utilisateur. Le mot FUR est utilisé dans cette recherche sous cette définition. La FUR représente les pratiques et les procédures de l'utilisateur que le logiciel doit accomplir pour répondre aux besoins de celui-ci. La FUR exclut les besoins en qualité et les besoins techniques. Pour cette recherche, huit ensembles de FUR différents à mesurer ont été sélectionnés. Cinq d'entre elles proviennent du rapport technique ISO (ISO/TR 14143-4); ce sont : *Automatic Line Switching System*, *Hotel Reservation System*, *L-Euchre Application*, *SAGA System* et *Valve Control System*. Un autre ensemble de FUR appartient à l'application *Rice Cooker*. Les deux autres appartiennent aux documents de formation de la compagnie IBM-Rational, ce sont *C-Registration System* et *Collegiate Sports Paging System*. La compagnie a donné une permission écrite à l'étudiant mandataire pour utiliser ces cas dans cette thèse (voir Annexe 9). Les critères de sélection de ces spécifications logicielles sont leur disponibilité et leur origine. De plus, chacun des étalons de mesure sera vérifié par des experts d'ISO et de l'industrie.

Le référentiel construit porte sur les mesures de la taille fonctionnelle (MTF) des logiciels avec une seule méthode MTF. La portée de cette recherche pourrait probablement couvrir d'autres MTF et potentiellement d'autres mesures des logiciels ainsi de nombreux travaux ou programmes de recherche peuvent être envisagés.

Mots-clés : Génie logiciel, Étalon de mesure, Mesure de logiciel, Métrologie, Standardisation, Taille fonctionnelle, COSMIC-FFP, ISO 19761, ISO 14143.

A SET OF REFERENCES FOR SOFTWARE MEASUREMENT WITH ISO 19761 (COSMIC-FFP): AN EXPLORATORY STUDY

Adel Khelifi

ABSTRACT

Abstract:

The major topic of this thesis addresses the lack of standards and references in designing and using software measurements. This issue makes the measuring process a challenging and complex task in software engineering.

The main objective of this thesis is the construction of a set of references for software measurements. The set is composed of eight software standard etalons. In our context, a standard etalon is "a material measure, measuring instrument, reference material or measuring system intended to define, realize, conserve or reproduce a unit or one or more values of a quantity to serve as a reference" (ISO, 1993).

These standard etalons treat only one attribute of the software, that is the functional size, and they cover three types of software; business applications, real-time system and hybrid system. To measure the functional size, these standard etalons use a measure called Function Point Analysis (FPA). «FPA, introduced in 1979 by Allan Albrecht, was proposed to help measure the productivity of software development. Function points have the advantage of well-known norms and well defined detailed procedures that allow a uniform method of counting and the establishment of a reference base for making comparisons, provided, of course, that the rules of calculation and accounting are correctly applied.» (Abran, 1994)

The set adopts a specific method of software measurement which is COSMIC-FFP (ISO 19761). COSMIC-FFP is a functional size measurement method, which generalizes the measurement process to address management information systems issues, as well as real-time and hybrid software projects. The reasons of choosing this measurement method, and its description, are explained in chapter four.

The method is applied to a set of Functional User Requirements (FUR). The term Functional User Requirement is an ISO expression designating a subset of the user requirements. The word FUR is used in thesis under this definition. It represents the user practices and procedures that the software must perform to fulfil the user's needs, excluding quality requirements and any technical requirements (ISO/IEC 14143-1,

1998). For this research we selected eight sets of different FUR to measure. Five of them come from the ISO 14143-4 technical report; they are *Automatic Line Switching System*, *Hotel System Reservation*, *L-Euchre Application*, *SAGA System* and *Valve System Control*. A FUR belongs to the *Rice Cooker application*. The last two sets of FUR belong to the training documents of the IBM-Rational company: they are *C-Registration System* and *Collegiate Sports Paging System*. The IBM-Rational gave permission to the software engineering laboratory of ETS (GÉLOG) to use their training cases studies for this research (see Annexe 9).

The criteria of selection of these software specifications are their availability and their origin. Each standard etalon proposed (case study) will be checked by ISO and industrial experts.

This research could cover other software functional size measurement (FSM) and potentially other measurements. Thus many research activities can be derived from this work. The set of references, here, relates to only FSM and deals with a single FSM method.

Keywords: Software Engineering, Standard Etalon, Software Measurement, Metrology, Standardisation, Functional Size, COSMIC-FFP, ISO 19761, ISO 14143.

REMERCIEMENTS

Finalement, après des années de travail ardu, vient le moment des remerciements. Chaque personne qui y est citée ici mérite la plus belle phrase de gratitude.

Cette thèse n'aurait vu le jour sans la confiance, la patience et la générosité de Monsieur Alain Abran, mon directeur de recherche, professeur au Département de génie logiciel et des TI, que je veux vivement remercier. La pleine confiance qu'il m'a accordée, m'a permis d'élaborer un plan de travail personnel et propre à mes aspirations. Je le remercie pour le temps et la patience qu'il m'a accordés tout au long de ces années, d'avoir cru en mes capacités et de m'avoir fourni un excellent climat de travail. Les conseils qu'il m'a prodigués tout au long de ces années ont toujours été clairs et succincts, me facilitant grandement la tâche et me permettant d'aboutir à la production de cette thèse.

Je tiens à exprimer toute ma reconnaissance à M. Witold Suryn qui m'a fait l'honneur d'accepter d'être président de mon jury de thèse.

J'exprime ma profonde gratitude à M. Jean-Marc Desharnais et M. Ali Idri pour avoir acceptés de faire partie de mon jury de thèse.

Je remercie tous les amis, chercheurs, enseignants et membres du personnel du laboratoire GÉLOG ainsi que le personnel du Département de génie logiciel et des TI pour leur amitié et leur aide pendant ces trois années de thèse.

Je tiens à témoigner ma sympathie et ma reconnaissance aux membres du groupe de travail ISO/IEC JTC/SC7 WG12 et tout particulièrement à M. Charles Symons; grâce à lui, j'ai pu découvrir et apprivoiser quelques mystères de la mesure des logiciels. Il fait incontestablement partie des gens sans qui ce travail n'aurait sans doute jamais abouti.

Je tiens à remercier aussi les développeurs et les gestionnaires du processus RUP, M. P. Kruchten, M. P. Kroll et M. T. J. Collins de m'avoir donné la permission officielle d'utiliser deux études de cas de la documentation de formation du processus RUP.

Mes plus grands remerciements vont bien évidemment à mon épouse qui, pendant ma thèse, a donné naissance à notre fille Ala et notre fils Bara, tout en travaillant elle-même à sa thèse dans des conditions pas toujours faciles.

Mes profonds remerciements à mes frères et sœurs pour leur intérêt envers mon travail ainsi qu'à ma belle famille, pour son aide précieuse.

Je termine par un grand remerciement à mes parents auxquels je dédie cette thèse.

TABLE DES MATIÈRES

	Page
SOMMAIRE	i
ABSTRACT	iii
REMERCIEMENTS.....	v
TABLE DES MATIÈRES	vii
Liste des tableaux.....	xii
Liste des figures.....	xiv
Liste des abréviations et des sigles.....	xvi
GLOSSAIRE	xvii
CHAPITRE 1 INTRODUCTION	21
1.1 Mise en contexte.....	22
1.2 Composition de la thèse.....	23
CHAPITRE 2 LES MESURES EN GÉNIE LOGICIEL	29
2.1 Introduction	29
2.2 Place de la mesure en génie logiciel.....	29
2.3 Le besoin en mesure dans le génie logiciel	32
2.4 La classification des mesures en génie logiciel	34
2.5 Sommaire.....	38
CHAPITRE 3 PROPRIÉTÉS DES MESURES EN GÉNIE LOGICIEL	39
3.1 Introduction	39
3.2 Inventaire des propriétés proposées pour les mesures des logiciels.....	39
3.3 Constatation sur les propriétés des mesures des logiciels	49
3.4 Les étalons de mesure.....	50
3.5 Conclusion.....	51
CHAPITRE 4 LA MÉTROLOGIE ET LA STANDARDISATION	52
4.1 Introduction	52
4.2 Définition de la métrologie.....	52

4.3	La standardisation.....	54
4.3.1	La définition du standard.....	54
4.3.2	Le contenu d'une norme.....	55
4.3.3	Rôle des normes.....	56
4.3.4	Types de normes.....	57
4.4	La métrologie dans ISO.....	58
4.5	Conclusion.....	59
CHAPITRE 5 LES MESURES DE LA TAILLE FONCTIONNELLE (MTF).....		61
5.1	Introduction.....	61
5.2	Les points de fonction.....	63
5.3	Lignes de code ou points de fonction.....	64
5.3.1	Les avantages de la méthode des lignes de code.....	64
5.3.2	Les inconvénients de la méthode des lignes de code.....	64
5.3.3	Les avantages de la méthode des points de fonction.....	65
5.3.4	Les inconvénients de la méthode des points de fonction.....	66
5.4	La méta-norme ISO 14143.....	67
5.5	IFPUG.....	71
5.6	NESMA - ISO/IEC 24570.....	72
5.7	MKII - ISO/IEC 20968.....	74
5.8	COSMIC-FFP (ISO 19761).....	75
5.9	Comparaison entre les méthodes IFPUG ET COSMIC-FFP.....	78
5.10	Pourquoi COSMIC-FFP ?.....	81
5.11	Sommaire.....	82
CHAPITRE 6 LA CONSTRUCTION D'UN ÉTALON.....		84
6.1	Introduction.....	84
6.2	Généralités.....	84
6.3	Qu'est ce qu'un étalon ?.....	85
6.3.1	Création du premier étalon universel : Le mètre.....	85
6.3.2	Les étalons universels et spécifiques.....	87
6.3.3	Organismes de normalisation.....	88
6.4	Étalonnage et calibration.....	88
6.5	La traçabilité des étalons.....	89
6.6	Conclusion.....	91
CHAPITRE 7 PRÉSENTATION DU TRAVAIL.....		92
7.1	Introduction.....	92
7.2	Problématique de recherche.....	93
7.3	Questions de recherche.....	99
7.4	Objectif de recherche.....	99
7.5	Sommaire.....	100
CHAPITRE 8 LA MÉTHODOLOGIE DE RECHERCHE.....		101

8.1	Introduction	101
8.2	Définition du projet	103
8.2.1	Motivation	103
8.2.2	Utilisateurs.....	103
8.2.3	Objectif.....	104
8.2.4	Les outils de travail pour faire la recherche	104
8.2.5	Limites de la recherche.....	105
8.3	Étapes de la méthodologie.....	106
8.3.1	Identifier la méthode de mesure pour développer des étalons.....	106
8.3.2	Réalisation d'un étalon	107
8.3.3	Ajout des diagrammes UML	108
8.3.4	Déterminer le gabarit du rapport d'étalon	109
8.3.5	La sélection des experts.....	111
8.4	Le développement d'un site web	112
8.5	Méthode de liaison d'une mesure au référentiel.....	114
8.6	Développement du référentiel	115
8.7	Les partenaires de la recherche.....	119
8.8	Sommaire.....	119
CHAPITRE 9 DÉROULEMENT DE LA RECHERCHE		121
9.1	Introduction	121
9.2	Sélection des FUR	121
9.3	Le processus de mesurage	122
9.4	Gabarit d'étalon utilisé	126
9.5	L'intervention des experts	129
9.6	Le développement d'un site web	130
9.7	Personnes impliquées dans la construction des étalons.....	133
9.8	Sommaire.....	136
CHAPITRE 10 DISCUSSION DES RÉSULTATS		137
10.1	Introduction	137
10.2	Le contenu d'un étalon	137
10.3	Le valve control system etalon	140
10.4	Overview	143
10.4.1	Introduction	143
10.4.2	Measurement viewpoint, purpose and scope.....	143
10.5	Requirements – as documented in ISO 14143-4:2000	144
10.5.1	Context.....	144
10.5.2	Input	145
10.5.3	Output	145
10.6	Requirements	145
10.6.1	Part a – determine general operating condition	145
10.6.2	Part b – control to open hydraulic valve slowly from its closed state	146
10.6.3	Part c – control to open hydraulic valve quickly from its closed state.....	147

10.7	COSMIC-FFP measurement procedure.....	147
10.7.1	Identification of layers.....	147
10.7.2	Identification of users.....	147
10.7.3	Boundary.....	148
10.7.4	Identification of triggering events.....	148
10.7.5	Identification of data groups.....	149
10.7.6	Identification of functional processes.....	150
10.8	Identify data movements.....	151
10.8.1	Use case diagrams.....	152
10.8.2	Message sequence diagram.....	152
10.8.3	List of data movements.....	153
10.8.4	Observations on the clarity of the documented requirements.....	154
10.9	Analysis of measurement results.....	156
10.10	Conclusions and observations.....	157
10.11	Questions and answers.....	158
10.12	Sommaire.....	159
CHAPITRE 11 LE RÉFÉRENTIEL POUR LA MTF.....		160
11.1	Introduction.....	160
11.2	Description du référentiel.....	161
11.3	Description des résultats trouvés.....	164
11.3.1	Premier étalon.....	165
11.3.2	Deuxième étalon.....	167
11.3.3	Troisième étalon.....	171
11.3.4	Quatrième étalon.....	174
11.3.5	Cinquième étalon.....	179
11.3.6	Sixième étalon.....	182
11.3.7	Septième étalon.....	184
11.3.8	Huitième étalon.....	187
11.4	Affichage sur le web.....	190
11.5	Évaluation des résultats.....	192
11.6	Conclusion.....	193
CHAPITRE 12 VÉRIFICATION DES ÉTALONS.....		195
12.1	Introduction.....	195
12.2	Absence de consensus.....	196
12.3	Nécessité d'un consensus.....	196
12.4	Procédures de vérification.....	197
12.4.1	Choix et rôles des experts.....	197
12.4.2	Définition de la vérification.....	198
12.4.3	Hypothèses de vérification.....	199
12.4.4	Conception de la vérification.....	200
12.5	Déroulement de la vérification.....	201
12.6	Portée de la vérification.....	207

12.7	Sommaire.....	208
CHAPITRE 13 CONCLUSION GÉNÉRALE.....		210
13.1	Introduction	210
13.2	Ce qui a été fait.....	211
13.3	Ce qui a été appris	213
13.4	Les contributions de la recherche	214
13.5	Conséquences des résultats.....	215
13.5.1	Utilité de ce travail pour la certification des mesureurs MTF	215
13.5.2	Les principes fondamentaux de certification.....	216
13.6	Les limites et les difficultés rencontrées.....	218
13.7	Un dernier mot.....	219
RÉFÉRENCES		220

ANNEXES:

1. Automatic Line Switching Etalon
2. SAVAT Gateway System Etalon
3. Hotel Reservation System Etalon
4. L-Euchre Card Game System Etalon
5. Rice Cooker System Etalon
6. C-Registration System Etalon
7. Collegiate Sports Paging System Etalon
8. Notes biographiques de quelques experts
9. Permissions

Voir CD-ROM.

LISTE DES TABLEAUX

	Page
Tableau I	Échelles de mesure (A. Abran, 1994)37
Tableau II	Les propriétés des mesures de logiciels proposées par Bache 40
Tableau III	Les propriétés des mesures de logiciels identifiées par Conte et al. ...41
Tableau IV	Les propriétés des mesures de logiciels exigées par IEEE41
Tableau V	Les propriétés des mesures de logiciels exigées par Basili et Reiter .42
Tableau VI	Les propriétés des mesures de logiciels exigées par Ejiogu42
Tableau VII	Les propriétés des mesures de logiciels exigées par Fenton43
Tableau VIII	Les propriétés des mesures de logiciels exigées par Jones44
Tableau IX	Les propriétés des mesures de logiciels exigées par Kearny et al.....45
Tableau X	Les propriétés des mesures de logiciel suggérées par Weyuker45
Tableau XI	Les propriétés des mesures exigées par Lakshmanan et al.47
Tableau XII	Les propriétés des mesures de logiciels exigées par Watts.....48
Tableau XIII	Les propriétés des mesures de logiciels exigées par Navlaka.....48
Tableau XIV	Les propriétés des mesures de logiciels définies par ISO 14143-3....49
Tableau XV	Objectifs d'utilisation des points de fonction (IFPUG, 1994)63
Tableau XVI	Comparaisons entre IFPUG et COSMIC-FFP 78
Tableau XVII	Comparaisons au niveau des processus.....79
Tableau XVIII	Comparaison d'utilisation des résultats pour estimation79
Tableau XIX	Comparaison des ressources80
Tableau XX	Cadre de Basili modifié pour la recherche exploratoire..... 112
Tableau XXI	Méthode de liaison au référentiel, adapté de Baize et al. (1995)114
Tableau XXII	La comparaison entre les deux gabarits 126
Tableau XXIII	Étalon 1 - Automatic Line Switching System..... 133
Tableau XXIV	Étalon 2 - SAGA System 133
Tableau XXV	Étalon 3 - Valve Control System 133
Tableau XXVI	Étalon 4 - Hotel Reservation System 134

Tableau XXVII	Étalon 5 - L-Euchre Application	134
Tableau XXVIII	Étalon 6 - Rice Cooker Application	135
Tableau XXIX	Étalon 7 - C-Registration System.....	136
Tableau XXX	Étalon 8 - Collegiate Sports Paging System	136
Tableau XXXI	List of Cosmic-FFP data movements.....	154
Table XXXII	Distribution of size units	156
Table XXXIII	Contribution of functional process to total Cfsu size.....	156
Table XXXIV	Percentage of the Cosmic-FFP data movement types.....	156
Tableau XXXV	Étalons - Type A	201
Tableau XXXVI	Étalons - Type B.....	202
Tableau XXXVII	Étalons - Type C	204

LISTE DES FIGURES

	Page
Figure 1	Présentation synoptique de la structure de la thèse.....24
Figure 2	Étapes de la définition des mesures numériques (Fenton et al. 1997) ...30
Figure 3	Lampe d'intensité lumineuse.....34
Figure 4	Classification de Fenton (1997)34
Figure 5	Place de la métrologie dans le concept Mesure (Abran et al., 2003)53
Figure 6	Relations entre les termes de métrologie (A. Abran et al., 2002)59
Figure 7	Types de fonction FPA71
Figure 8	La chaîne de traçabilité - adaptée de Preben et al. (2003)90
Figure 9	Le logiciel est difficile à mesurer.....94
Figure 10	Mesures des logiciels sans référence.....96
Figure 11	Difficultés de mesurer à partir de la documentation97
Figure 12	Problématique de recherché98
Figure 13	Gabarit initial d'étalon.....110
Figure 14	Gabarit final d'étalon.....111
Figure 15	Processus, intrants et livrables117
Figure 16	Les étapes principales de la méthodologie.....118
Figure 17	Modèle de mesure COSMIC-FFP (Abran et al., 2003)123
Figure 18	Méthode générale de mise en correspondance COSMIC-FFP124
Figure 19	Procédure générale de la phase de mesure COSMIC-FFP125
Figure 20	Site web des étalons de mesure.....131
Figure 21	Forum de discussion sur les étalons de mesure.....132
Figure 22	Le contenu d'un étalon138
Figure 23	Software Boundary.....148
Figure 24	Valve Control System - Use Case Diagram.....152
Figure 25	Valve Control Application - Message Sequence Diagram152
Figure 26	Valve Control System - Sequence Diagramme.....153

Figure 27	Proportion of COSMIC FFP data movement types	157
Figure 28	Le référentiel	161
Figure 29	Site web des étalons de mesure	191
Figure 30	Forum de discussion sur les étalons de mesure.....	192
Figure 31	Vérification individuelle	205
Figure 32	Vérification des étalons par des experts à travers le site web	206
Figure 33	Cycle d'amélioration de la qualité des etalons	207

LISTE DES ABRÉVIATIONS ET DES SIGLES

COSMIC	Common Software Measurement International Consortium
DET	Data Element Type
DG	Data Groupe
ÉTS	École de technologie supérieure
EP	Elementary Processus
FFP	Full Functional Point
FPA	Function Point Analysis
FUR	Fonctionnalités Utilisateurs Requises
GÉLOG	Laboratoire de recherche en génie logiciel
IEC	International Electrotechnical Commission
IFPUG	International Function Points User Group
ISO	International Organisation for Standardisation
JTC1	Joint Technical Committee One
MMRE	Mean Magnitude Relative Error
MTF	Mesure de la Taille Fonctionnelle
PF	Point de Fonction
PFE	Point de Fonction Étendu
RU	Références Utilisateurs
RUR	Références Utilisateurs Requises
SC7	Sub-Committee Seven
SI	Système International d'unité
TR	Technical Report
UML	Unified Modeling Language
WG12	Working Group number 12

GLOSSAIRE

Les termes suivants sont utilisés dans toutes les sections de la thèse sur la base des définitions de ce glossaire. La définition de la plus part de ces termes sont celles du manuel de mesure de la méthode COSMIC-FFP (A. Abran *et al.*, 2002).

Attribut (type) : La plus petite parcelle d'information codée, dans un groupe de données, possédant une signification dans la perspective des Fonctionnalités Utilisateurs Requises (FUR).

Cfsu : Unité COSMIC-FFP de la taille fonctionnelle (*COSMIC functional size unit*). Étalon de mesure COSMIC-FFP.

Couche : Le résultat du partitionnement fonctionnel de l'environnement du logiciel où tous les processus fonctionnels inclus s'exécutent au même niveau d'abstraction.

Dans un environnement logiciel à plusieurs couches, le logiciel dans une couche échange des données avec un logiciel dans une autre couche via leurs processus fonctionnels respectifs. Ces interactions sont hiérarchiques par nature ; lorsque considérée par paire, une couche est « cliente » de l'autre. Un morceau de logiciel d'une couche « cliente » utilise les services fonctionnels fournis par le logiciel des couches subalternes. Des morceaux de logiciel dans la même couche peuvent aussi échanger des données. Ce type d'échange de données est habituellement appelé un échange de données pair-à-pair.

Écriture (type) : Une écriture (C) est un type de mouvement de données qui, dans le contexte d'un processus fonctionnel, déplace un groupe de données depuis ce processus fonctionnel vers sa partie de stockage persistant associée dans ce contexte.

Remarque : Une écriture inclut certaines manipulations associées aux données nécessaires pour réaliser l'écriture –(voir la section 4.1 pour plus de détails).

Entrée (type) : Une Entrée (E) est un type de mouvement de données qui déplace, à travers la frontière, un groupe de données depuis un utilisateur vers le processus fonctionnel où il est requis.

Remarque 1 : Dans COSMIC-FFP, une entrée inclut aussi certaines manipulations associées des données (ex. : validation des données entrantes). Voir la section 4.1 pour plus de détails.

Remarque 2 : Une manipulation de données est tout ce qui se passe sur une donnée sauf un mouvement de données.

Étalon : un étalon est défini comme mesure matérialisée, appareil de mesure, matériau de référence destiné à définir, réaliser, conserver ou reproduire une unité ou une ou plusieurs valeurs d'une grandeur pour servir de référence (ISO, 1993).

Événement déclencheur (type) : Type d'événement qui se produit en dehors de la frontière du logiciel mesuré et qui initie un ou plusieurs types de processus fonctionnels. Dans un ensemble de Fonctionnalités Utilisateurs Requises (FUR), chaque type d'événement qui déclenche un processus fonctionnel est indivisible pour cet ensemble de FURs.

Remarque 1 : Les horloges et les événements temporels peuvent être des événements déclencheurs.

Remarque 2 : Un événement est survenu dans le passé ou pas ; il est instantané.

Fonction de mesure (COSMIC-FFP) : Fonction mathématique qui assigne une valeur numérique à sa variable sur la base de l'étalon de mesure COSMIC-FFP. La variable de la fonction de mesure COSMIC-FFP est le type de mouvement de données.

Fonctionnalité Utilisateur Requête (FUR) : L'expression « Fonctionnalité Utilisateur Requête » (FUR) réfère à un sous-ensemble des besoins de l'utilisateur. La FUR représente les pratiques et les procédures de l'utilisateur que le logiciel doit accomplir pour répondre aux besoins de celui-ci. La FUR exclut les besoins en qualité et les besoins techniques.

Frontière : Une interface conceptuelle entre le logiciel à mesurer et ses utilisateurs

Remarque : La frontière d'un morceau de logiciel est la ligne conceptuelle séparant ce morceau de l'environnement dans lequel il s'exécute, tel que perçu d'un point de vue externe par ses utilisateurs. La frontière permet à la personne qui mesure de distinguer, sans ambiguïté, ce qui est inclus dans le logiciel mesuré de ce qui fait partie de l'environnement dans lequel fonctionne ce logiciel.

Groupe de données (type) : Ensemble distinct non vide, non ordonné et non redondant de types d'attributs. Chaque type d'attribut dans un groupe de données décrit un aspect complémentaire du même objet d'intérêt (voir définition). Un groupe de données (type) est caractérisé par sa persistance (voir définition).

Input : Les données pour lesquelles la valeur est indépendante du logiciel et qui sont utilisées par le logiciel lors de son opération. La définition du terme « entrée » utilisée dans ce manuel est différente de la définition utilisée par l'International Function Point Users Group (IFPUG) pour le terme « input ». Pour COSMIC-FFP, le terme « input » comprend toutes les entrées qui participent dans un processus fonctionnel particulier.

Lecture (type) : Une lecture (L) est un type de mouvement de données qui, dans le contexte de son processus fonctionnel, déplace un groupe de données depuis sa partie de stockage persistant pour le mettre à la portée de ce processus fonctionnel (type) qui le requiert.

Remarque : Une lecture inclut certaines manipulations associées de données nécessaires à son accomplissement. Voir la section 4.1 pour plus de détails.

Mesure de la taille fonctionnelle (MTF) : Processus qui permet de mesurer la taille fonctionnelle.

Méthode de mesure : Séquence logique d'opérations, décrite de façon générique, utilisée pour réaliser la mesure.

Méthode de mesure de la taille fonctionnelle : C'est une réalisation spécifique de la MTF définie par un ensemble de règles, lesquelles sont conformes aux caractéristiques obligatoires de la norme ISO/IEC 14143-1 : 1998.

Mouvement de données (type) : C'est un composant fonctionnel de base qui déplace un ou plusieurs types d'attributs appartenant à un seul type de groupe de données.

Remarque 1 : COSMIC-FFP définit quatre types de mouvements de données : Entrée (E), Sortie (S), Lecture (L), Écriture (C).

Remarque 2 : Pour préciser, c'est l'occurrence d'un mouvement de données, non son type, qui déplace un groupe d'attributs. Cette remarque s'applique aux définitions des quatre types de mouvement de données : Entrée (E), Sortie (S), Lecture (L), Écriture (C).

Output : Les données dont la valeur dépend de l'opération du logiciel et qui sont ainsi créées ou encore modifiées par le logiciel pendant son opération. La définition de l'expression « données de sortie » utilisée dans ce manuel est différente de la définition utilisée par le International Function Point Users Group (IFPUG) pour le terme « OUTPUT ». Pour COSMIC-FFP, le terme « output » comprend toutes les sorties qui participent dans un processus fonctionnel particulier.

Persistance (d'un groupe de données) : C'est une qualité indiquant la durée temporelle de rétention d'un groupe de données dans le contexte de la FUR. Trois types de persistance sont définis : transitoire (pour la durée de vie du processus fonctionnel seulement), courte (au-delà de la durée du processus fonctionnel tant que le logiciel est opérationnel) et durable (au-delà de la durée des opérations du logiciel).

Processus de mesure : Processus visant à établir, à planifier, à réaliser et à évaluer la mesure du logiciel dans un projet général ou dans une structure de mesure organisationnelle.

Sortie (type) : Une sortie (S) est un type de mouvement de données qui déplace à travers la frontière un groupe de données d'un processus fonctionnel vers l'utilisateur qui le demande.

Remarque : Une sortie inclut aussi les manipulations associées de données nécessaires à l'exécution de la « sortie » (voir la section 4.1 pour plus de détails).

Sous-processus (type) : Un sous-processus COSMIC-FFP est un mouvement de données rencontré pendant l'exécution d'un processus fonctionnel. Il y a quatre sortes de types de mouvements de données : Entrée (E), Sortie (S), Lecture (L) et Écriture (É). Un mouvement de données comprend aussi un nombre de sous-processus de manipulation des données. Un sous-processus COSMIC-FFP est l'équivalent d'un CFB ISO. Un sous-

processus COSMIC-FFP n'exprime que la Fonctionnalité Utilisateur Requise (FUR) et exclut les requis de qualité et les requis techniques.

Stockage persistant : C'est un type de stockage qui permet à un processus fonctionnel de conserver des données au-delà de la vie du processus fonctionnel et/ou qui permet à un processus fonctionnel de retrouver une donnée stockée par un autre processus fonctionnel, ou stockée plus tôt au cours d'une occurrence du même processus fonctionnel.

Remarque 1 : Dans le modèle COSMIC-FFP, puisque la partie de stockage est du côté logiciel de la frontière, elle n'est pas considérée comme un utilisateur du logiciel mesuré.

Remarque 2 : Un autre exemple de type de stockage persistant est la mémoire à lecture seulement.

Taille fonctionnelle : La taille d'un logiciel dérivée par la quantification des Fonctionnalités Utilisateurs Requises (FUR).

UML: Acronyme de Unified Modeling Language est un langage de description normé, c'est une formalisation très aboutie et non-propriétaire de la modélisation objet utilisée en génie logiciel (Grady Booch *et al.*, 2000).

Unité (de mesure) : Grandeur particulière, définie et adoptée par convention, à laquelle on compare les autres grandeurs de même nature pour les exprimer quantitativement par rapport à cette grandeur. Les unités de mesure ont par convention des noms et des symboles correspondants.

Utilisateur : N'importe quelle personne qui spécifie des FURs et/ou n'importe quelle personne ou chose qui communique ou interagit avec le logiciel à n'importe quel moment.

Remarque 1 : Dans ce *Manuel de mesures*, le terme utilisateur est limité à la deuxième signification donnée dans cette définition, c'est-à-dire « toute personne ou chose qui communique ou qui interagit avec le logiciel à tout moment ».

Remarque 2 : Les utilisateurs peuvent être des humains, des logiciels ou des dispositifs techniques.

Valeur (d'une grandeur) : Expression quantitative d'une grandeur particulière, généralement sous la forme d'un étalon de mesure multiplié par un nombre.

CHAPITRE 1

INTRODUCTION

The royal Architect or foreman of each construction site was responsible for maintaining and transferring the unit of length to the workers, cubit sticks. It was required that the cubit sticks be brought at each full moon to be compared to the royal cubit Master. Failure to do so was punishable by death (EOS, 2001).

Selon l'organisation de standardisation EOS (2001), bien que la punition prescrite fût grave, l'ancienne civilisation égyptienne avait déjà prévu l'esprit contemporain du système de métrologie légale, des normes, de la traçabilité et du calibrage. Avec ce travail d'étalonnage et d'uniformité de longueur, ils réalisaient des mesures avec une exactitude étonnante.

Pour les premières mesures de distance, on utilisait des parties du corps de l'homme. Dès 3 000 ans avant J.-C., et suite aux échanges de biens entre nations à cette époque une mesure étalon ou standard était nécessaire, mais il a fallu attendre environ 4 800 ans pour que l'humanité ait une mesure standard « à tous les peuples et à tous les temps » (Bureau national de la métrologie, 2003) qui est le mètre.

Par la suite, d'autres mesures universelles ont vu le jour comme le kilogramme défini comme l'étalon international de mesure de la masse et la seconde qui correspond à l'unité de base de la mesure du temps.

Le quotidien sans utilisation de mesure serait difficile à concevoir. Une illustration avec une contre-analogie serait un film de science-fiction où on construirait des routes et des ponts sans prendre les mesures du processus de construction ni *a priori* ni *a posteriori*, où on bâtirait des maisons sans effectuer les calculs de mesures préalables ou ultérieures,

où on achèterait de la marchandise sans peser son poids, où on assisterait à des spectacles sans en savoir la durée, où on se procurerait des vêtements sans mesurer la taille d'avance et, pire encore, où on préparerait des médicaments sans faire le dosage des constituants!

1.1 Mise en contexte

Bien que cette illustration paraisse illogique, l'absence des mesures est bien fréquente dans le développement des logiciels et quoique le génie logiciel ait progressé au cours de la dernière décennie, les mesures sont encore peu utilisées et difficiles à appliquer dans ce nouveau domaine de l'ingénierie.

À cause de son jeune âge, il est normal que le domaine du génie logiciel soit confronté à une telle difficulté. Abran et Jacquet (1999) déclarent, dans leur article *From Software Metrics to Software Measurement Methods: A Process Model*, que « de nombreuses métriques de logiciel ont été fondées sur une approche intuitive non vérifiée et elles ne sont pas appuyées sur des fondements vérifiables. Donc, un nombre important de métriques de logiciel ne sauraient être qualifiés de méthodes de mesures ». En effet, dans plusieurs cas, le nombre élevé de mesures du logiciel proposées et non normalisées vient au contraire compliquer la situation.

De ce fait, un référentiel pour la mesure en génie logiciel avec une reconnaissance officielle internationale intéresse particulièrement l'équipe de cette recherche. L'équipe de cette recherche se compose essentiellement de l'étudiant mandataire, du directeur de la recherche et de neuf experts internationaux en mesure des logiciels.

Le grand dictionnaire terminologique (Office de la langue française, 2001) donne trois définitions du terme « référentiel » dans le domaine informatique : 1) ensemble de tous les éléments considérés en vue d'un traitement particulier; 2) ensemble de références;

et 3) ensemble universel; dans cette thèse, il est utilisé pour désigner **un ensemble de références**.

Le référentiel à développer sera composé de huit étalons de mesure pour la taille fonctionnelle des logiciels. Selon VIM (ISO, 1993), un étalon est une mesure matérialisée, un appareil de mesure, un matériau de références ou un système de mesure destiné à définir, à réaliser, à conserver ou à reproduire une unité ou une ou plusieurs valeurs de grandeur pour servir de référence :

Exemples :

étalon de masse de 1 kg;
résistance étalon de 100 Ω ;
ampèremètre étalon;
étalon de fréquence à césium;
électrode de référence à hydrogène;
solution de référence de cortisol dans le sérum humain, de concentration certifiée.

1.2 Composition de la thèse

Le chapitre 1 présente l'introduction générale de la thèse ainsi que sa composition.

Le chapitre 2 traite des mesures en génie logiciel. Il positionne la mesure dans le génie logiciel, puis il montre le besoin en mesure dans ce domaine et sa classification par rapport à ce génie. Le schéma synoptique de la figure 1 résume la structure de la thèse.

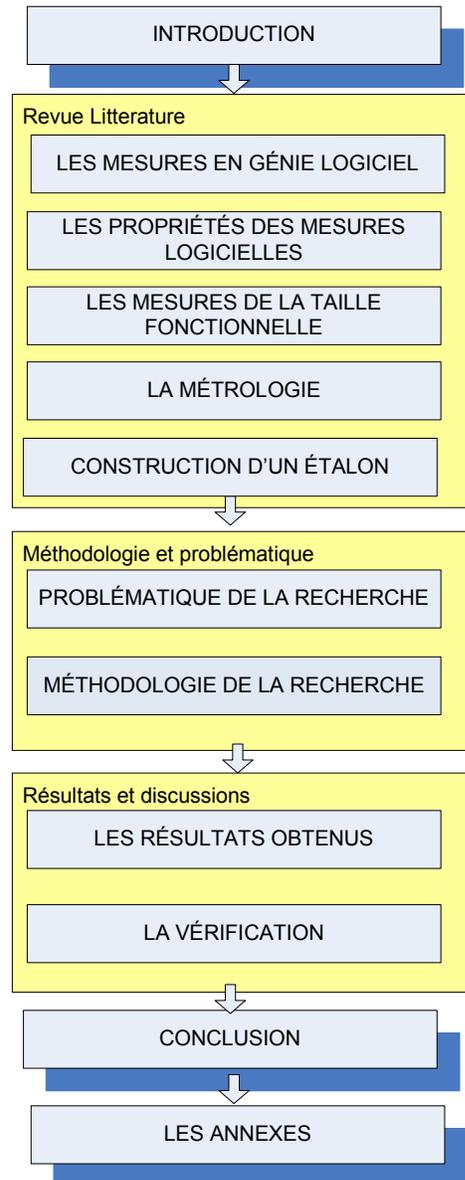


Figure 1 Présentation synoptique de la structure de la thèse

Le chapitre 3 présente un inventaire des propriétés proposées pour les mesures des logiciels et il présente des observations sur les propriétés des mesures des logiciels.

Le chapitre 4 aborde la métrologie et la standardisation en génie logiciel, incluant une présentation des définitions de métrologie, standardisation et standard. Il présente aussi

le contenu des normes, leurs rôles et leurs types. Il finit par situer la métrologie dans ISO.

Le chapitre 5 explique les mesures de la taille fonctionnelle (MTF), incluant la mesure des points de fonction. Ce chapitre présente également une comparaison entre deux types de méthodes de mesure de la taille des logiciels : les lignes de code et les points de fonction, en listant les avantages et les inconvénients de chaque type de méthode. Puis, il fait une présentation de la métanorme ISO 14143 puis des méthodes spécifiques de mesure de la taille fonctionnelle des logiciels telles que IFPUG, NESMA, MKII et COSMIC-FFP. Après, il compare les méthodes IFPUG et COSMIC-FFP. Le chapitre se termine par la présentation des raisons de la sélection de la méthode COSMIC-FFP pour la conception des étalons.

Le chapitre 6 explique le processus de la construction d'un étalon. Ce chapitre définit le concept d'étalon et donne un aperçu sur la création du premier étalon universel : le mètre. Ensuite, le chapitre présente les étalons universels et spécifiques, les organismes de normalisation, l'étalonnage et la calibration ainsi que la traçabilité des étalons.

Le chapitre 7 fait ensuite la présentation du travail de recherche. Il expose la problématique, la question de recherche et l'objectif de recherche.

Le chapitre 8 décrit la méthodologie de recherche. Il définit le projet, la motivation pour le sujet de recherche, les utilisateurs des résultats prévus de cette recherche, les outils de travail pour faire la recherche et ses limites. Après, ce chapitre présente les étapes de la méthodologie, incluant le développement d'un site web, la méthode de liaison d'une mesure au référentiel, le développement du référentiel et les partenaires de la recherche.

Le chapitre 9 présente ensuite le déroulement de la recherche, incluant la sélection des FUR, du processus de mesurage et l'intervention des experts. Le chapitre se termine par

la présentation du développement d'un site web et des personnes impliquées dans la construction des étalons.

Le chapitre 10 discute des résultats, incluant le contenu d'un étalon de mesure développé dans cette recherche et présente un exemple détaillé d'étalon : le *Valve Control System Etalon*. Cet étalon est classé troisième dans l'ensemble de huit étalons de mesure. L'ordre des étalons de mesure est fait selon la chronologie de leur réalisation dans cette recherche.

Le chapitre 11 présente un référentiel pour la MTF conçu dans cette thèse ainsi que, les résultats trouvés. Tous les étalons y sont présentés de façon sommaire. Puis le chapitre explique l'affichage sur le web des étalons et l'évaluation des résultats.

Le chapitre 12 présente ensuite le processus de la vérification des étalons, incluant la détermination du choix et des rôles des experts, les hypothèses de vérification, de conception et de vérification. Le chapitre se termine par la présentation du déroulement de la vérification et sa portée.

Le chapitre 13 est le dernier chapitre et présente une conclusion générale et suggère des pistes de recherches futures qui pourraient découler de ce travail.

À la fin de la thèse, les références utilisées pour réaliser ce travail et les Annexes sont présentées. Les annexes ont pour but d'illustrer plus en détail les étalons de mesures de la taille fonctionnelle des logiciels proposés dans cette recherche. Ces étalons sont rédigés en anglais car plusieurs partenaires de ce travail ne parlent pas la langue de Molière.

L'annexe 1 décrit le premier étalon de mesure appelé *Automatic Line Switching Etalon*. Les FUR du logiciel *Automatic Line Switching* sont documentées dans le rapport technique ISO/IEC TR 14143-4 (Version 2000). L'*Automatic Line Switching* utilisé,

correspond à l'ensemble RUR B.8 du document ISO14143-4. Cet ensemble RUR B.8 documente les FUR d'un système de commutation, en temps réel, entre une ligne de travail et une ligne de secours qui sont données pour une chaîne de transmission : si la ligne de travail se dégrade ou échoue, la ligne de secours est utilisée à sa place.

L'annexe 2 décrit le deuxième étalon de mesure appelé *SAVAT Gateway System Etalon*. Les FUR du logiciel *SAVAT Gateway System* sont documentées dans le rapport technique ISO/IEC TR 14143-4 (Version 2000). Le *SAVAT Gateway System* utilisé, correspond à l'ensemble RUR B.10 du document ISO14143-4. Cet ensemble RUR B.10 documente les FUR d'un système qui fournit une interface vidéotex au public pour commander des produits de la compagnie SAVAT.

L'annexe 3 décrit le quatrième étalon de mesure appelé *Hotel Reservation System Etalon*. Le troisième étalon est présenté au Chapitre 9 pour illustrer le contenu d'un étalon dans cette recherche. Les FUR du logiciel *Hotel Reservation System* sont documentées dans le rapport technique ISO/IEC TR 14143-4 (Version 2000). Le *Hotel Reservation System* utilisé correspond à l'ensemble RUR A.1 du document ISO14143-4. Cet ensemble RUR A.1 documente les FUR d'un système de réservation qui fait partie d'un système général de logement d'hôtel. Le système de réservation offre la maintenance et la confirmation des réservations.

L'annexe 4 décrit le cinquième étalon de mesure appelé *L-Euchre Card Game System Etalon*. Les FUR du logiciel *L-Euchre Card Game System* sont documentées dans le rapport technique ISO/IEC TR 14143-4 (Version 2000). Le *L-Euchre Card Game System* utilisé, correspond à l'ensemble RUR B.11 du document ISO14143-4. Cet ensemble RUR B.11 documente les FUR d'un système en temps réel. Le *L-Euchre* est un jeu de cartes d'euchre de quatre joueurs. Le jeu relie à distance quatre joueurs à travers Internet par le protocole de TCP/IP. Le même noyau de jeu doit s'exécuter sur l'ordinateur de chaque joueur. Un joueur est choisi pour contrôler le jeu et les trois joueurs restants se connectent au jeu par l'adresse IP du serveur central.

L'annexe 5 décrit le sixième étalon de mesure appelé *Rice Cooker System Etalon*. Les FUR du logiciel *Rice Cooker System* sont documentées dans son rapport de spécifications et d'exigences. Ces exigences et ces spécifications documentent les FUR d'un système de contrôle de cuiseur de riz.

L'annexe 6 décrit le septième étalon de mesure appelé *C-Registration System Etalon*. Les FUR du logiciel *C-Registration System* sont rapportées dans la documentation de formation du processus RUP. Cet ensemble documente les FUR d'un système qui permet aux étudiants de s'inscrire aux cours en ligne. Le système permet aussi aux professeurs de choisir les cours à enseigner et de mettre à jour les notes des étudiants.

L'annexe 7 décrit le huitième étalon de mesure appelé *Collegiate Sports Paging System Etalon*. Les FUR du logiciel *Collegiate Sports Paging System* sont rapportées dans la documentation de formation du processus RUP. Cet ensemble documente les FUR d'un système qui permet aux abonnés d'être avisés des événements collégiaux ou des événements des équipes de sports auxquels ils sont inscrits. Le système permet aussi aux abonnés de regarder le contenu de ces événements.

L'annexe 8 résume la bibliographie de quelques experts qui ont participé à cette recherche.

Enfin, l'annexe 9 présente l'autorisation fournie à l'étudiant mandataire par l'équipe de RUP pour utiliser la documentation de *Course Registration System* et le *Collegiate Sports Paging System*.

CHAPITRE 2

LES MESURES EN GÉNIE LOGICIEL

«L’homme est la mesure de toute chose» (Protagoras, 485 av. J.-C.)

Ce chapitre propose une revue de la littérature sur les mesures en génie logiciel en présentant brièvement la place de la mesure dans le développement du logiciel et les besoins en mesure des différents acteurs dans le cycle de développement. Dans ce chapitre, l’emphase est mise sur la classification de la mesure en se basant sur les classifications proposées par plusieurs chercheurs du domaine. Le chapitre montre, en particulier, la classification des mesures des logiciels suggérées par Fenton. Un aperçu sur les échelles de mesure est présenté à la fin.

2.1 Introduction

La mesure est une partie intégrante de toute activité humaine : sociale, économique, industrielle, académique, environnementale, médicale, etc. Présente dans toutes nos activités quotidiennes pour avoir une vision objective sur la qualité, la mesure est devenue une pierre angulaire du développement industriel, scientifique et social. L'apport de la métrologie (e.g. la science de la mesure) dans le développement du commerce international et la réduction des barrières techniques aux échanges des biens ne sont plus à discuter.

2.2 Place de la mesure en génie logiciel

Il y a trente ans, la mesure des logiciels était une réflexion confinée à quelques chercheurs universitaires et organisations industrielles. Le premier article sur la mesure du logiciel est probablement celui de Rubey et al. (1968). Depuis la fin des années 1970,

la mesure en génie logiciel est recommandée pour planifier et pour contrôler les propriétés du développement des logiciels et des projets logiciels.

Par exemple, Tom DeMarco, un des pionniers de la mesure en génie logiciel, indique «You cannot control what you cannot measure» (T. DeMarco, 1982). Ainsi, dans le domaine du génie logiciel, la mesure de logiciels a été introduite pour contrôler les coûts, améliorer la qualité et livrer le produit logiciel à temps.

Depuis quelques décennies, plusieurs chercheurs tels que Fenton (1997), Pfleeger (1997) et Zuse (1998) voient la mesure de logiciel comme une mise en correspondance entre le système empirique et le système formel (voir figure 2 suivante).

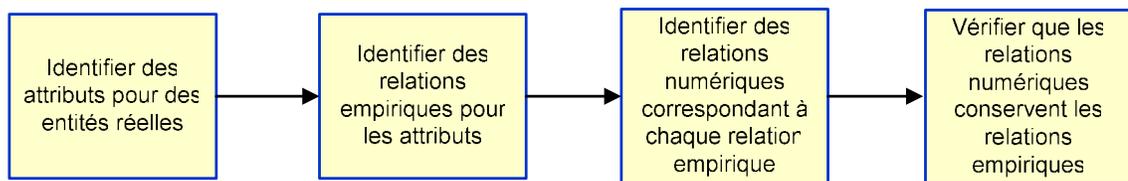


Figure 2 Étapes de la définition des mesures numériques (Fenton et al. 1997)

Pour Fenton, la mesure est une relation empirique obtenue par un consensus lorsqu'elle s'applique au monde réel :

«When the two people being compared are very close in height, we may find a difference of opinion; you may think that Jack is taller than Jill, while we are convinced that Jill is taller than Jack. Our empirical relations permit this difference by requiring only a consensus of opinion about relationships in the real world. A (binary) empirical relation is one for which there is a reasonable consensus about which pairs are in the relation» (N. Fenton et S.L. Pfleeger, 1997).

L'auteur définit le monde empirique de la taille du logiciel comme une composition d'artefacts, de livrables et de documents qui résultent du processus de construction du logiciel et qui proviennent des spécifications, des designs, des codifications et des tests.

Le monde formel est composé des conventions ou des règles de mesure du logiciel à travers ses artefacts, ses livrables et ses documents. C'est à partir de là que l'on peut définir la mesure de la taille du logiciel.

Selon Fenton et Pfleeger (1997) les objectifs principaux de la mesure de logiciels sont :

1. La compréhension : l'évaluation des attributs d'un logiciel permet de comprendre ce qui s'est passé durant le développement et la maintenance de ce logiciel. Cette compréhension permet d'éviter les erreurs commises antérieurement et de profiter des bénéfices de cette expérience dans un projet logiciel futur. Cela pourrait améliorer les processus de développement et de maintenance afin de produire des logiciels de qualité très élevée avec des coûts raisonnables.
2. Le contrôle : la prédiction des attributs d'un projet logiciel permet de contrôler son état d'avancement. Par exemple, les modèles d'estimation des coûts de développement de logiciels permettent de prédire le coût d'un logiciel à une étape assez précoce dans son cycle de vie et, ceci afin de bien gérer la répartition du coût estimé sur les différentes phases de développement.
3. L'amélioration : l'ajout de nouvelles révisions ou de nouveaux types de révision, basées sur les mesures du logiciels, pourrait améliorer le processus de développement du logiciel ou les produits.

Toujours selon Fenton et Pfleeger, une mesure concerne un attribut d'une entité logicielle. On distingue pour chaque entité :

1. des attributs internes : dépendent de l'entité seulement.
2. des attributs externes : dépendent de l'entité dans un environnement ou contexte donné.

Les attributs externes sont plus difficiles à mesurer cependant les attributs internes sont souvent mesurés indirectement. La taille est un exemple d'un attribut interne pour l'entité spécification du produit. La « maintenabilité » est un autre exemple d'un attribut externe pour l'entité spécification du produit.

Mais souvent dans la littérature ces propositions dites de « mesure » du logiciel se limitent à des concepts mathématiques avec représentations numériques sans faire appel

à un système de référence ni à une instrumentation de mesure ou aux protocoles de mesure.

Or, dans la plupart des disciplines de génie, c'est le domaine de la « métrologie » qui est la base du développement et du design des instruments et des processus de mesure. Selon le BIPM (Bureau international des poids et mesures, qui se trouve à Paris, a pour mission d'assurer l'uniformité mondiale des mesures et leur « traçabilité » au Système international d'unités - SI), « La métrologie est la science de la mesure; elle embrasse à la fois les déterminations expérimentales et théoriques à tous les niveaux d'incertitude et dans tous les domaines des sciences et de la technologie » (BIPM, 2005).

Avec l'introduction des concepts de la métrologie tels que des modèles théoriques et des instruments de mesure, la mesure des logiciels pourrait permettre d'obtenir des informations de plus en plus pertinentes nécessaires pour prendre les décisions influant la gestion et la performance des projets.

2.3 Le besoin en mesure dans le génie logiciel

D'après Zuse (1998), le concept de la mesure est évoqué d'une manière ou d'une autre dans la plupart des conférences en génie logiciel. Zuse mentionne que le besoin en mesure est reconnu dans la définition même du génie logiciel dans IEEE Std 610.12 (1990) : «The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software». L'auteur dit que la définition du mot « quantifiable » illustre qu'au moins une organisation professionnelle internationale a reconnu la mesure comme partie intégrale d'une approche bien conçue de logiciel, et non pas une simple adjonction.

La mesure des logiciels est essentielle dans plusieurs aspects comme le contrôle de la qualité et l'estimation du coût des logiciels. Giles (1995) résumait les avantages de la mesure en génie logiciel comme suit :

- permet le contrôle des produits et des processus;
- démontre la productivité, l'efficacité et la qualité du travail;
- attire le respect du client et la crédibilité en gestion;
- identifie où des améliorations peuvent être effectuées.

Quoiqu'il existe beaucoup de propositions de mesures en génie logiciel, il y a encore un grand nombre de questions sur l'utilité de ces propositions. Par exemple, il y a plus de vingt ans que Brown et al. (1981) ont signalé que le manque des principes de base dans le génie logiciel rend la modélisation des relations entre les variables et la spécification des systèmes de mesure difficiles.

De plus, l'absence d'étalons de mesures pour les logiciels rend la tâche des mesureurs de logiciels délicate en termes de précision et de « traçabilité » des résultats de mesure obtenus. Selon le Vocabulaire International de Métrologie - VIM (ISO, 1993) « un étalon est défini comme mesure matérialisée, appareil de mesure, matériau de référence destiné à définir, réaliser, conserver ou reproduire une unité ou une ou plusieurs valeurs d'une grandeur pour servir de référence ». La figure 3 suivante présente un exemple d'étalon.



Figure 3 Lampe d'intensité lumineuse

« Cette lampe, d'apparence un peu désuète et pour le moins inhabituelle, matérialise et conserve l'une des sept unités de base du Système International : la candela. À l'heure du laser et du semi-conducteur, cet instrument reste associé à ces nouvelles techniques dans les mesures photométriques où sa fiabilité et sa simplicité en font un outil métrologique de référence » (ISO, 1993).

2.4 La classification des mesures en génie logiciel

Selon Zuse (1998), il y avait déjà en 1998 à peu près 1 500 différentes mesures des logiciels dont plus de 200 pour le langage de programmation orienté objet. La classification la plus utilisée pour ces mesures est celle de Fenton *et al.* (1995). Dans sa classification, Fenton distingue trois classes de mesures (voir figure 4 suivante) :

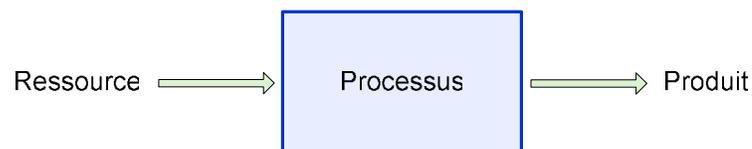


Figure 4 Classification de Fenton (1997)

1. Ressources : ce sont les objets qui contribuent au processus (exemple : expérience du personnel, degré de communication dans l'équipe, prix des progiciels, performance de l'équipement);
2. Processus : ce sont les activités liées au développement logiciel et comprenant normalement un facteur de temps (exemple : effort de construction du logiciel, nombre de changements pour une période donnée, durée de construction du logiciel);
3. Produits : ce sont les objets livrables qui sont le résultat du processus (exemple : taille de l'application, complexité des algorithmes, qualité du produit).

Dans la littérature, quand on parle de la mesure on évoque quelques chercheurs en mesure des logiciels, comme Zuse (1998), qui ont discuté du concept d'« Unité » dans le processus de mesure. ISO (1993) définit l'unité de mesure comme une « grandeur particulière, définie et adoptée par convention, à laquelle on compare les autres grandeurs de même nature pour les exprimer quantitativement par rapport à cette grandeur ». Dans cette définition, on indique qu'une grandeur particulière est définie et adoptée par convention. Ceci implique que dans toute activité de mesure, il y aura des règles à suivre.

Fenton écrit (1995) :

« In any measurement activity, there are rules to be followed. The rules help us to be consistent in our measurement, as well as providing a basis for interpreting data. Measurement theory tells us the rules, laying the groundwork for developing and reasoning about all kinds of measurement. »

Le même auteur voit que ceci n'est pas unique au seul domaine du génie logiciel :

«This rule-based approach is common in many sciences. »

La plupart des mesures des logiciels manquent aussi d'une échelle d'interprétation comme par exemple la température de congélation de l'eau qui est 0 degré Celsius, et de son ébullition qui est 100 degrés Celsius. En fait, on ne peut pas évoquer la mesure sans le concept d'unité ou d'une échelle d'interprétation ou au moins d'un intervalle de

rapports. Pourtant, c'est ce qui se passe en génie logiciel où la définition de l'unité est rarement abordée.

Enfin, puisque cette recherche traite de la mesure, il est préconisé d'apporter ici un bref rappel sur les échelles de mesure. Les exemples de mesure les plus utilisés sont celui de Stevens (1946). Il a défini cinq types d'échelles : nominale, ordinale, intervalle, ratio et absolue.

L'échelle nominale place les éléments par catégorie, sans ordre. L'échelle ordinale est souvent utile pour ajouter à l'échelle nominale des informations sur l'ordre des catégories. L'échelle par intervalle donne plus d'informations pour les entités que les échelles nominale et ordinale. Cette échelle définit la taille de l'intervalle qui sépare les catégories. La caractéristique clé qui distingue l'échelle ratio des échelles nominale, ordinale et intervalle est l'existence de relations empiriques reflétant les ratios. Par l'échelle ratio, on peut dire que la période de développement d'un projet est deux fois plus longue qu'un autre. Enfin, l'échelle absolue est la plus restrictive. Par exemple, le comptage des employés d'une usine ne se fait que d'une seule façon. L'unicité de la mesure est une différence importante entre les échelles absolue et ratio.

Ces types d'échelles sont hiérarchiquement ordonnés de la moins granulaire à la plus granulaire. La détermination du type d'échelle est très importante car elle met en évidence les propriétés empiriques associées à l'attribut mesuré. Par exemple, on ne peut avoir une échelle de type Ratio que pour les attributs possédant des relations empiriques qui utilisent des opérations de rapport telles que n fois.

Le type d'échelle détermine les opérations statistiques applicables sur les valeurs de l'échelle. Par exemple, la moyenne arithmétique nécessite que l'échelle soit au minimum du type intervalle. Le tableau 1 (A. Abran, 1994) présente un résumé des types d'échelles de mesures.

Tableau I

Échelles de mesure (A. Abran, 1994)

Types d'échelle	Représentation	Transformations admissibles	Description des opérations	Exemples
Nominale	(R,=)	F est une correspondance de 1 à 1	Nommer, identifier	Couleurs, formes
Ordinale	(R,>=)	F fonction monotone strictement croissante	Ranger, ordonner	Préférence, dureté
Intervalle	(R,>=,+)	$F(x) = ax + b, a > 0$	Additionner	La date du calendrier, température Celsius
Ratio	(R,>=,+)	$F(x) = ax, a > 0$	Additionner, multiplier, diviser	Masse, distance, température Kelvin
Absolue	(R,>=,+)	$F(x) = x$	Additionner, multiplier, diviser	Compter les entités

En 1998, Zuse présentait plusieurs indicateurs qui exigent le concept d'unité dans le processus de mesure des logiciels, par exemple :

1. Lignes source
2. Personne mois
3. Pages par mois
4. Mois
5. Cas de test par jour
6. Lignes de code
7. Effort
8. Coût de la maintenance
9. Nombre de défauts
10. Nombre de boucles
11. Nombre de modules

Dans son livre *A Framework for Software Measurement*, Zuse présente une liste illustrative des propositions de mesure des logiciels qui sont :

1. Lignes de code (LOC);
2. Mesures de Halstead pour la prédiction de l'effort de maintenance;
3. Mesures de McCabe pour la complexité cyclomatique;
4. Mesures d'Oviedo pour le calcul des complexités du contrôle et des données;
5. Mesure Défaut–Densité (très utilisée en industrie). Les défauts sont ceux qui apparaissent après une période de livraison, par exemple six mois;
6. Mesures d'Henry et al. pour l'analyse de la complexité totale d'un système informatique;
7. Points de fonction pour la taille fonctionnelle.

2.5 Sommaire

Ce chapitre a présenté un aperçu sur la place de la mesure dans le développement du logiciel. La classification de la mesure est illustrée dans ce chapitre en se basant sur les classifications proposées par plusieurs chercheurs du domaine. À la fin, la classification des mesures des logiciels suggérées par Fenton a été exposée. Cette première revue de littérature a permis d'avoir une idée sur les mesures des logiciels et planifier les lectures à faire afin d'aborder le sujet de cette recherche.

Le prochain chapitre présente les propriétés des logiciels discutées par plusieurs chercheurs du génie logiciel pour les prendre en considération pendant la conception des étalons.

CHAPITRE 3

PROPRIÉTÉS DES MESURES EN GÉNIE LOGICIEL

Ce chapitre présente 70 propriétés des mesures de logiciels proposées par des chercheurs du domaine ainsi que six autres propriétés recommandées par la norme ISO 14143. Le chapitre présente une vue sur ces propriétés et leur utilité dans la conception des étalons.

3.1 Introduction

Dans son livre *A Framework for Software Measurement*, Zuse (1998) présente les raisons de formuler les propriétés des mesures des logiciels : *The reason for formulating properties of software measures is, among others, to provide a standard of software measures*. En effet, pour assurer la vérification des mesures, plusieurs chercheurs ont proposé des ensembles de propriétés, parfois appelés *axiomes*, que la mesure à vérifier devrait satisfaire. Mais, dans la littérature en génie logiciel, il n'y a pas de consensus, ce qui se traduit par un ensemble incohérent de propriétés, du fait que les attributs à mesurer peuvent avoir différentes interprétations. Par exemple, certains axiomes de Weyuker (1988) exigent que l'attribut de la complexité du logiciel soit défini comme étant sa compréhensibilité; pour d'autres chercheurs, tel que McCabe (1976), la complexité est liée à la testabilité du code source. Cette diversité d'opinions et cette absence de consensus rendent les utilisateurs potentiels de ces mesures sceptiques et les travaux pour normaliser les mesures elles-mêmes difficiles ainsi que le design et le choix des étalons de mesure une tâche ardue.

3.2 Inventaire des propriétés proposées pour les mesures des logiciels

Zuse présente dans son livre *A Framework of Software Measurement* un inventaire de 70 propriétés des mesures des logiciels désirables proposées par plusieurs chercheurs, qui

sont rapportées comme telles en langue anglaise dans les tableaux suivants pour respecter leur définition d'origine.

Les propriétés des mesures de logiciels proposées par Bache (1990) sont présentées au tableau 2.

Tableau II

Les propriétés des mesures de logiciels proposées par Bache

No	Nom de la propriété	Définition de la propriété
1	Axiom of Positivity	Positivity means adding something increases the complexity.
2	Sequence is more Complex Than the Maximum of the Components	It says that the whole, consisting of a sequence, is more complex than the maximum of the components.
3	Weak Commutativity	Is defined as $u(P1 \circ P2) = u(P2 \circ P1)$ for all $P1, P2 \in P$, where P is, for example, a set of flowgraphs and u is a measure.
4	Strongest Independence Substitution Condition	
5	Invariance of Nesting	
6	Dominance of the Relation $H > G$ in Nested Flowgraph I	
7	Dominance of the Relation $H > G$ in Nested Flowgraph I	
8	Nested Structures are more Complex than Sequences	
9	Dominance of a more Complex Component	
10	VINAP Measures	Measures control flow complexity.

Les propriétés des mesures de logiciels identifiées par Conte et al. (1986) sont présentées au tableau 3.

Tableau III

Les propriétés des mesures de logiciels identifiées par Conte et al.

No	Nom de la propriété	Définition de la propriété
1	Simplicity	Does the measure lead to a simple result? The measure should produce a single value.
2	Robustness	Is the measure sensitive to the artificial manipulation of some factors that do not affect the performance of the software?
3	Predescriptiveness	Can the measure be used to guide the management of software development or maintenance?
4	Analyzeability :	Can the value of the measure be analyzed using standard statistical tools?

IEEE (1993) a publié son standard 1061 *Quality Metrics Methodology* qui contient des directives pour la validation des mesures de logiciel. Les propriétés des mesures de logiciels exigées par IEEE sont présentées au tableau 4.

Tableau IV

Les propriétés des mesures de logiciels exigées par IEEE

No	Nom de la propriété	Définition de la propriété
1	Tracking	If a metric M is directly related to a quality factor F , for a given product or process, then a change in a quality factor value from F_{T1} to F_{T2} at times $T1$ and $T2$, shall be accomplished by a change in metric value from M_{T1} to M_{T2} , which is the same direction.
2	Consistency	If factor values F_1, F_2, \dots, F_n , corresponding to products or processes 1,2, ..., n, which have the relationship $F_1 > F_2 > F_n$, then the corresponding metric values shall have the relationship $M_1 > M_2 > M_n$.
3	Discriminative power	A metric shall be able to discriminate between high quality software components and low quality software components. For instance, the set of metric values associated with the former should be significantly higher than those with latter.

Tableau IV (Suite)

Les propriétés des mesures de logiciels exigées par IEEE

No	Nom de la propriété	Définition de la propriété
4	Reliability	A metric shall demonstrate the above correlation, tracking, consistency, predictability, and discriminative power properties for P percent of the application of the metric.

Les propriétés des mesures de logiciels exigées par Basili et Reiter (1979) sont présentées au tableau 5.

Tableau V

Les propriétés des mesures de logiciels exigées par Basili et Reiter

No	Nom de la propriété	Définition de la propriété
1	Sensitivity	The measure should be sensitive to externally observable differences in the development environment.
2	Intuitiveness	The relative values of the measure for specific environments should correspond to some intuitive notion about the characteristic differences for those environments.

Les propriétés des mesures de logiciels exigées par Ejiogu (1991) sont présentées au tableau 6.

Tableau VI

Les propriétés des mesures de logiciels exigées par Ejiogu

No	Nom de la propriété	Définition de la propriété
1	Empirically and Intuitive Persuasive	The empirical behaviour of the target being measured should dictate its manner of measurement.

Tableau VI (Suite)

Les propriétés des mesures de logiciels exigées par Ejiogu

No	Nom de la propriété	Définition de la propriété
2	Simple and Computable	The measure should be convenient to teach and use by the general practitioner or student; should require only simple and well-formed formulas.
3	Consistent and Objective	An independent observer or practitioner should be able to confirm the same measure using the same formula or guidelines.
4	Measure Rationalism	Must belong to the class of mathematical functions called measure functions; only these satisfy the well-known postulates of measure functions.
5	Consistency of Units and Dimensions	Must satisfy the mathematical validity of units and dimensions; combinations of target measures must only be made in admissible mathematical fashions.
6	Programming language independence	Each measure should be expressive of the software structure and not dependent on the “internals” of any programming language; in other words, the formulas should be invariant with respect to every programming language.
7	Feedback Effect	Essentially, the measure should be used to reinforce good programming habits and expedite debugging.

Les propriétés des mesures de logiciels exigées par Fenton sont présentées au tableau 7.

Tableau VII

Les propriétés des mesures de logiciels exigées par Fenton

No	Nom de la propriété	Définition de la propriété
1	Sequential Determinism	The complexity of a sequential flowgraph should be uniquely determined by the complexities of the components.
2	Non-Sequential Determinism	The complexity of a non-sequential flowgraph should be determined by both, the complexity of the components and the complexity of the non-sequential structure.
3	Linear Combination	If a measure satisfies a proper axiom scheme then any positively weighted linear combination also satisfies that axiom scheme.

Les propriétés des mesures de logiciels exigées par Jones (1994) sont présentées au tableau 8.

Tableau VIII

Les propriétés des mesures de logiciels exigées par Jones

No	Nom de la propriété	Définition de la propriété
1	Standard Definition	The metric should have a standard definition and be unambiguous.
2	Large scale statistics	The metric should not be biased and unsuited for large scale statistical studies.
3	Formal User Group	The metric should have a formal user group and adequate published data standard definition and be ambiguous.
4	Measures and Tools	The metric should be supported by tools and automation.
5	Conversion Rules Between Measures	It is helpful to have conversion rules between the metric and other metrics.
6	Not Code but Software Deliverables	The metric should deal with all software deliverables, and not just code.
7	Support all Types of Software Projects	The metric should support all types of software projects.
8	Support all Types of Programming Languages	The metric should support all types of Programming Languages.

Les propriétés des mesures de logiciels exigées par Kearny et al. (1986) sont présentées au tableau 9.

Tableau IX

Les propriétés des mesures de logiciels exigées par Kearny et al.

No	Nom de la propriété	Définition de la propriété
1	Robustness	If software complexity measures are to be used to evaluate programs, then it is important to consider the measures responsiveness to program modifications.
2	Normativeness	The interpretation of complexity measurements is facilitated if the metrics provide a norm against which measurements can be compared.
3	Specificity	Software complexity analysis may provide an assessment tool that can be used during program development and testing.
4	Predescriptive-ness	If software complexity measures are to prove useful in the containment of program complexity, then they must not only index the level of program's complexity, but also should suggest methods to reduce the amount of complexity.
5	Property Definition	The properties identified above are not rigorously defined, and it is sometimes difficult to tell whether or not a measure possesses one or another of these properties. Although a list of properties

Dans son ouvrage, Zuse (1998) confirmait que les propriétés des mesures de logiciel suggérées par Weyuker (1988) sont désirées, largement connues et utilisées pour l'évaluation des mesures de logiciel. Il réaffirmait que l'avantage est leur présentation claire qui leur permet d'être discutées facilement. Ces propriétés sont présentées au tableau 10.

Tableau X

Les propriétés des mesures de logiciel suggérées par Weyuker

No	Nom de la propriété	Définition de la propriété
1	Basic Assumption of a Measure	A measure should distinguish between at least two programs P and Q in at least one case.

Tableau X (Suite)

Les propriétés des mesures de logiciel suggérées par Weyuker

No	Nom de la propriété	Définition de la propriété
2	Finitely many Identifiers	Let c be a non-negative number, then there are only finite many programs of complexity c .
3	Equivalence Classe	There are programs P and Q such that $u(P) = u(Q)$, where u is a measure.
4	Same Functionality, but different Complexity	Even though two programs the same function, it is the details of the implementation that determine the program's complexity.
5	Weak Positivity	The measure should be sensitive to adding something.
6	Rejection of the Weakest Independence Condition C1	Programs are objects composed from simpler programs (or more properly program bodies). Thus it is important to consider the relative complexities of program bodies. The Condition C1 means if a program $P1$ is equal to program $P2$ it implies that $P1$ concatenated with P are equal to $P2$ concatenated with P . The implication is irreversible.
7	Weak Commutativity	Program complexity should be responsive to the order of statements and hence the potential interaction among statements.
8	Renaming	If P is a renaming of Q , then $u(P) = u(Q)$. This property states that uniformly changing variable names should not affect a program's Complexity.
9	Wholeness	The whole must be at least as big as the sum of the parts.

Les propriétés des mesures de logiciels exigées par Lakshmanan et al. (1991) sont présentées au tableau 11.

Tableau XI

Les propriétés des mesures exigées par Lakshmanan et al.

No	Nom de la propriété	Définition de la propriété
1	Non-negativity	If the program only contains sequential code (referred to as a basic block B) then $\text{Complexity}(\text{SB}) = 0$. If the program X is not a basic block, then $\text{Complexity}(\text{SX}) > 0$
2	Functional independence under sequencing	$\text{Complexity}(\text{SX};\text{B}) = \text{Complexity}(\text{SX})$ $\text{Complexity}(\text{SX};\text{B}) = \text{Complexity}(\text{SX}) + \text{Complexity}(\text{SB}) = \text{Complexity}(\text{SX})$
3	Symmetry under sequencing	$\text{Complexity}(\text{SX};\text{Y}) = \text{Complexity}(\text{SY};\text{X})$ $\text{Complexity}(\text{SX};\text{Y}) = \text{Complexity}(\text{SX}) + \text{Complexity}(\text{SY}) = \text{Complexity}(\text{SY};\text{X})$
4	Monotonicity under sequencing	$\text{Complexity}(\text{SX};\text{Y}) < \text{Complexity}(\text{SX};\text{Z})$ if $\text{Complexity}(\text{SY}) < \text{Complexity}(\text{SZ})$ $\text{Complexity}(\text{SX};\text{Y}) = \text{Complexity}(\text{SX};\text{Z})$ if $\text{Complexity}(\text{SY}) = \text{Complexity}(\text{SZ})$
5	Additivity under sequencing	$\text{Complexity}(\text{SX};\text{Y}) = \text{Complexity}(\text{SY}) + \text{Complexity}(\text{SX})$
6	Functional independence under nesting	Adding a basic block B to a system X through nesting does not increase its complexity $\text{Complexity}(\text{SB}@\text{X}) = \text{Complexity}(\text{SX})$
7	Monotonicity under nesting	$\text{Complexity}(\text{SY}@\text{Xi}) < \text{Complexity}(\text{SZ}@\text{Xi})$ if $\text{Complexity}(\text{SY}) < \text{Complexity}(\text{SZ})$
8	Monotonicity under nesting	$\text{Complexity}(\text{SY}) < \text{Complexity}(\text{SY}@\text{X})$
9	Sensitivity to nesting	$\text{Complexity}(\text{SX};\text{Y}) < \text{Complexity}(\text{SY}@\text{X})$ if $\text{Complexity}(\text{SY}) > 0$

Les propriétés des mesures de logiciels exigées par Watts (1987) sont présentées au tableau 12.

Tableau XII

Les propriétés des mesures de logiciels exigées par Watts

No	Nom de la propriété	Définition de la propriété
1	Comparability	The measure must be comparable with other measures of the same criteria
2	Economy	The simpler, and therefore, the cheaper the measure is to use, the better
3	Objectivity	The results should be free from subjective influences. It must not matter who the measurer is
4	Reliability	The result should be precise and repeatable
5	Standardisation	The measure should be unambiguous and allow for comparison
6	Usefulness	The measure must address a need, not simply measure for its own sake
7	Validity	The metric should measure the right attribute

Navlaka (1986) exige qu'une bonne mesure ait les propriétés suivantes (voir tableau 13 suivant).

Tableau XIII

Les propriétés des mesures de logiciels exigées par Navlaka

No	Nom de la propriété	Définition de la propriété
1	Correctness	From the same data and rules, the same results must always be obtained.
2	Repetitiveness	It doesn't matter the person who makes the measurement, results must always be exactly the same at different time.

Les propriétés des mesures de logiciels définies par ISO TR 14143-3 (2002) sont présentées au tableau 14.

Tableau XIV

Les propriétés des mesures de logiciels définies par ISO 14143-3

No	Nom de la propriété	Définition de la propriété
1	Répétitivité	L'étroitesse de l'accord entre les résultats de mesurages successifs du même mesurande, mesurages effectués dans la totalité des mêmes conditions de mesure (AFNOR, 1994).
2	Reproductibilité	L'étroitesse de l'accord entre les résultats des mesurages du même mesurande, mesurages effectués en faisant varier les conditions de mesure (AFNOR, 1994).
3	Exactitude	L'étroitesse de l'accord entre les résultats d'un mesurage et une valeur vraie d'un mesurande.
4	Convertibilité	La mesure est convertible en d'autres mesures
5	Seuil de Tolérance	
6	Applicabilité aux domaines fonctionnels	

3.3 Constatation sur les propriétés des mesures des logiciels

D'après l'inventaire ci-dessus des propriétés exigées par les chercheurs en mesure des logiciels, neuf propriétés seulement ont été recommandées par plus d'un groupe de chercheurs. Ce sont la standardisation, la simplicité, l'uniformité, l'objectivité, l'indépendance du langage de programmation, la fiabilité, la robustesse, la pre-description et l'intuition. Il est important de noter que la propriété « standardisation » est la plus exigée, ce qui rejoint l'avis de l'équipe de cette recherche et renforce la motivation de développer un référentiel afin d'améliorer la standardisation des mesures des logiciels.

La majorité des propriétés des mesures de logiciels exigées par Ejiogu sont citées par d'autres chercheurs. Les propriétés données par Watts et Navlaka attirent plus l'attention

de l'équipe de cette recherche car elles sont plus près de sa compréhension de la mesure des logiciels.

Pour la construction du référentiel, cet inventaire est à considérer et en particulier les propriétés de mesure proposées par la norme ISO 14143-3 qui sont la répétabilité, la reproductibilité, l'exactitude, la convertibilité, le seuil de discrimination et l'applicabilité aux domaines fonctionnels, pour donner plus de crédibilité à ce travail.

3.4 Les étalons de mesure

Parmi toutes les propriétés citées, une en particulier a été choisie pour la problématique de recherche choisie pour cette thèse, soit la standardisation des mesures. Une définition récente d'un étalon de mesure est présentée par (Antoine *et al.*, 2003) : « Un étalon est un type de grandeur qui sert à déterminer ou à représenter matériellement l'unité de mesure, éventuellement l'un de ses multiples. Un étalon doit être précis, exact, reproductible et universel ». Cette recherche, toutefois adopte la définition reconnue par la communauté internationale de la normalisation en mesure, soit celle définie dans le *Vocabulaire international des termes fondamentaux et généraux de métrologie* : « Un étalon est défini comme mesure matérialisée, appareil de mesure, matériau de référence destiné à définir, réaliser, conserver ou reproduire une unité ou une ou plusieurs valeurs d'une grandeur pour servir de référence. » (ISO, 1993).

Selon le dictionnaire de sciences économiques, la standardisation est « la fixation de normes rigoureuses pour la fabrication de produits ou de leurs composants, qui deviennent ainsi tous identiques » (2003); la standardisation débouche sur le fait que les produits, parce qu'ils sont identiques, sont interchangeables. Lorsque l'on étudie certaines caractéristiques des mesures des logiciels sur deux types de logiciels distincts, les différences observées peuvent être affectées par des facteurs qui différencient les deux logiciels (par exemple, l'environnement de développement et l'expérience des développeurs peuvent être différents d'un logiciel à l'autre). La standardisation est un

moyen de réduire les effets de ces facteurs et, par conséquent, de rendre les mesures effectuées comparables.

3.5 Conclusion

En génie logiciel, il est loin d'y avoir un consensus sur les propriétés requises pour les mesures du logiciel et l'inventaire dressé plus haut en fait le constat. La recherche d'un consensus sur les propriétés des mesures des logiciels doit préoccuper aussi bien les chercheurs que les industriels. Ce chapitre a présenté un inventaire dressé à partir des propriétés désirables des mesures des logiciels proposées par plusieurs chercheurs du domaine et exposé par Zuse dans son livre *A Framework for Software Measurement*, et des propriétés des mesures des logiciels recommandées par la norme ISO 14143-3. Cet inventaire peut aider dans la sélection des propriétés des mesures de logiciel, comme les mesures de la taille fonctionnelle des logiciels. De cet inventaire, une seule propriété est retenue comme focus de recherche soit la **standardisation** des mesures.

Afin d'approfondir les connaissances en matière de standardisation des mesures, le chapitre suivant présente les concepts de standardisation et de métrologie.

CHAPITRE 4

LA MÉTROLOGIE ET LA STANDARDISATION

« Les poids et mesures font partie des nécessités de la vie de chaque être humain »
(John Quincy Adams, 1821)

La métrologie et la standardisation des mesures est bien présente dans la conception et l'application des mesures dans les domaines classiques tels que le génie électrique, le génie civil, le génie mécanique, mais pratiquement absente en génie logiciel. Ce chapitre est composé de trois sections. La première section présente la définition de la métrologie. La deuxième section donne un aperçu sur le concept de standardisation. Le chapitre situera la métrologie par rapport à ISO.

4.1 Introduction

Selon Carty (2000), la métrologie et les étalons facilitent l'échange des biens, favorisent l'automatisation de la production, haussent la qualité des produits, permettent au consommateur d'être plus confiant et donnent un niveau de vie meilleur. « La métrologie – et le fait que les mesures physiques, chimiques, électriques et autres reposent sur des étalons communs – a un effet énorme sur nos vies quotidiennes » (A. Carty, 2000).

4.2 Définition de la métrologie

Le terme « métrologie » inclut tous les aspects de la mesure que ce soit théorique ou pratique. La figure 5 présentée par Abran et al. (2003) est une approche Top-down pour montrer la place de la métrologie dans le concept « Mesure » et représente une modélisation des principaux concepts de métrologie documentés dans le Vocabulaire International ISO sur la métrologie (ISO, 1993).

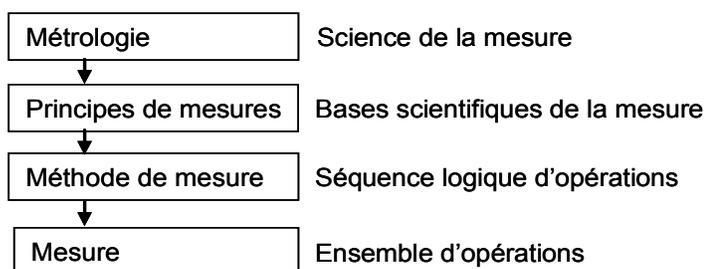


Figure 5 Place de la métrologie dans le concept Mesure (Abran et al., 2003)

Le passage à une connaissance se fait fréquemment par un nombre; la mesure qui produit ce nombre ne peut se créer sans unités, étalons et dispositifs de mesure. La métrologie n'est pas uniquement une discipline particulière des sciences physiques mais c'est aussi la base de nos tâches journalières (Bureau national de la métrologie, 2003). Selon le bureau national de la métrologie - BNM (2003), la métrologie assure un cadre théorique qui permet de faire confiance aux bilans des mesures pour la production et les échanges de produits, le contrôle des avions, le traitement des eaux, le fonctionnement des stations nucléaires etc.

En effet, de nos jours, le poids stratégique de la métrologie pour l'industrie et l'ensemble des disciplines humaines n'a jamais été aussi fondamental.

Trois éléments sont à l'origine de cette importance de la métrologie :

1. L'échange international des biens, la globalisation d'investissement et de production;
2. L'application de normes internationales pour développer, pour produire et pour livrer des produits;
3. L'évolution ultra rapide de la haute technologie dans presque tous les secteurs économiques.

Pour ces trois éléments étroitement reliés, le besoin d'une métrologie mature augmente dans tous les domaines de génie, et la coopération internationale est de plus en plus demandée dans ces domaines.

Cependant, l'intérêt à la métrologie dans la communauté ISO du génie logiciel est récent : il n'y a actuellement que quelques travaux de recherche pour appliquer les concepts de métrologie classique aux mesures des logiciels (A. Abran et al., 2003; A. Khelifi et al., 2004). De même, parmi toutes les centaines, si ce n'est des milliers de mesures, proposées pour le logiciel, les seules méthodes de mesure qui ont été abordées à date au niveau de la normalisation internationale avec ISO sont les méthodes de mesure de la taille fonctionnelle (MTF) des logiciels.

4.3 La standardisation

En cette ère où la compétitivité est reine, la conformité à des normes internationales confère un avantage considérable sur la concurrence étrangère.

4.3.1 La définition du standard

Norme ou standard? En anglais, il existe un seul mot qui est standard. En français deux mots sont utilisés, soit « norme » et « standard ».

« Les deux termes désignent une publication finale résultante d'un accord consensuel. La différenciation semble se situer essentiellement au niveau des acteurs en jeu et des procédures de consensus attachées » (G. Chartron, 2000).

- La norme fait référence surtout à ISO et à ses instances nationales telles que le SCC (Standards Council of Canada) au Canada avec des processus d'élaboration de consensus assez longs.
- Le standard est plus assimilé à un processus réactif de consensus du monde économique ou du monde technique.

Un standard est un ensemble de recommandations développées et préconisées par un groupe représentatif d'utilisateurs. Une norme selon ISO est « un document établi par un consensus et approuvé par un organisme reconnu, qui fournit, pour des usages communs et repérés, des règles, des lignes directrices ou des caractéristiques, pour des activités ou leurs résultats, garantissant un niveau d'ordre optimal dans un contexte donné. »

4.3.2 Le contenu d'une norme

Les normes varient en caractère, en sujet et en moyen. Les normes peuvent couvrir plusieurs disciplines. Elles traitent des aspects techniques, économiques et sociaux de l'activité humaine et touchent toutes les disciplines de base telles que la langue, les mathématiques, la physique, etc. En général, voici quelques caractéristiques des normes adaptées des caractéristiques proposées par le World Standards Services Network (2003):

- Cohérentes : les normes sont développées par des comités techniques, coordonnés par un corps spécialisé, et assurent que les barrières entre différents secteurs d'activité et différents commerces sont surmontées.
- Résultent de la participation : les normes reflètent les résultats du travail commun impliquant toutes les parties compétentes concernées et sont validées par consensus pour représenter tous les intérêts mutuels des producteurs, des utilisateurs, des laboratoires, des services publics, des consommateurs, etc.
- Des processus vivants : les normes sont basées sur une expérience réelle et mènent aux résultats matériels dans la pratique. Elles établissent un compromis entre la situation actuelle et les contraintes économiques du temps.
- À jour : les normes sont révisées de façon périodique pour assurer leur circulation et leur évolution avec le progrès technologique et social.
- Ont un statut de référence : dans les contrats commerciaux et devant le tribunal en cas de conflit.
- Reconnaissance nationale ou internationale : les normes sont des documents qui sont reconnus comme valides aux niveaux régional, national ou international.
- Disponibles pour tous : les normes peuvent être consultées et achetées sans restriction.
- En règle générale, les normes ne sont pas obligatoires, mais sont pour application volontaire. Dans certains cas, l'application est obligatoire (comme dans les

domaines liés à la sécurité, aux installations électriques et en relation avec les contrats publics, etc.).

4.3.3 Rôle des normes

Une norme représente un niveau de savoir-faire et de technologie qui rend la présence de l'industrie nécessaire pour sa préparation. C'est un document de référence utilisé en particulier dans le contexte des contrats publics ou dans celui du commerce international.

Une norme est employée par des industriels comme référence indiscutable, simplifiant et clarifiant des relations contractuelles entre partenaires économiques. C'est un document qui est employé de plus en plus en jurisprudence. Selon le World Standards Services Network (2003), pour les partenaires économiques, la norme est :

- Un facteur de rationalisation de la production : la norme permet de maîtriser les caractéristiques techniques, de satisfaire les clients, de valider les méthodes de fabrication, d'augmenter la productivité et de donner aux gens un sentiment de sécurité.
Un facteur de clarification des transactions : confronté aux offres surabondantes des produits et des services qui peuvent avoir des valeurs pratiques extrêmement différentes, l'existence des systèmes de référence permet d'évaluer mieux les offres et de réduire les incertitudes, d'aider dans la définition des besoins, et d'optimiser les relations de fournisseur pour éviter les tests additionnels.
- Un facteur d'innovation et de développement des produits : participer au travail d'étalonnage permet de prévoir et de développer des produits simultanément. Les normes jouent un rôle favorable dans l'innovation grâce aux transferts des connaissances.
- Un facteur de transfert de nouvelles technologies : l'étalonnage facilite et accélère le transfert des technologies dans les domaines qui sont essentiels pour les compagnies et les individus (nouveaux matériaux, systèmes d'information, biotechnologie, électronique, etc.).
- Un facteur de choix stratégique pour les compagnies : participer à l'étalonnage signifie présenter des solutions adaptées à la compétence de sa compagnie et bien s'équiper dans les environnements économiques concurrentiels.

L'étalonnage est aujourd'hui identifié comme étant une discipline essentielle pour tous les joueurs économiques. Il y a 20 ans, ce champ était réservé à quelques spécialistes. Aujourd'hui, les compagnies ont intégré l'étalonnage comme élément technique et commercial important. Les organisations doivent jouer un rôle actif dans ce domaine, ou bien elles se préparent à accepter des étalons établis sans elles et sans la considération de leurs intérêts.

4.3.4 Types de normes

Le World Standards Services Network (2003) propose quatre types de normes :

- Normes fondamentales qui concernent la terminologie, la métrologie, les conventions, les signes et les symboles, etc.;
- Les méthodes de test et les normes d'analyse qui mesurent les caractéristiques d'un produit ou d'un service;
- Les normes qui définissent les caractéristiques d'un produit ou d'un service ainsi que le seuil de performance;
- Les normes d'organisation qui traitent la description des fonctions d'une compagnie et ses relations, ainsi que la structure de ses activités (gestion et assurance de qualité, entretien, analyse de la valeur, logistique, gestion de qualité, gestion de projet ou de systèmes, gestion de la production, etc.).

Actuellement, le génie logiciel est doté de plusieurs normes (IBM, 2005; IEEE Computer Society, 1993; IEEE Std 610.12-1990, 1990; IEEE Std 830-1998, 1998; ISO/IEC DIS 9126-1 to 4, 2001; ISO/IEC Document: Guide 25, 1990; James Rumbaugh et al., 1999; James W. Moore, 1997; Object Management Group, 2005). L'existence de ces normes prouve que la discipline du génie logiciel devient de plus en plus mature, mais en même temps soulève des questions significatives. Par exemple, le génie logiciel ne possède pratiquement pas de normes spécifiques pour la mesure des logiciels, à une seule exception près, soit pour la mesure de la taille fonctionnelle du logiciel.

4.4 La métrologie dans ISO

Dans les disciplines matures d'ingénierie, la mesure est basée sur un corpus de connaissance riche accumulé au long des siècles, voire des milliers d'années, connu sous le nom du « domaine de la métrologie ». Dans les pays développés, ce domaine est soutenu par des agences gouvernementales de métrologie comme l'Institut des étalons nationaux de mesure du Canada, le Bureau National de Métrologie et l'Institut National de Métrologie de France, Asia Pacific Metrology Programme, Beijing Institute of Radio Metrology and Measurement, de Chine, Council for Optical Radiation Measurements des États Unis, The State Committee of the Russian Federation for Standardization, Metrology and Certification de la Russie, etc

Le document ISO Vocabulaire international des termes fondamentaux et généraux de métrologie (1993) est un accord international sur la terminologie, et découle de la collaboration entre des experts internationaux dans le domaine de la métrologie. « Ce vocabulaire traite de sujets en rapport avec le mesurage et comporte des informations sur la détermination des constantes physiques et des autres propriétés fondamentales des matériaux et substances » (ISO, 1993).

Bien que ce document clef soit largement connu dans le domaine de la métrologie, il est presque inconnu dans « la communauté des métriques de logiciel ». Ce standard contient 120 termes qui ont des relations entre eux. Pour représenter les rapports entre les termes vocabulaires de la norme ISO, les auteurs (A. Abran et al., 2002) ont choisi la représentation classique d'un processus de production pour illustrer ces multiples concepts. Un processus de production se compose d'une entrée, d'une sortie, de variables de contrôle et du processus lui-même de production. Sur la figure 6 qui représente l'ensemble des concepts de la métrologie, la sortie est représentée par les « résultats de mesure » et le processus lui-même par « la mesure » dans le sens des opérations de mesure, alors que les variables de contrôle sont « les étalons » et les « quantités et unités ». À noter que les opérations de mesure et les résultats de mesure sont

influencés par les « caractéristiques » des instruments de mesure. Dans cette figure 6, un terme présent dans le VIM d'ISO est représenté en fonte roman, alors que les termes ajoutés par les auteurs sont en italiques, par exemple le terme « entrée » :

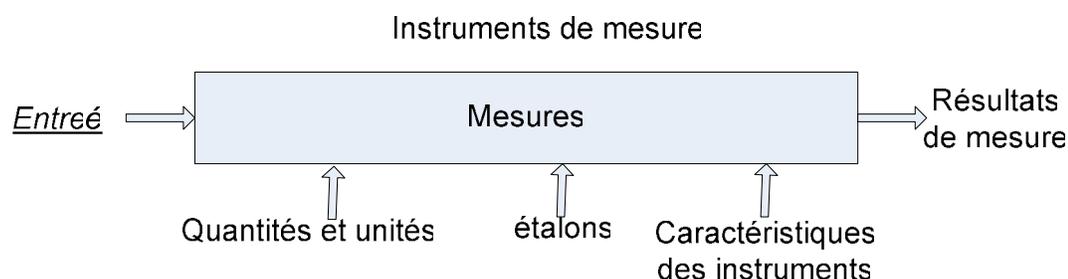


Figure 6 Relations entre les termes de métrologie (A. Abran et al., 2002)

Cet ensemble de concepts représente le processus de mesure. C'est évident que l'opération de la mesure et son résultat dépendent des éléments en entrée et également de l'instrument de mesure.

Tandis que ces concepts sont bien connus dans plusieurs domaines, ils sont souvent méconnus en génie logiciel : par exemple, une grande partie de ces concepts n'est pas introduite dans les pratiques des concepteurs de mesure et des mesureurs eux-mêmes.

4.5 Conclusion

Ce chapitre a présenté une vue d'ensemble de la métrologie et présenté le concept de standardisation qui est la propriété des mesures des logiciels à considérer dans ce travail. Le chapitre a porté sur une description sommaire du contenu d'une norme, les rôles des normes, et les types de normes. D'après la revue de la littérature, il a été constaté que la mesure de la taille fonctionnelle (MTF) des logiciels est presque le seul attribut des logiciels traité selon la propriété standardisation. Il sera montré ultérieurement que l'organisation ISO a accepté plusieurs normes pour cet attribut.

Le prochain chapitre discute des Mesures de la Taille Fonctionnelle (MTF) des logiciels. Il commence par un aperçu historique sur la mesure des logiciels, puis une comparaison est engagée entre cette méthode de mesure avec les mesures les plus connues du domaine. Ensuite, les normes ISO qui utilisent la MTF sont présentées. À la fin, une explication du choix de la méthode COSMIC-FFP (ISO 19761) comme méthode de mesure des logiciels pour les étalons est présentée.

CHAPITRE 5

LES MESURES DE LA TAILLE FONCTIONNELLE (MTF)

«Une science a l'âge de ses instruments de mesure » Louis Pasteur

Au début de ce chapitre, un aperçu historique sur les classes de mesures des logiciels est présenté. Puis une discussion est engagée sur le concept des points de fonction et de la taille fonctionnelle. Ensuite, une comparaison est discutée entre la méthode de Lignes de Code (LOC) qui permet le comptage du nombre de lignes de code d'un programme et la méthode des points de fonction. Ce chapitre décrit aussi la méta-norme de mesurage de la taille fonctionnelle - ISO 14143 et les MTF qui ont été reconnues comme des normes ISO. Enfin, un rapprochement est aussi réalisé entre deux de ces normes ISO, soit COSMIC-FFP et IFPUG.

5.1 Introduction

McCabe et Butler (1998) proposent trois grandes classes essentielles de mesures des logiciels; la complexité cyclomatique, le nombre de lignes de code et le nombre de points de fonction.

D'autre part, selon Fenton (1997), le logiciel est une entité physique qui peut être mesurée par sa taille, puisque les objets physiques sont facilement mesurables. L'opération de mesurer la taille d'un logiciel devrait être facile. Cependant, la mesure de la taille des logiciels présente beaucoup de difficultés, car il n'y a pas de compréhension profonde des attributs des logiciels pas plus que de disponibilité d'outils sophistiqués de mesure (N. Fenton et S.L. Pfleeger, 1997). Fenton *et al.* suggèrent que la taille des logiciels peut être décrite par trois attributs à savoir : la longueur, la fonctionnalité et la complexité.

La longueur est la taille physique du produit logiciel alors que la fonctionnalité mesure les fonctions fournies par le logiciel à l'utilisateur (N. Fenton et S.L. Pfleeger, 1997). Quant à la complexité, elle peut être interprétée de plusieurs façons. La complexité du problème mesure la complexité du problème à résoudre par le logiciel. La complexité algorithmique reflète la complexité de l'algorithme implanté pour résoudre le problème. La complexité structurelle mesure la structure du logiciel exécutant l'algorithme. La complexité cognitive mesure l'effort exigé pour comprendre le logiciel. D'après Fenton (1997), ces interprétations font de la complexité un attribut difficile à mesurer. De plus, il faudrait faire une distinction entre la complexité du problème et la complexité de la solution.

La méthode des lignes de code (LOC) peut être définie sommairement comme le nombre de lignes source écrites dans un programme; son unité est 1 ligne de code. Jones (1978) souligne que cette mesure de logiciel présente quelques problèmes fondamentaux. Le problème principal consiste à savoir comment déterminer précisément une ligne de code. Les lignes de code des applications sont formées des déclaratifs, des appels, des instructions et des commentaires. De plus, les lignes de code contiennent parfois des instructions complexes qui peuvent prendre plusieurs lignes physiques de code.

La méthode des lignes de code peut être définie de façon différente d'une entreprise à une autre : tantôt, une entreprise ne regarde que les instructions exécutables, dans d'autres situations une entreprise ne compte que les instructions et les déclarations, et quelquefois on considère toutes les lignes non blanches dans le code. À cause de ces disparités dans les façons de mesurer les lignes de code, il est délicat de comparer la taille des logiciels en utilisant uniquement cette méthode de mesure de la taille d'un logiciel. De plus, les différences entre les langages de programmation rendent cette méthode de mesures encore moins pertinente (Fenton *et al.*, 1999).

D'après Zuse, la méthode de lignes de code a été l'objet de critiques très sévères : la critique la plus évidente, c'est que cette technique ne mesure pas les fonctions du

système et ses attributs : *The obvious criticism of the simple Measure LOC is that it does not really measure the systems functions and features* (Horst Zuse, 1998).

5.2 Les points de fonction

Allan Albrecht d'IBM a identifié, au cours des années 1970, la nécessité d'avoir une méthode d'estimation de l'effort déployé pour réaliser un logiciel qui ne dépendrait pas des langages de programmation ni des techniques utilisées pour développer ce logiciel. Comme résultat, Albrecht (1979) a inventé la méthode des points de fonction (*Function Points Analysis – FPA*). C'était une idée originale de proposer, pour la première fois, une mesure des logiciels fondée sur les fonctions des utilisateurs. En résumé, les points de fonctions se basent sur l'analyse et la mesure des fonctionnalités livrées à l'utilisateur. Par design, cette méthode de mesure n'a pas certains des problèmes de la mesure LOC. La définition de la méthode des points de fonction a été discutée par beaucoup d'auteurs tels que (A. Abran, 1994; B. Kitchenham *et al.*, 1993; C. Jones, 1996; C. Symons, 1988; C.F. Kemerer, 1993; D. St-Pierre *et al.*, 1997; D.J. Reifer, 1991; J.B. Dreger, 1989; K. Paton *et al.*, 1995; M. Shepperd, 1994; R. Banker *et al.*, 1994; S. Galea, 1995; S.A. Whitmire, 1995; S.L. Pfleeger *et al.*, 1990).

Les objectifs d'utilisation des points de fonction sont présentés au tableau 15.

Tableau XV

Objectifs d'utilisation des points de fonction (IFPUG, 1994)

Version française (Abran)	Version anglaise IFPUG
1 Mesure ce que l'utilisateur a demandé et reçu	1-Measure what the user requested and received
2 Mesure indépendamment de la technologie implantée	2-Measure independently of technology used for implementation
3 Obtenir une mesure de taille pour permettre l'analyse de la qualité et de la productivité	3-Provide a sizing metric to support quality and productivity analysis

Tableau XVI (Suite)

Objectifs d'utilisation des points de fonction (IFPUG, 1994)

Version française (Abran)	Version anglaise IFPUG
4 Obtenir un véhicule pour l'estimation du logiciel	4-Provide a vehicle for software estimation
5 Obtenir un facteur de normalisation pour comparer les logiciels	5-Provide a normalization factor for software comparison

5.3 Lignes de code ou points de fonction

Dans son article *Function Points or Lines of Code? – An Insight*, Kurmanadham (2004) apporte la comparaison suivante entre les lignes de code et les points de fonction :

5.3.1 Les avantages de la méthode des lignes de code

1. *Scope for Automation of Counting* : puisque la ligne de code est une entité physique, l'effort du comptage manuel peut être éliminé en automatisant le processus de comptage.
2. *An Intuitive Metric* : La ligne de code sert de métrique intuitif pour mesurer la taille du logiciel parce qu'elle est visible et que son effet peut être visualisé.

5.3.2 Les inconvénients de la méthode des lignes de code

1. *Lack of Accountability* : il est incorrect de calculer la productivité d'un projet de développement avec les résultats d'une des phases (phase de codage) qui représente habituellement de 30 à 35 % de l'effort (Kurmanadham, 2004).
2. *Lack of Cohesion with Functionality* : Les développeurs expérimentés peuvent développer la même fonctionnalité avec moins de lignes de code que des programmeurs novices : ainsi un programme avec moins de LOC peut donner autant de fonctionnalité qu'un autre programme avec plus de LOC.
3. *Adverse Impact on Estimation* : Comme corollaire au point(a), les estimations faites basées sur des lignes de code peuvent être biaisées.
4. *Developer's Experience* : Un développeur expérimenté peut mettre en application certaines fonctionnalités en moins de lignes de code qu'un autre développeur qui a relativement moins d'expérience, bien qu'ils utilisent le même langage de programmation.

5. *Difference in Languages* : Considérer deux applications qui fournissent la même fonctionnalité (écrans, rapports, bases de données), une est écrite en C++ et l'autre est écrite en COBOL. Le nombre de points de fonction serait exactement identique, mais les lignes du code requises pour développer la même application ne seraient probablement pas identiques.
6. *Advent of GUI Tools* : Avec l'arrivée des langages et des outils basés sur l'Interface Utilisateur Graphique tel que le Visual Basic, beaucoup de travail de développement est effectué par « cliquer-et-glisser » où le programmeur n'écrit pratiquement aucun morceau de code. Cette différence rend la mesure avec les lignes de code moins pertinente pour les mesures de productivité.
7. *Far from OO Development* : La ligne de code ne fait aucune signification dans le cas du développement Orienté Objet où tout est traité en termes d'objets et de classes. Puisque l'objet est une représentation des données et des fonctionnalités, les points de fonction sont plus appropriés pour les logiciels développés en environnement Orienté Objet.
8. *Problems with Multiple Languages*: Maintenant dans le développement des logiciels, plusieurs langage de programmation sont utilisées. Dans ce cas la détermination et la traçabilité des taux de productivité et de défauts sont difficiles. Puisque les défauts ne peuvent pas être attribués à un seul langage ceci est due à l'intégration du système. Les points de fonction s'apprêtent d'être la meilleure mesure de taille dans ce contexte. (Kurmanadham, 2004)
9. *Lack of Counting Standards* : Il n'y a aucune définition standard d'une ligne de code.

5.3.3 Les avantages de la méthode des points de fonction

Les points de fonction fournissent une méthode pour mesurer la taille des logiciels, et cette mesure peut être utilisée pour gérer la taille pendant le développement. « La méthode des points de fonction est largement utilisée dans le domaine des systèmes d'informatique de gestion » (Banker *et al.*, 2002; Desharnais *et al.*, 1998). Elle permet :

1. *Helps Comparison* : La seule variable est la quantité d'effort requise pour fournir un ensemble de points de fonction; cette méthode est, par design, indépendante des technologies et des méthodologies de développement; la méthode peut donc être employée pour déterminer si un outil, un environnement, un langage de programmation est plus productif comparé à d'autres dans une même organisation ou par rapport à d'autres organisations.
2. *Helps Monitor Scope Creep* : Le comptage des points de fonction à la fin de la phase de spécification ou de design peut être comparé aux points de fonction réellement fournis et implantés à la fin d'un projet. Si la taille du projet a grandi,

il y a eu changement de portée et ce changement de taille peut être mesuré avec les points de fonction.

3. *Ease of Contract Negotiations* : Pour le client, les points de fonction peuvent être employés pour aider à indiquer et à quantifier pour un fournisseur, les principaux livrables et ce, sur une base commune et acceptée par l'industrie.
4. *Handling Volatility* : L'avantage est que les points de fonction fournit des mesures a partir des exigences du logiciel. Donc, les points de fonction permet de montrer l'état actuel des exigences. Quand une nouvelle caractéristique est ajoutée, le total des points de fonction augmente en conséquence. Si l'organisation décide d'enlever des caractéristiques, le total des points de fonction change en reflétant l'état réel des caractéristiques du logiciel. (Kurmanadham, 2004)
5. *Use of Historic Data* : Une fois que la taille de projet a été déterminée en points de fonction, des évaluations pour la durée, l'effort, et d'autres coûts peuvent être calculées en employant des données historiques.
6. *Availability of Empirical Formulae* : Les points de fonction peuvent être employés plus efficacement pour développer beaucoup de formules prédictives telles que le taux de défaut et l'effort de maintenance.
7. *Enables Better Communication* : Les points de fonction peuvent améliorer la communication avec la haute direction puisqu'ils parlent en termes de fonctionnalité et non pas en termes de détails d'exécution, d'aspects techniques ou de code physique.
8. *Offers Better Benchmarking* : Puisque les points de fonction sont indépendants du langage de programmation, de la méthodologie de développement, des pratiques de programmation, et du domaine technologique, les projets utilisant les points de fonction sont de meilleurs candidats pour faire du *benchmarking* à travers les organismes.

5.3.4 Les inconvénients de la méthode des points de fonction

Kurmanadham (2004) souligne que les points de fonction offrent de nombreux avantages en donnant la taille du logiciel du point de vue fonctionnel. Cependant, la méthode a quelques inconvénients :

1. *Requires Manual Work* : En raison de sa nature, les points de fonction doivent être comptés manuellement. Le processus de comptage est difficilement automatisable.
2. *Necessitates Significant Level of Detail* : L'information a un niveau de granularité très fin sur les entrées, les sorties, les écrans, les tables de base de données, et même les enregistrements et les champs sont nécessaires pour faire la mesure.

3. *Requires Experience* : L'analyse de points de fonction exige de l'expérience pour avoir des mesures précises.

De plus, selon des auteurs tels que (Desharnais *et al.*, 1998; Galea, 1995; C. Jones, 1991; Maya *et al.*, 1996; Whitmire, 1992) la méthode est moins adéquate pour les logiciels qui ont plusieurs fonctions de contrôle ou plusieurs opérations de calculs, tels que les logiciels temps réel. Pour ces types de fonctionnalité, cette méthode ne tient pas compte des sous-processus fonctionnels à l'intérieur d'un processus de contrôle, ce qui génère des mesures fonctionnelles trop petites ne reflétant pas de façon précise la taille fonctionnelle des logiciels en temps réel. Pour remédier à certains de ces défauts, le Consortium COSMIC (Common Software International Consortium) a développé la méthode de mesure COSMIC-FFP.

5.4 La méta-norme ISO 14143

À la fin des années 1990, une norme générique ISO en mesure de taille fonctionnelle a été créée pour définir ce qu'est une mesure de taille fonctionnelle - MTF (ISO/IEC 14143-1, 1998) et comment en vérifier certaines caractéristiques de qualité. Le but de cette norme est de fournir un ensemble de documents (internationalement acceptés comme normes ou rapports techniques) qui décrivent le concept et la pratique de la MTF. Sur la base de critères définis dans cette norme générique ISO, plusieurs méthodes spécifiques de mesures fonctionnelles ont par la suite été acceptées comme norme spécifiques ISO reconnues comme rencontrant les critères obligatoires spécifiés dans la norme générique de ISO 14143-1. La norme COSMIC-FFP (ISO/IEC 19761, 2002) est un exemple d'une norme spécifique en MTF.

ISO 14143 est une norme générique (qui pourrait également être appelée une métanorme) dont le titre est « Technologies de l'information - Mesurage du logiciel - Mesurage de la taille fonctionnelle » et qui est composée de six parties, les deux premières ayant un statut de normes ISO et les quatre autres le statut de rapport technique ISO. La publication de ces parties s'est effectuée de 1998 à 2005 :

Partie 1 — Définition des concepts (ISO/IEC 14143-1, 1997)

ISO/IEC 14143- 1 définit les concepts fondamentaux de la MTF des logiciels et décrit les principes généraux pour le design d'une méthode MTF. Cette partie constitue le repère contre lequel toutes les méthodes MTF doivent être examinées pour s'assurer qu'elles se conforment à ses critères.

Partie 2 – Évaluation de la conformité des méthodes MTF des logiciels (ISO/IEC 14143-2, 2001)

ISO/IEC 14143-2 est une norme de conformité qui propose une méthodologie pour examiner si une méthode a les caractéristiques exigées selon la norme de référence 14143-1. Cette partie exige un processus rigoureux d'analyse, un rapport complet et une partie indépendante pour vérifier la conformité de la méthode plutôt que de se contenter des déclarations des fournisseurs. Le résultat de ce processus peut aider les éventuels utilisateurs d'une méthode candidate de MTF pour juger si elle est appropriée à leurs besoins.

Partie 3 – Vérification des méthodes MTF (ISO/IEC 14143-3, 2002)

La partie 3 est un rapport technique ISO développé pour fournir un processus afin d'aider l'utilisateur à choisir une méthode appropriée à ses besoins, en fournissant un processus pour vérifier à quel point les déclarations faites pour certaines propriétés d'une méthode MTF sont vraies. Cette partie définit :

1. un cadre pour vérifier certaines propriétés d'une méthode MTF
2. plusieurs propriétés contre lesquelles une méthode MTF peut être vérifiée
3. les types d'essais qui peuvent être réalisés
4. le processus pour la vérification d'une méthode MTF
5. des exemples pour le rapport de vérification.

La partie 3 d'ISO/IEC 14143 établit un cadre pour vérifier les propriétés d'une méthode MTF par rapport aux critères suivants :

1. répétabilité et reproductibilité;
2. exactitude;
3. convertibilité;
4. seuil de tolérance;
5. applicabilité aux domaines fonctionnels.

La partie 3 vise à ce que le résultat de la vérification soit objectif, impartial, conforme et répétable. Le rapport de vérification, produit en conformité avec cette partie d'ISO/IEC 14143, permettra à un éventuel utilisateur de choisir la méthode MTF qui répond bien à ses besoins.

Partie 4 – Modèle de référence (ISO/IEC 14143-4, 2000)

D'après l'*International Software Benchmarking Standards Group* (ISBSG, 2005), le but de cette partie est de fournir des points de référence aux utilisateurs du processus de validation peuvent évaluer l'efficacité d'une méthode FSM pour différents types de logiciels et dans plusieurs environnements.

Ce rapport technique peut permettre :

- Aux développeurs d'une méthode de FSM de tester les domaines fonctionnels dans lesquels leur méthode peut être utilisée efficacement et de raffiner leur méthode FSM;
- Aux personnes vérifiant une méthode de FSM, d'avoir des points de référence pour appliquer et pour comparer des méthodes FSM;
- La réduction de l'utilisation inadéquate de quelques méthodes courantes de mesure des logiciels;
- Une meilleure comparaison des données utilisées pour le *benchmarking* de qualité et de productivité;
- Aux mesureurs de vérifier qu'une méthode particulière de FSM est une méthode efficace;
- Aux mesureurs d'avoir un ensemble d'exigences d'utilisateurs fonctionnelles dans différents domaines fonctionnels.

Partie 5 – Détermination des domaines fonctionnels pour les utiliser avec les MTF (ISO/IEC 14143-5, 2004)

Selon l'*International Software Benchmarking Standards Group (ISBSG, 2005)*, le but de cette partie 5 est d'établir une norme pour classer les fonctionnalités utilisateurs requises pour utilisation dans l'application des FSM.

Ce rapport technique permet aux :

- Utilisateurs des FSM d'évaluer les caractéristiques de leurs fonctionnalités utilisateurs requises et de les classer par catégorie de domaine fonctionnel;
- Développeurs d'une méthode FSM d'énoncer clairement les domaines fonctionnels pour lesquels leur méthode peut être employée efficacement;
- Testeurs de conformité des FSM d'avoir des directives claires pour déterminer les domaines fonctionnels pour lesquels leur méthode est efficace;
- Mesureurs d'évaluer le domaine fonctionnel d'un ensemble particulier de FUR afin de déterminer quelle méthode de FSM est la plus appropriée pour leurs besoins.

Partie 6 – Guide d'utilisation des séries ISO 14143 et les normes internationales qui y sont reliées (ISO/IEC 14143-6, 2003)

La partie 6 présente :

1. une description des relations entre :
 - i. la série de documents d'ISO/IEC 14143 qui fournit les définitions d'une MTF et les éléments qui sont exigés par les définitions;
 - ii. les normes ISO 19761, 20926, 20968 et 24570 qui sont des normes spécifiques pour la MTF ;
2. un processus de sélection d'une méthode MTF qui répond aux exigences d'un utilisateur.

Cette revue de la littérature va maintenant présenter un aperçu des normes spécifiques en MTF mais ne va pas s'attarder sur les règles et sur les procédures des méthodes IFPUG, NESMA et MKII, car l'objectif ici est différent de leur description détaillée. Tout lecteur

intéressé à connaître plus en détail les concepts et les étapes de ces méthodes peut trouver dans les références les informations nécessaires pour ces trois méthodes de mesure).

5.5 IFPUG

L'*International Function Point Users Group (IFPUG)* fut fondé en 1986. Ce groupe d'utilisateurs de la méthode FPA a apporté des raffinements à la technique originale d'Albrecht. Ce groupe a comme objectif d'assurer une plus grande diffusion de cette méthode. La méthode de comptage en points de fonction est décrite en détail dans le document *Counting Practices Manual - CPM (IFPUG, 1994)*. Le manuel CPM contient aussi des exemples pour illustrer ses procédures, ses règles et ses formules de calcul.

Les règles IFPUG :

La méthode FPA est définie selon le point de vue de l'utilisateur. La vue de l'utilisateur est basée sur cinq types de fonction. Ces derniers sont partagés en deux catégories :

1. Les fonctions données; fichier logique interne (FLI) et fichier interface externe (FIE).
2. Les fonctions transactionnelles; entrée externe (EE), sortie externe (SE) et interrogation externe (IE).

Les relations entre ces cinq types de fonctions sont illustrées à la figure 7.

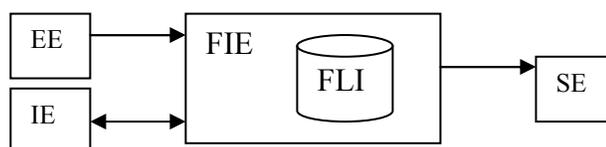


Figure 7 Types de fonction FPA

Les concepts de la méthode à un niveau plus fin de granularité sont :

Data Element Type-DET, *Record Element Type-RET* et *File Type Referenced-FTR*, se traduisant par ; Type d'élément de données-TED, Type d'élément d'enregistrement-TEE et Type de fichier référencé-TFR.

Les processus de comptage comprennent les activités suivantes :

1. Déterminer le type de comptage;
2. Identifier la frontière de l'application;
3. Compter les types de fonctions transactionnelles;
4. Déterminer la valeur du facteur d'ajustement;
5. Calculer le nombre total des PF ajustés.

Il est à noter que bien que les activités 4 et 5 fassent partie de la version initiale de la méthode FPA, ces activités ne font pas partie de la norme ISO correspondante parce que ne rencontrant pas les critères requis par ISO 14143-1.

La frontière entre des applications qui sont liées entre elles ne doit pas être basée sur des considérations technologiques mais sur la séparation des fonctions de gestion telles qu'elles sont vues par l'utilisateur.

5.6 NESMA - ISO/IEC 24570

NESMA (Netherlands Software Metrics Users Association) a été fondé en 1989 sous le nom de NEFPUG (groupe hollandais d'utilisateurs de points de fonction). Son objectif est de :

1. Rassembler, maintenir, échanger et développer la connaissance au sujet de la méthode FPA ainsi que son application;
2. Favoriser la standardisation de la méthode FPA;
3. Favoriser l'utilisation de la méthode FPA.

Les similitudes avec la méthode IFPUG sont résumées ainsi :

« Both the NESMA (CPM 2.0) and the IFPUG (CPM 4.1) now use the same philosophy, the same concepts and terms, and the same rules and guidelines within FPA » (NESMA, 2003).

Les utilisateurs des deux méthodes devraient obtenir des résultats comparables car les deux groupes emploient les mêmes philosophies, concepts et limites dans FPA.

Les deux groupes considèrent :

1. le comptage des fonctions que les utilisateurs peuvent identifier;
2. la précision sur le fait que les fonctions soient des processus élémentaires;
3. l'utilisation des mêmes types de fonction : entrée externe, sortie externe; interrogation externe, fichier logique interne, fichier interface externe;
4. le comptage des points de fonction non ajusté;
5. l'utilisation du facteur d'ajustement de valeur;
6. le comptage d'application en points de fonction;
7. le comptage de projet en points de fonction.

Les différences avec IFPUG ont été résumées comme suit :

The NESMA and the IFPUG have resolved almost all their differences in regard to how the complexity of functions should be established (NESMA, 2003). Les différences entre la méthode NESMA (CPM 2.0) et IFPUG (CPM 4.1) sont minimales, étant donné le fait que les deux méthodes partagent les mêmes règles et les mêmes directives dans FPA. Ces différences ont un impact négligeable sur les résultats de comptage des points de fonction.

À l'inverse d'IFPUG, NESMA considère l'option de pouvoir travailler avec les données de sortie qui changent de taille. Cette différence n'a pas une grande influence sur le nombre de points de fonction d'une application ou d'un projet, parce qu'elle affecte seulement le type de fonction mais pas la quantité de fonctions identifiables.

5.7 MKII - ISO/IEC 20968

La méthode Mk II (dans ce document le mot MK II veut dire MarK II FPA *Functional Point Analysis*) est définie par Charles Symons en 1991. Le comité des pratiques de mesures du Royaume-Uni est responsable de la méthode et de son développement.

Les activités de MKII pour le comptage de la taille fonctionnelle des logiciels (C. Symons, 1998) sont :

1. Déterminer le but et le type de mesure. Cette activité consiste à identifier le client qui a demandé ce comptage. S'agit-il de mesurer le rendement d'une équipe de développeurs? Est-ce que le but est de compter toutes les fonctionnalités qui ont été exigées, ou les fonctionnalités fournies à l'utilisateur?
2. Définir la frontière de comptage;
3. Identifier les transactions logiques;
4. Identifier et catégoriser les types d'entités de données;
5. Compter les types élément de données « entrée » (Te), les types référencés d'entités (Tr) et les types élément de données « sortie » (Ts) pour chaque transaction logique;
6. Calculer la taille fonctionnelle. La taille fonctionnelle est la somme de toutes les transactions logiques mesurées en Te, Tr et Ts en multipliant chaque type par son poids correspondant. La taille fonctionnelle de l'application est donnée par l'équation suivante : $NFP = Te * Pe + Tr * Pr + Ts * Ps$.
7. Déterminer l'effort et le temps passé pour développer le projet;
8. Calculer la productivité et d'autres performances;
9. Marquer les degrés d'influence;
10. Calculer l'ajustement de la complexité technique;
11. Calculer la taille des points de fonction ajustés.

Comparaison entre IFPUG 4.0 et MkII Version 3.0 :

« The MK II method works at a much finer level of granularity than the IFPUG method and this leads to lower sizes for small enhancements » (C. Symons, 1999).

Pour plus de détails sur la comparaison des deux méthodes, consulter le document en ligne de Grant Rule sur le web à: <http://www.gifpa.co.uk/library/Papers/Rule/MK2IFPUG.html>.

Les deux méthodes mesurent la taille de façon sensiblement différente d'un produit logiciel. En termes de taille produite, la différence est que la méthode MKII a une granularité plus fine et un processus de mesure continu alors que la méthode IFPUG limite la taille d'un composant à un seuil.

Les concepts utilisés pour mesurer la taille sont basés sur des entités et des transactions logiques permettant d'exprimer les exigences du logiciel et les spécifications fonctionnelles.

Les ajouts fournis par Charles Symons ont été conçus pour délivrer une mesure pour la taille d'une magnitude similaire pour les deux méthodes. En général, les méthodes donnent les mêmes tailles de logiciels allant jusqu'à 400 points de fonction. Pour des tailles plus grandes, la méthode Mk II FPA tend à produire une taille plus élevée que la méthode IFPUG (C. Symons, 1999).

Selon Symons, pour la méthode MK II, la mesure de la taille a été conçue pour être plus sensible aux petits changements des fonctionnalités que la méthode IFPUG.

5.8 COSMIC-FFP (ISO 19761)

La méthode COSMIC-FFP (*Common Software Measurement International Consortium – Full Function Points*) a été conçue pour mesurer la taille fonctionnelle du logiciel en temps réel et multicouches tel qu'utilisé dans les applications de télécoms, les systèmes d'exploitation, ainsi que pour les logiciels de gestion. La première version de la méthode COSMIC-FFP a été publiée en 1999 et en seulement quatre ans elle a réussi à franchir

toutes les étapes de développement d'un consensus international pour devenir une norme internationale (ISO 19761 : 2003).

L'association COSMIC a été fondée en 1998. C'est une association volontaire d'experts en matière de mesure de logiciels provenant d'Australie, du Canada, de la Finlande, d'Allemagne, d'Irlande, d'Italie, du Japon, de la Hollande et du Royaume Uni.

La méthode de mesure COSMIC-FFP se base sur les exigences fonctionnelles de l'utilisateur (Fonctionnalités Utilisateurs Requises – FUR – en anglais *Functional User Requirement*) pour mesurer un logiciel. Les FUR sont les pratiques et les procédures que le système doit réaliser pour satisfaire les besoins des utilisateurs.

COSMIC-FFP est composée principalement de quatre phases :

- A. Appliquer les FUR du logiciel à mesurer au modèle générique COSMIC-FFP ; Cette phase permet d'identifier les éléments de base indispensable à l'opération de mesure, elle est composée de quatre activités :
 - i. Identifier les couches du logiciel à mesurer : une couche est le résultat du partitionnement fonctionnel de l'environnement du logiciel. Par exemple le système d'exploitation, les interfaces utilisateurs et le SGBD sont des couches différentes;
 - ii. Délimiter la frontière du logiciel : la frontière est une ligne conceptuelle séparant le logiciel de son environnement. Par exemple, la souris, le clavier, l'imprimante, le disque dur, la RAM et la ROM participent à démarquer la frontière du logiciel à mesurer ;
 - iii. Définir l'utilisateur du système : un utilisateur peut être un individu, un autre logiciel ou un dispositif qui interagit avec le logiciel à mesurer;
 - iv. Identifier des éléments candidats de l'opération de mesure. Ces éléments ne seront pris en compte dans la mesure que s'ils sont validés par le modèle COSMIC-FFP. Ces éléments sont :
 - a) les événements déclencheurs : un événement déclencheur se passe à l'extérieur de la frontière du logiciel à mesurer

et qui initie un ou plusieurs processus fonctionnels (e.g. entrée des données);

- b) les processus fonctionnels : un processus fonctionnel est un ensemble unique et ordonné (e.g. soustraction des opérands) de mouvements de données (entrée, sortie, lecture et écriture).

B. Faire la mesure de la taille fonctionnelle proprement dite sur les FUR du logiciel à mesurer. Après avoir défini le modèle COSMIC-FFP, comme ci-dessus, on applique un ensemble de règles et de procédures fournies par la méthode COSMIC-FFP. La règle principale est le calcul des mouvements de données. En effet, la méthode permet de valider les processus fonctionnels définis précédemment et d'identifier tous les mouvements de données (un mouvement de données est un déplacement élémentaire d'un groupe de données pendant l'exécution d'un processus fonctionnel).

C. La présentation des résultats détaillés de la mesure :

- a) Le rang du processus dans la liste des processus fonctionnels
- b) L'identifiant du processus qui est composé de son numéro de couche et son rang dans la couche
- c) La description du processus fonctionnel
- d) L'événement déclencheur du processus fonctionnel
- e) La liste de tous les mouvements de données qui le composent :
 - i. leur description
 - ii. le groupe de données mis en cause
 - iii. le type du mouvement de données (entrée, sortie, lecture et écriture)
 - iv. la taille associée
 - v. la taille totale du processus fonctionnel.
- f) La taille totale du logiciel.

D. La représentation des résultats agrégés de la mesure

Cette phase finale résume les résultats de l'opération de mesure selon :

1. Le nombre de mouvements de données de chaque type par processus fonctionnel;
2. La proportion de la taille de chaque processus fonctionnel dans l'application;

3. La proportion de la taille de chaque type de mouvement de données dans l'application (Entrée, Sortie, Lecture et Écriture);
4. La représentation graphique de la proportion de chaque type de mouvement de données dans l'application.

5.9 Comparaison entre les méthodes IFPUG ET COSMIC-FFP

Dans son article *COSMIC-FFP and IFPUG 4.1 Similarities and Differences*, Pam Morris (2003), présente une comparaison des deux méthodes telles qu'illustrées dans les tableaux suivants.

La comparaison au niveau des concepts est présentée au tableau 16, la comparaison au niveau des processus au tableau 17, la comparaison de l'utilisation des résultats dans le modèle d'estimation au tableau 18 et la comparaison des ressources au tableau 19.

Tableau XVI

Comparaisons entre IFPUG et COSMIC-FFP

Concept	IFPUG	COSMIC-FFP
Méthodes de travail avec les logiciels multicouches	N'est pas explicite dans le guide de comptage	Explicite dans les règles de mesure des architectures multicouches
Vue utilisateur	Mesure selon la vue utilisateur « humaine » uniquement	Mesure à partir de différents points de vue
Exigences techniques et qualité	N'est pas mesuré explicitement	Les fonctionnalités requises par les autres couches utilisatrices sont considérées dans la couche mesurée

Tableau XVII

Comparaisons au niveau des processus

Processus	IFPUG	COSMIC-FFP
Comptage de DET	Oui	Non, la méthode compte les « groupes logiques » des DET
Valeur par défaut pour la complexité des données en industrie	Non	Non
Règles pour déterminer les pondérations	Différentes règles pour chaque type de processus	Les mêmes règles pour tous les processus
Regroupement répétable des groupes de données logiques	Exige une expérience en comptage pour assurer la « répétabilité »	Exige une expérience en modélisation des données pour assurer la « répétabilité »
Niveau de détail demandé dans les spécifications fonctionnelles	Oui	Oui, pour identifier chaque mouvement de données et accès aux fichiers

Tableau XVIII

Comparaison d'utilisation des résultats pour estimation

Résultat	IFPUG	COSMIC-FFP
Corrélation aux efforts à travers tous les domaines fonctionnels	Pour les systèmes informatiques de gestion il existe un appui industriel évident. Les données pour les logiciels scientifiques et les systèmes en temps réel sont limitées	Une recherche de 100 projets est approuvée pour les systèmes en temps réel. Les données pour les systèmes informatiques de gestion sont limitées.
Données industrielles	Domaine public et privé	Domaine de recherche ou privé
Certification internationale	Praticiens à IFPUG CFPS	Laboratoires ISO pour les tests sont planifiés
Sensibilité aux grandes variations dans le processus de complexité	La sensibilité maximale est une variation de deux fois	Pas de seuil ni de limite prédéterminée

Tableau XVIII (Suite)

Comparaison d'utilisation des résultats pour estimation

Résultat	IFPUG	COSMIC-FFP
Sensibilité aux processus qui déplacent beaucoup de données sans qu'ils accèdent aux groupes de données	La mesure de taille maximale exige les mouvements et l'accès aux données	La mesure de taille maximale n'exige pas de processus pour accéder aux groupes de données
Contribution des données dans la taille totale	Les données persistantes contribuent de 30 % de la taille totale, en y ajoutant la contribution des accès aux processus de données	Les données persistantes contribuent seulement via les accès aux processus de données
Compte l'accès multiple aux groupes de données	Oui	Oui

Tableau XIX

Comparaison des ressources

Ressource	IFPUG	COSMIC-FFP
Manuels	À acheter d'IFPUG À acheter d'ISO 390 pages (à 294,00 CHF)	Téléchargement gratuit du guide COSMIC d'implantation de la norme http://www.lrgl.uqam.ca/cosmic-ffp/manual.html À acheter d'ISO 75 pages (à 83,00 CHF)
Formation	Plusieurs vendeurs de cours dans le monde entier Certification	Plusieurs cours disponibles dans le monde entier Pas de certification
Outils	Plusieurs vendeurs d'outils Certification	Limités Pas de certification
Études de cas	Deux études de cas pour les fonctionnalités utilisateurs requises À acheter d'IFPUG	Cinq études de cas pour les fonctionnalités utilisateurs requises Téléchargement gratuit

5.10 Pourquoi COSMIC-FFP ?

Comme vu dans la section précédente, en parallèle à l'adoption de COSMIC-FFP (ISO/IEC 19761) comme norme, trois autres méthodes ont été approuvées par ISO comme normes spécifiques pour les mesures de la taille fonctionnelles des logiciels; IFPUG (ISO/IEC 20926), provenant des États-Unis, MkII FPA (ISO/IEC 20968), qui est proche de la méthode IFPUG, fournie par le Royaume-Uni et la méthode hollandaise NESMA (ISO/IEC 24570) qui est une variante de la méthode IFPUG. Ces trois dernières méthodes datent des années 1970-1980 et ont été conçues pour mesurer la taille fonctionnelle des logiciels de gestion. Elles sont utilisées pour mesurer la performance des logiciels et l'estimation de leur coût.

Seule la méthode COSMIC-FFP a apporté une modification structurelle à la méthode IFPUG, puisqu'elle peut être utilisée pour mesurer les logiciels de types autres que les systèmes d'informatique de gestion, tel que le système en temps réel.

Dans leur article « Mesure de la taille fonctionnelle des logiciels temps réel », Desharnais *et al.* montrent que dans la version 2.0 de COSMIC-FFP, la mesure se fait de façon individuelle pour les sous-processus de lecture et de mise à jour des fichiers logiques ainsi que pour l'envoi de l'information à l'utilisateur. Ainsi la méthode COSMIC-FFP a un niveau de granularité plus fin que la méthode IFPUG, parce qu'elle prend en compte le niveau pour les sous-processus. « Ce niveau de granularité plus fin prend toute son importance dans le cas des logiciels en temps réel, puisque leurs processus contiennent un nombre variable de sous-processus, chacun ayant une contribution individuelle à la mesure de leur taille fonctionnelle » (Desharnais *et al.*, 1998).

Souvent, on reproche à la méthode IFPUG de fournir des mesures fonctionnelles trop petites dans un environnement en temps réel (Galea, 1995; Capers Jones, 1996). Ceci est dû à la granularité de la méthode IFPUG qui n'est pas assez fine pour permettre de tenir

compte des sous-processus, car elle ne dispose pas d'un niveau pour ces derniers. Ainsi, les mesures fournies à la fin du processus ne reflètent pas de façon adéquate, selon ces auteurs, la taille réelle des fonctionnalités données par les logiciels mesurés. Cependant COSMIC-FFP prend en considération les sous-processus fonctionnels qui peuvent exister à l'intérieur d'un processus de contrôle. De plus, la méthode permet de déterminer tous les groupes de données envoyés, lus, écrits et reçus ce qui génère un résultat plus grand en mesure fonctionnelle et reflète le nombre de fonctionnalités réellement fournies à l'utilisateur du logiciel, qu'il soit humain, un autre logiciel ou un appareil qui interagit avec le logiciel.

Enfin, la méthode COSMIC-FFP est fondée sur une théorie solide et des décennies d'expérience internationale. Selon le *Software Measurement Services* (2001), COSMIC-FFP a été conçue dès le début pour être conforme à la norme ISO 14143 pour la MTF et pour être compatible avec les techniques modernes de spécification des exigences comme UML et le prototypage. Cette méthode de mesure est applicable à plusieurs types de logiciels. Elle reconnaît que le développement des logiciels modernes se sert des composants à différents niveaux dans une architecture de logiciel, ce qui permet d'atteindre des niveaux que d'autres méthodes ne tiennent pas compte en termes de fonctionnalité.

5.11 Sommaire

Ce chapitre a présenté un aperçu des normes portant sur la mesure de fonctionnalité des logiciels. Il a aussi situé la méthode lignes de code qui permet de mesurer le nombre de lignes de code dans un programme, par rapport aux MTF. Il a inclus également une description générale de la méta-norme de mesure de la taille fonctionnelle ISO 14143, puis des normes spécifiques qui ont été acceptées par ISO. Une comparaison détaillée entre COSMIC-FFP et IFPUG a été aussi présentée dans ce chapitre. Enfin, le chapitre a présenté les raisons de la sélection de COSMIC-FFP comme la méthode adoptée dans cette thèse pour bâtir les premiers étalons de mesure du logiciel :

1. La méthode COSMIC-FFP se démarque par le fait qu'elle permet de mesurer la taille fonctionnelle des systèmes en temps réel, des systèmes de gestion et des systèmes hybrides, assez tôt dans le cycle de développement, soit dès la phase de spécifications du logiciel.
2. COSMIC-FFP appliquée au début du cycle de développement peut être utilisée, par exemple, pour construire des modèles d'estimation de l'effort requis pour réaliser un projet informatique. Appliquée à la fin du cycle de développement, elle peut servir à bâtir des modèles de productivité ainsi qu'à comparer la productivité d'un projet informatique à un autre.
3. La méthode COSMIC-FFP permet aussi de résoudre quelques imperfections de la méthode des points de fonction pour les systèmes en temps réel et de simplifier la tâche de mesure. Elle est conforme à la norme ISO/IEC 14143-1, qui spécifie un ensemble de conditions obligatoires génériques pour une méthode de mesure pour être une méthode de mesure de taille fonctionnelle. En outre, la méthode est indépendante de toute technologie.
4. La méthode est compatible avec les techniques modernes de spécification des exigences comme UML et le prototypage.

Donc, pour ces raisons la méthode COSMIC-FFP sera utilisée pour la préparation des premiers étalons de mesure des logiciels

CHAPITRE 6

LA CONSTRUCTION D'UN ÉTALON

Ce chapitre permet de définir, en premier lieu, le terme étalon et donne un aperçu sur le processus de développement du premier étalon universel qui est le mètre. Puis, il décrit les étalons universels et spécifiques et présente les concepts d'étalonnage et de calibration. Il termine en introduisant le concept de traçabilité des étalons.

6.1 Introduction

La sonde spatiale Mars Climate Orbiter conçue pour analyser les variations du climat martien, s'est écrasée sur la planète Mars car une équipe exprimait les longueurs en mètres alors que l'autre les exprimait en pieds (I. Douglas et al., 1999). Cette anecdote illustre bien l'importance des étalons dans la vie humaine.

6.2 Généralités

La mesure est le fait d'attribuer une valeur numérique à une grandeur. C'est une notion indispensable en sciences tout comme dans la vie quotidienne. Elle permet d'exprimer une grandeur par un nombre, un symbole, un mot, un graphique, etc. Les nombres peuvent ensuite être utilisés par un modèle numérique.

L'assignation d'un nombre à une mesure est liée à la définition d'une unité basée sur un étalon. Par exemple, l'étalon de la distance est conservé au Bureau international des poids et mesures (BIPM, Paris). On compare toute longueur d'un objet à cet étalon d'une distance, à ses multiples ou ses sous multiples de sorte que la mesure conduit à dire que « l'objet fait n fois l'étalon de distance ».

Un concept appelé « unité » est défini pour chaque étalon. L'unité associée à la distance est le mètre (abrégé en « m »), de sorte que « un objet fait n mètres ». Une grandeur s'exprime par la formule $\text{grandeur} = \text{mesure} \times \text{unité}$ qui représente la mesure de la grandeur dans une unité donnée.

Toute mesure a, en pratique, une marge d'erreur car il n'y a pas de processus parfait de mesurage. Par ailleurs, une mesure n'est considérée comme correcte que si on lui associe une estimation d'erreur. Par exemple, la poutre mesure 3 m de long à 0,005 m près. Cette estimation de la précision s'appelle « erreur absolue » ou « incertitude absolue » et elle est exprimée avec la même unité que celle utilisée pour exprimer la mesure de la grandeur. « Dans le cas des modèles numériques, la mesure doit être associée à une incertitude et à un intervalle confiance » (Encyclopédie libre Wikipedia, 2005).

Selon Henry (1997), un modèle est une représentation abstraite, simplifiée d'un objet du monde réel, ou d'un système de relations, ou d'un processus évolutif, issus d'une description de la réalité. La métrologie est dotée des méthodes et des techniques pour paramétrer un modèle destiné à représenter la réalité. La réalité concernée est souvent la réalité physique, mais peut aussi être économique, sociologique, etc

6.3 Qu'est ce qu'un étalon ?

Selon le VIM (ISO, 1993), un étalon est une « mesure matérialisée, appareil de mesure, matériau de références ou système de mesure destiné à définir, réaliser, conserver ou reproduire une unité ou une ou plusieurs valeurs de grandeur pour servir de référence ». Un étalon doit être précis, exact, reproductible et universel (Antoine et al., 2003).

6.3.1 Création du premier étalon universel : Le mètre

Le mètre date d'un peu plus de deux siècles, ce qui est jeune à l'échelle de l'humanité, mais par rapport au domaine de la mesures des logiciels, c'est très ancien. Alors, avant lui quel était le système de mesure?

Jusqu'à la Renaissance européenne, les outils de mesure dérivait des parties du corps humain, comme le pied ou le pouce pour les longueurs. Souvent le standard était les membres (pieds, pouces, coudes) du roi. De plus, il y avait plusieurs systèmes de mesures et c'était bien là le problème. « Chaque région de France possède son propre système : en Bretagne, on mesure en perches, à Marseille en palmes, à Paris en pieds ailleurs encore en toises, en pouces, en lignes, en brasses, en coudées, en empan, ... Il en est de même pour les mesures de masse où la livre est par exemple plus légère à Toulouse qu'à Strasbourg » (M. Lange, 2005). Dans ces conditions, les échanges commerciaux sont particulièrement complexes.

Pour remédier à cette diversité et à cette complexité dans les échanges commerciaux, l'Assemblée Nationale française décidait en 1790 de faire un système de mesure unique. Mais la question qui se posait a été de comment définir le mètre? Le projet fut confié à des savants bien connus de l'époque. Ces savants définirent, par convention et par consensus, le mètre comme le dix millionième du quart du méridien terrestre. L'idée reposait sur une référence bien connue par tous et qui serait acceptée de tous : la Terre. Mais en pratique, la mise sur pied de cette convention et de ce consensus a été particulièrement longue et difficile. Le projet se concrétisera dans des conditions difficiles sur une période de six ans, de 1792 à 1798 : pour mesurer la circonférence du méridien à l'équateur, il faudra franchir des montagnes, confronter des conditions climatiques sévères, la réticence des habitants et surtout une situation politique agitée.

Enfin, en 1795, le mètre remplaçait officiellement toutes les unités existantes. « Le mot mètre vient du grec metron qui veut dire mesure » (Picchiottino et al., 2000). On introduit ses multiples et ses sous-multiples. Le mètre s'étendra d'abord en Europe puis dans la plupart des pays du monde, sauf dans quelques pays anglo-saxons où des unités telles que le « pied » sont encore utilisées. L'amélioration de la définition du mètre s'est poursuivie au cours des siècles jusqu'en 1983 avec l'adoption de la plus récente

définition dans la convention consensuelle du mètre: c'est la distance parcourue dans le vide par la lumière en $1/299\,792\,458$ e de seconde.

6.3.2 Les étalons universels et spécifiques

Les étalons dits universels sont les étalons de la Convention du Mètre. Ils définissent les unités du système international (SI). Selon le BIPM (2005), le SI est fondé sur un choix de sept unités de base bien définies et considérées par convention comme indépendantes du point de vue dimensionnel : le mètre, le kilogramme, la seconde, l'ampère, le kelvin, la mole et la candela. Bien qu'ils servent à des déterminations précises, ces étalons ne sont pas facilement utilisables, vu les précautions sévères exigées pendant leur manipulations sur les lieux où doit se faire l'étalonnage. Alors, il faut des étalons dits « spécifiques », plus simples d'utilisation, qui sont eux-mêmes calibrés à partir des étalons universels. Par exemple, le mètre (étalon de distance) du BIPM sert de référence pour des étalons spécifiques de distance qui permettent d'étalonner l'instrument « règle » chez les fabricants.

Les étalons peuvent être :

- un objet inaltérable, comme la masse étalon;
- un phénomène physique, comme l'étalon seconde, l'étalon mètre, l'étalon intensité du courant électrique;
- une réaction chimique, comme l'électrode normale à hydrogène ou l'électrode au calomel saturée en KCl utilisée en électrochimie.

Toutefois, un utilisateur, un organisme ou un groupe de recherche peut fabriquer lui-même ses propres étalons. Par exemple, dans un domaine où le corpus des connaissances n'est pas bien défini, comme le génie logiciel, un laboratoire qui a les compétences nécessaires pour développer des étalons peut suggérer des étalons de mesure des attributs des logiciels : « En analyse chimique, les utilisateurs fabriquent souvent des solutions à partir de produits purs pour étalonner leurs appareils d'analyse » (Encyclopédie libre Wikipedia, 2005).

6.3.3 Organismes de normalisation

Le rôle du développement et de la reconnaissance des étalons est délégué à des organismes de normalisation (standardisation en anglais).

Au niveau international, il existe deux organismes compétents pour définir des étalons internationaux, à savoir :

le Bureau international des poids et mesures de France auquel adhèrent 51 pays, est un organisme officiel;

ISO – International Organization for Standardization, qui réunit les organismes nationaux de normalisation.

Chaque pays a, par la suite, son propre organisme de normalisation. Par exemple :

- Conseil canadien des normes — CCN au Canada
- Association française de normalisation — AFNOR en France
- National Institute for Science and Technology — NIST aux États-Unis
- Deutsches Institut für Normung — DIN en Allemagne
- Institut belge de normalisation — IBN en Belgique
- British Standards Institution — BSI au Royaume-Uni.

6.4 Étalonnage et calibration

Le vocabulaire international des termes fondamentaux et généraux de métrologie (VIM) définit l'étalonnage comme suit : « L'étalonnage est l'ensemble des opérations établissant dans des conditions spécifiées, 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 » (ISO, 1993).

Dans certains cas, le phénomène que l'on veut évaluer n'est pas homogène, il faut donc faire plusieurs mesures. Par exemple, si l'on veut mesurer un système informatique à plusieurs composantes, il faut le faire sur plusieurs parties du système car chaque partie

peut avoir ses spécificités (langage de programmation utilisé, expérience du codeur, etc.). Si l'on veut connaître la composition chimique d'une huile dans des citernes de stockage, il faut faire des prélèvements en plusieurs endroits (notamment en raison de la décantation, les produits lourds sont au fond et les produits légers au-dessus).

Cependant, la calibration est simple. Elle consiste à passer un objet de référence pour vérifier que l'objet mesuré a des propriétés proches de l'objet de référence. En anglais, le terme standard ou standard-étalon signifie « étalon », et le terme calibration signifie « étalonnage ».

6.5 La traçabilité des étalons

Lorsqu'on fait une mesure, on est sensé tout de même se rattacher quelque part à un étalon, ou à plusieurs étalons par un chemin compliqué : en métrologie, le terme correspond au concept de traçabilité. Les étalons doivent être « traçables » aux éventuels étalons nationaux et internationaux de mesure. Selon l'Institut des étalons nationaux de mesure (2004), la traçabilité implique la possibilité de relier les résultats d'une mesure individuelle, comportant une incertitude donnée, par l'intermédiaire d'une chaîne non interrompue de comparaisons, à une référence déterminée, habituellement les étalons nationaux de mesure ou à des étalons intrinsèques basés sur des constantes naturelles fondamentales dont les valeurs sont attribuées ou acceptées au niveau national ou international. La figure 8, adaptée de Preben et al. (2003), montre la chaîne de traçabilité des mesures. À signaler que l'incertitude augmente en bas de la chaîne de traçabilité.

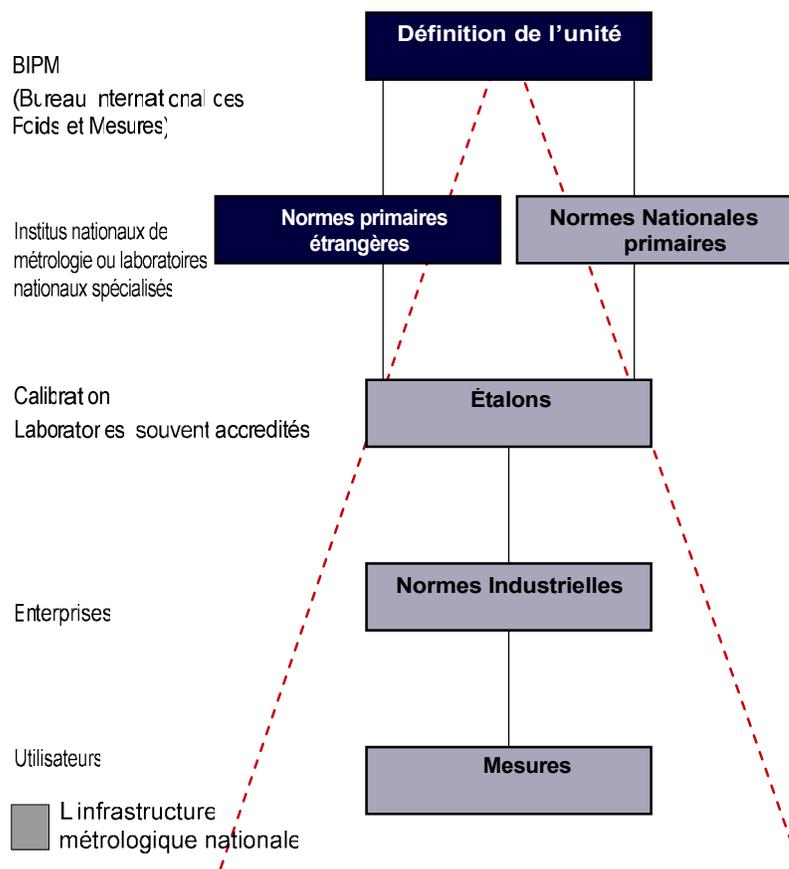


Figure 8 La chaîne de traçabilité - adaptée de Preben et al. (2003)

Toujours selon l'Institut des étalons nationaux de mesure (Bureau de communications, 2004), un résultat d'étalonnage devrait comprendre les données suivantes pour permettre d'établir adéquatement une chaîne permettant de vérifier la traçabilité :

- la valeur attribuée;
- l'incertitude déclarée;
- l'identité de l'étalon ou des étalons utilisés lors de l'étalonnage.

Des précisions sur les conditions du milieu de l'étalonnage auxquelles des facteurs de correction doivent être appliqués si l'étalon ou l'équipement est utilisé dans des conditions différentes du milieu.

6.6 Conclusion

Ce chapitre a défini le terme d'étalon et il a présenté un aperçu sur le processus de création du mètre, des étalons universels et spécifiques, de l'étalonnage et de la calibration ainsi que de la traçabilité des étalons.

Le chapitre suivant définira la problématique de recherche à traiter dans cette thèse, la question de recherche soulevée et l'objectif derrière cette recherche.

CHAPITRE 7

PRÉSENTATION DU TRAVAIL

Ce chapitre expose en premier lieu les sous-problématiques de cette thèse, sa problématique générale de recherche et sa problématique spécifique. Par la suite, le chapitre présente la question de recherche choisie et l'objectif visé dans ce projet de recherche.

7.1 Introduction

Le domaine de la mesure en génie logiciel est encore en émergence. Il s'ensuit que les mesures conçues et proposées pour mesurer les logiciels ne font pas référence à certains des concepts fondamentaux de la métrologie, soit à des étalons de mesure et à des standards de mesure. L'utilisation des étalons de mesure dans le domaine des logiciels et plus précisément pour les mesures de la taille fonctionnelles des logiciels n'a pas encore été, à notre connaissance, mis en pratique puisque, d'une part, de tels étalons n'existent pas et que d'autre part, il n'est pas encore connu comment concevoir un design d'étalons de mesure dans le domaine du logiciel.

Pour les domaines d'ingénierie plus matures, le corpus des connaissances de la mesure est une composante de base pour la réalisation des méthodes et des outils de mesure. « Si l'on considère que le génie logiciel est une discipline mature et reconnue de l'ingénierie, il est nécessaire qu'elle soit supportée par des mesures, des méthodes de mesure et des modèles quantitatifs et descriptifs bien testés » (A. Abran, 2000).

La problématique de recherche portera sur la conception des étalons de mesure pour la taille fonctionnelle des logiciels avec la méthode COSMIC-FFP (ISO 19761). Le but à long terme est que ces étalons soient présentés et acceptés par les experts ISO et les industriels du domaine.

7.2 Problématique de recherche

Dans les disciplines d'ingénierie bien matures, un résultat de mesure est accepté s'il fait appel à une ou à plusieurs références. On se demande toujours qui donne ce résultat de mesure et quel est son référentiel. Une mesure est toujours relative à un référentiel. Un résultat absolu de mesure, sans référentiel, n'est pas un résultat au sens métrologique généralement accepté en science, mais il peut être une affirmation personnelle. Il suffit de constater les interminables discussions et débats qui ne peuvent avoir de résultat puisqu'il ne peut y avoir de conclusion si, pour les différents intervenants, le référentiel est différent. En effet, sans référentiel, un résultat est mystérieux.

C'est pourquoi, avant toute discussion, faut-il mettre au clair les systèmes de référence utilisés sinon il devient difficile d'arriver à un ensemble de conclusions (résultats de mesure) acceptables pour tous.

Le domaine du génie logiciel présente plusieurs caractéristiques qui font de l'application de la théorie de mesure à ce domaine une tâche très complexe. Le génie logiciel est une science encore jeune dont les caractéristiques et les attributs diffèrent beaucoup de ceux des autres produits des industries classiques tels que les maisons, les vêtements ou les voitures. Les mesures en génie logiciel sont des concepts nouveaux auxquels on peut ajouter ou retrancher des sous-concepts. Par exemple, des attributs de logiciels tels que la complexité, la taille et la fiabilité ne sont pas encore clairement définis. Il a été vu que certains axiomes de Weyuker (1988) exigent que l'attribut « complexité » soit défini comme compréhensibilité; tandis que pour d'autres, tel que McCabe (1976), la complexité est liée à la testabilité du code source. Par conséquent, on retrouve dans la littérature, pour le même attribut, une multitude de définitions qui dépendent des besoins de l'environnement impliqué.

La problématique de la recherche choisie pour ce travail de recherche porte sur la réalisation de références métrologiques pour les MTF des logiciels. Elle est composée de trois sous-problématiques :

A) Sous-problématique liée à la nature du produit à mesurer qui est le logiciel

Les caractéristiques suivantes d'un logiciel le rendent difficile à mesurer:

1. Un logiciel est un produit atypique comparativement aux produits industriels. On peut parler d'avions type, de voitures type, mais il est pour le moment impossible de parler de logiciel type. Les logiciels présentent de grandes différences à la fois en taille, en complexité, en techniques de conception, en méthodes de test, en domaine d'applications, etc.
2. Un logiciel est un produit continuellement évolutif étant destiné à satisfaire un ensemble de besoins appartenant à un monde réel qui évolue. Ces besoins évoluent et le logiciel que l'on développe est amené à évoluer, sinon il est désuet et sera remplacé par un autre logiciel répondant mieux à ces nouveaux besoins. Les logiciels coûtent cher du fait de leur taille et de leur complexité inhérente;
3. La nature des logiciels comme produits d'apparence intangibles, a rendu difficile à date le design de méthodes de mesure ayant toutes les propriétés des mesures au sens métrologique général accepté par ISO.

La figure 9 suivante résume cette sous-problématique :

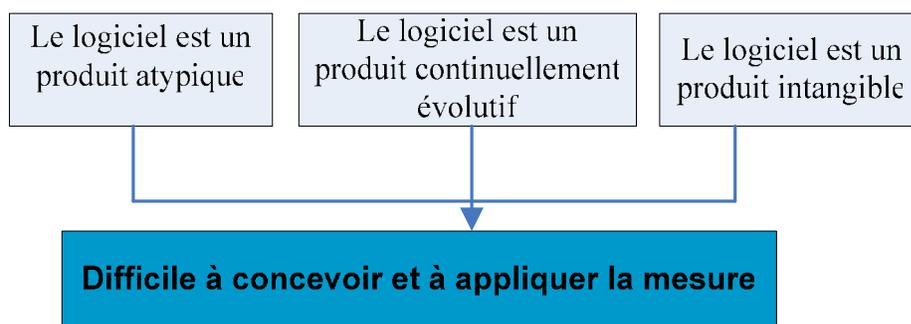


Figure 9 Le logiciel est difficile à mesurer

B) Sous-problématique liée au concept de la mesure en génie logiciel

Dans la littérature du génie logiciel, les mesures sont souvent définies de façon imprécise comme concepts. Par exemple, même le mot « métrique » a plusieurs définitions. Bieman *et al.* (1996) illustrent cette confusion par les phrases suivantes :

«Interestingly, the term metric has more than one meaning in software measurement. It is used in at least the three ways depicted below:

- A number derived from a product, process or resource. For example one hears about the metric number of function points or the metric lines of code LOC per programmer month;
- A scale of measurement. For example one hears about a proposed nominal scale i.e. classification of software failures as a metric;
- An identifiable attribute. For example one hears about the metric portability of programs or the metric coupling in designs, even though no number or function is necessarily attached. »

Ces « métriques » n'ont pas forcément de relations étroites avec la mesure telle que définie dans la métrologie classique des disciplines d'ingénierie. Par exemple, les métriques proposées par des chercheurs tels que Kemerer (1987), Kafura *et al.* (1985) n'ont pas ni une base théorique ni une validation expérimentale importante. Ceci génère des occurrences de « métriques » non satisfaisantes pour les exigences proposées par la norme ISO 14143-3 : la répétitivité, la reproductibilité, l'exactitude, la convertibilité, le seuil de tolérance et l'applicabilité aux domaines fonctionnels.

Dans leur article « Metrology, Measurement and Metrics in Software Engineering », Abran *et al.* (2003) ont montré qu'en génie logiciel, la majorité des propositions de mesure ne se réfèrent pas à un référentiel, ne suggèrent pas un instrument de mesure et n'adoptent pas de méthodes standards de mesure.

Quand les mesureurs abordent des spécifications fonctionnelles à mesurer, ils doivent faire face à des difficultés dans la détermination de ce qu'il faut mesurer et dans l'utilisation de la méthode de mesure. Par exemple, Desharnais (2003) écrit que «

l'application de la méthode de mesure est une phase du processus de mesure à la fois dépendante de la qualité des données à collecter et de la capacité du mesureur à modéliser le logiciel à mesurer ».

La figure 10 suivante résume cette sous-problématique :

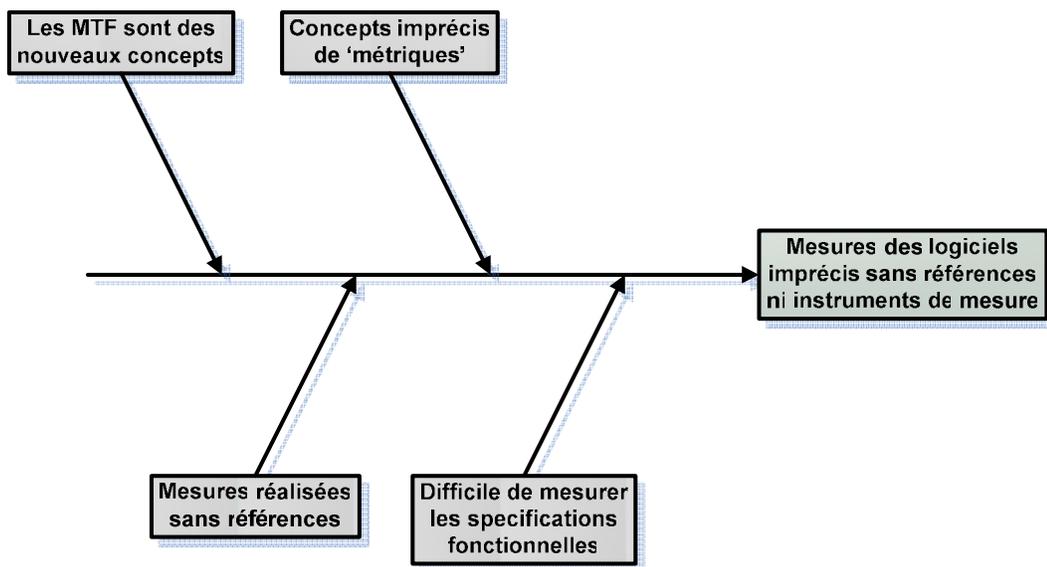


Figure 10 Mesures des logiciels sans référence

C) Sous-problématique liée à la documentation des logiciels.

L'identification des processus (ou composants élémentaires) d'un système informatisé repose sur la partie généralement non automatisable d'une application, c'est-à-dire sa documentation fonctionnelle (A. Abran, 1994). Selon Desharnais et Abran (2001), l'application de la méthode de mesure fonctionnelle est difficile à cause des facteurs suivants :

- différentes sources d'information pour obtenir ce qui est nécessaire à la mesure;
- sources peu homogènes puisque l'origine est fréquemment différente;

- documentation souvent incomplète.

De plus, selon Nishiyama (1994), la qualité de la documentation joue un rôle important pendant le processus de mesure des logiciels. Nishiyama (1994) indique qu'il faut jusqu'à quatre fois plus d'effort pour mesurer un logiciel dont la documentation est non structurée, par rapport à un logiciel dont la documentation est bien structurée. En effet, plus la documentation est claire et concise plus l'effort de mesure de la taille fonctionnelle des logiciels est faible. De plus, l'expérience (S.M. Ndagijimana, 2002) a montré qu'une qualité médiocre de documentation génère un problème de cohérence des résultats entre mesureurs de la taille fonctionnelle des logiciels.

Enfin, plusieurs chercheurs (V. Prince, 1996) ont constaté que la documentation des logiciels est souvent incomplète, obsolète et même parfois erronée. La figure 11 suivante illustre cette sous-problématique.

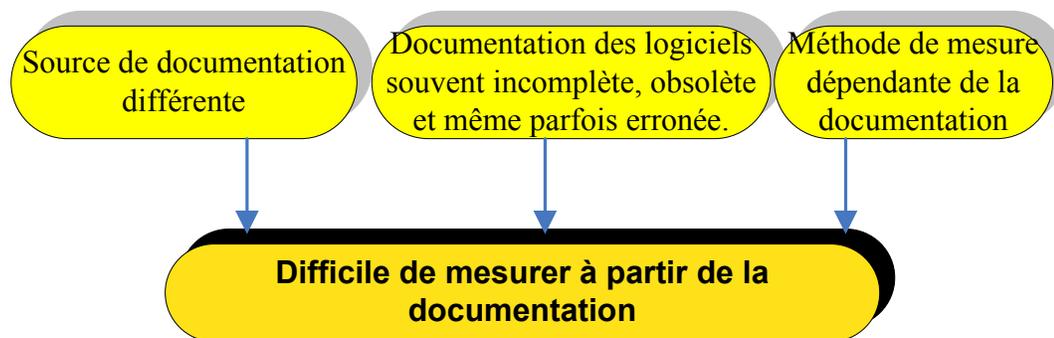


Figure 11 Difficultés de mesurer à partir de la documentation

Ainsi, les problèmes de documentation, lors de la conception et de la mise en place des logiciels, affectent la compréhension des logiciels pour des fins de mesure.

Il ressort de la revue de littérature que le design et l'utilisation des mesures en génie logiciel sont des activités complexes, manquent de références et de standards, et

dépendent de la qualité de la documentation. La Figure 12 illustre la problématique générale traitée dans cette thèse :

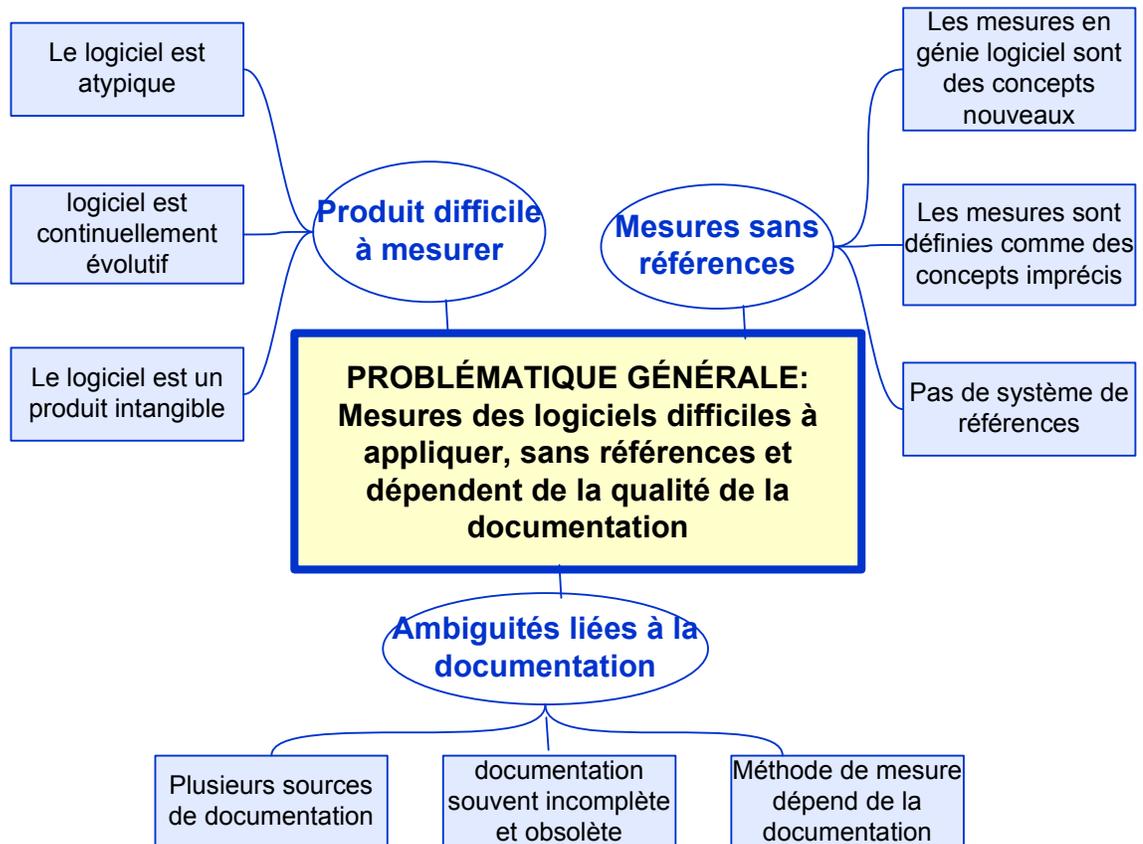


Figure 12 Problématique de recherché

Pour contribuer à résoudre la problématique générale, **cette recherche s'attaque à une problématique bien spécifique qui est l'absence de références pour la mesure des logiciels.**

De ce fait, l'introduction d'étalons de mesure dans ce domaine :

- servira principalement de point commun, même si le produit est continuellement évolutif, intangible et atypique;
- offrira un point de référence pour les mesureurs du logiciel;

- permettra de remonter aux unités de référence et donc de parler d'une même voix.

Dès lors, il devient très important de développer, pour les mesureurs, un référentiel composé d'étalons de mesure des logiciels, éléments indispensables d'une structure métrologique permettant d'avoir une référence commune et apportant plus de confiance dans le processus de mesure. En effet, les étalons facilitent la réalisation des résultats de mesures sur des bases communes.

7.3 Questions de recherche

Les constats précédents amènent à étudier les questions suivantes :

Question 1: Comment aborder le design d'étalonnage en génie logiciel?

Question 2 : Comment développer une première version d'étalons de mesure de la taille fonctionnelle des logiciels qui puissent être utilisés comme références?

7.4 Objectif de recherche

Les mesures en génie logiciel aident les différents acteurs à comprendre les artefacts des logiciels, à prédire le comportement du logiciel dans son contexte, à contrôler le processus de développement et à améliorer la qualité du logiciel. Selon Idri (2003), les mesures de logiciels devraient permettre d'élever les activités de production des logiciels au rang d'une industrie dont les technologies seraient contrôlées et maîtrisées, afin d'améliorer la qualité des logiciels et de réduire leurs coûts de développement et de maintenance.

L'objectif principal dans cette recherche, consiste à construire un référentiel pour la mesure fonctionnelle en génie logiciel. L'existence d'un tel référentiel permettra la standardisation des mesures de la taille fonctionnelle des logiciels, facilitera l'utilisation des mesures de la taille fonctionnelle des logiciels, réduira les efforts des mesureurs dans

leurs tâches et, accessoirement, contribuera à évaluer la qualité de la documentation du logiciel mesuré.

Dans le développement des logiciels, l'incertitude sur la qualité des mesures est importante compte tenu qu'il s'agit d'une mobilisation de moyens et de compétences dont l'évaluation s'avère délicate. Les étalons permettent de fournir aux clients et aux professionnels des éléments communs afin de faciliter le processus de la mesure et son évaluation.

7.5 Sommaire

Les sous-problématiques de la recherche et sa problématique générale sont exposées au début de chapitre. Puis la question de recherche ainsi que son objectif sont présentés à la fin.

Le chapitre suivant commence par présenter la définition de ce projet. Cette définition comporte la motivation pour faire la recherche, les utilisateurs de ce projet, les outils de la recherche et ses limites. Puis, il expose la méthodologie adoptée pour résoudre la problématique spécifique de la thèse. Une partie du contenu du chapitre se concentre sur le développement du site web, sur la méthode de liaison d'une mesure au référentiel. Le chapitre suivant s'achèvera en décrivant la méthodologie de développement du référentiel et en énumérant les partenaires de la recherche.

CHAPITRE 8

LA MÉTHODOLOGIE DE RECHERCHE

Ce chapitre présente le design de la méthodologie de recherche pour aborder la problématique choisie. Les étapes principales de la méthodologie sont décrites afin d'expliquer la façon adoptée pour conduire ce travail de recherche.

8.1 Introduction

Selon Merriam (1988), les données communiquées par les mots sont « qualitatives », alors que les données présentées par des chiffres sont « quantitatives ». Ce travail permet de développer et de proposer, pour la première fois en génie logiciel, des étalons de mesures des logiciels. Pour cela, une méthodologie exploratoire et qualitative est adoptée.

Une étude exploratoire est entreprise lorsqu'il n'existe pas d'information sur la façon de résoudre des problèmes similaires (U. Sekaran, 1992) à la problématique choisie dans cette recherche. Ce qui est bien le cas dans ce travail. Elle est qualitative puisqu'elle traite la conception des étalons, ce qui est un travail qui ne demande pas des calculs et des statistiques mais plutôt des analyses. En somme, cette recherche se situe initialement dans la mouvance de la recherche qualitative. Une recherche qualitative doit réunir un certain nombre de caractéristiques : inductive, globaliste, valide et objective (M.Q. Patton, 1990; R.C. Bogdan *et al.*, 1992; S.B. Merriam, 1988; S.J. Taylor *et al.*, 1984; Y. Lincoln *et al.*, 1985) pour être menée avec rigueur et rapidité. Cette recherche tient compte de ces caractéristiques afin de suivre une méthodologie efficace pour réaliser un premier référentiel pour la mesure des de logiciels avec ISO 19761(COSMIC-FFP).

En génie logiciel, il existe un cadre méthodologique bien connu par les chercheurs du domaine. C'est le cadre expérimental de Basili *et al.* (1991). Selon ces auteurs, le cadre

permet d'analyser la majorité des travaux expérimentaux en génie logiciel. Ils définissent le cadre comme une aide pour structurer les processus expérimentaux :

« The framework for experimentation is intended to help structure the experimental process and to provide a classification scheme for understanding and evaluating experimental studies » (V.R. Basili *et al.*, 1986).

Comme sa définition l'indique, le cadre est conçu pour des projets de recherche expérimentale où le corpus de connaissance est bien développé. Mais pour une recherche exploratoire où le corpus des connaissances n'est pas mature, une modification du cadre est bien requise. En effet, pour mieux aborder les problèmes reliés à la recherche exploratoire, des adaptations réussies du cadre ont été faites par les chercheurs du GÉLOG (A. Abran *et al.*, 1999; P. Bourque *et al.*, 1991; S. Wolff, 1999). Selon ces auteurs, le cadre de Basili constitue un excellent outil pour définir le but et les objectifs du projet avant de s'aventurer plus loin. L'adaptation proposée par Abran *et al.* (1999) conserve les quatre phases du modèle initial. Une telle adaptation facilite la documentation du processus de recherche exploratoire et assure sa répétabilité qui est considérée comme une exigence fondamentale de ce genre de travail. Le cadre est composé de quatre catégories correspondant aux phases du processus expérimental : définition, planification, exécution et interprétation.

Comme mentionné auparavant, cette recherche exploratoire ne comporte pas de collectes de données formelles ni de statistiques; les étapes d'échantillonnage et des statistiques sont enlevées du cadre dans un contexte de recherche exploratoire. Les sections suivantes présentent le cadre adapté pour cette recherche exploratoire.

8.2 Définition du projet

Cette section présente le contenu de la première phase du cadre de Basili adaptée au contexte de ce travail. Cette section décrit la motivation, les utilisateurs, les objectifs et les limitations de la thèse.

8.2.1 Motivation

La motivation de ce projet de recherche est de fournir une première version des références pour les mesures de la taille fonctionnelle des logiciels et par là même contribuer au développement futur des standards pour la mesure des logiciels. Ceci peut aider dans la maturité du génie logiciel.

L'objet de l'étude est la mesure des logiciels portant sur les fonctionnalités utilisateurs requises qu'un logiciel doit livrer aux utilisateurs.

Le propos du travail concerne le développement d'un référentiel avec la norme ISO 19761 (COSMIC-FFP) pour les mesures de la taille fonctionnelle des logiciels sur la base des fonctionnalités utilisateurs requises (FUR), et tel que documentées dans le document ISO (ISO/IEC 14143-4, 2000), le matériel de formation sur le processus RUP (Rational Software, 2002a, 2002b) et les spécifications de l'application *Rice Cooker* (GELOG, 2003).

8.2.2 Utilisateurs

Les groupes d'utilisateurs qui pourraient bénéficier de cette recherche sont les suivants :

Les mesureurs des logiciels en général et en taille fonctionnelle en particulier, qui disposeront de nouvelles références;

Les experts en programmes de mesures des logiciels, qui pourront trouver dans cette recherche de nouveaux standards de travail;

Les organisations de production d'étalons et d'instruments de mesure des logiciels, qui pourront bénéficier de ce travail pour concevoir leurs propres produits;

Les chercheurs en mesure des logiciels qui profiteront du sujet traité dans cette thèse pour développer plusieurs thématiques de recherche et de nouveaux horizons de recherche.

8.2.3 Objectif

Tel que spécifié lors de la proposition de cette thèse, l'objectif principal de ce travail de recherche est le développement d'un référentiel, au sens métrologique, pour les mesures de la taille fonctionnelle des logiciels avec la norme ISO 19761 (COSMIC-FFP).

8.2.4 Les outils de travail pour faire la recherche

Pour atteindre cet objectif, la recherche a besoin des normes ISO 14143 et ISO 19761, le logiciel *FrontPage* pour le développement d'un site web, l'environnement RUP, les contributions d'experts internationaux en MTF des logiciels et des contacts avec le groupe de travail (WG12) d'ISO/IEC JTC1/SC7 spécialisé en mesure de la taille fonctionnelle des logiciels.

Selon ISO « toutes les publications de l'ISO sont protégées par le droit d'auteur. Par conséquent, sauf prescription différente, aucune partie d'une publication de l'ISO ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'éditeur » (ISO, 2004). Puisque le GÉLOG assume le leadership pour le développement de normes internationales en génie logiciel et des systèmes et il a la présidence et le secrétariat du groupe ISO responsable du génie logiciel, et ce depuis

2001, l'équipe de recherche impliquée dans cette thèse, membres du GÉLOG, ont un accès aux normes citées ci-dessus pour des fins de préparation aux contributions canadiennes au processus ISO de normalisation. Le GÉLOG peut aussi diffuser les travaux de recherche issus de l'utilisation des normes ISO à des fins éducatives et de recherche.

Les communications avec les autres membres internationaux de ce groupe ISO de travail, spécialisé dans la mesure de la taille fonctionnelle des logiciels, est facilité par une participation de l'équipe de cette recherche comme membre canadien du groupe de travail WG 12.

8.2.5 Limites de la recherche

Le travail exigé comprend plusieurs difficultés de recherche sur le terrain, incluant des collaborations à distance avec des experts internationaux.

Les limites principales de cette recherche exploratoire sont les suivantes :

L'échantillon des FUR disponibles et mesurées qui sont relativement hétérogènes;

Le nombre relativement restreint des FUR disponibles n'est pas nécessairement représentatif de tous les types de logiciels existants;

L'étude se base seulement sur huit ensembles de FUR différents, cinq provenant du rapport technique ISO TR 14143-4 (soit : *Automatic Line Switching System*, *Hotel Reservation System*, *L-Euchre Application*, *SAGA System* et *Valve Control System*). Un autre ensemble de FUR appartient à l'application *Rice Cooker* dont le copyright appartient au GÉLOG et qui est utilisé avec permission. Les deux autres appartiennent à la compagnie *IBM-Rational C-Registration System* et *Collegiate Sports Paging System*, pour faire des mesures fonctionnelles et ressortir des résultats vérifiés.

Cette recherche porte sur la préparation d'un référentiel pour une mesure des logiciels bien spécifique, soit la méthode COSMIC-FFP (ISO 19761).

Le référentiel ne représente qu'un ensemble limité d'étalons qui ne couvre que trois types de logiciels, à savoir les systèmes en temps réel, les systèmes de gestion et les systèmes hybrides des deux types précédents.

8.3 Étapes de la méthodologie

Cette section est divisée en cinq sous-sections représentant les étapes de déroulement de cette recherche exploratoire.

8.3.1 Identifier la méthode de mesure pour développer des étalons

D'après la revue de la littérature, COSMIC-FFP constitue une excellente méthode de mesure des logiciels. Les raisons de ce choix sont les suivantes :

Seule la méthode COSMIC-FFP peut être utilisée pour mesurer les logiciels de types autres que les systèmes d'informatique de gestion, tel que le système en temps réel.

La méthode COSMIC-FFP a un niveau de granularité plus fin que la méthode IFPUG. Ce niveau de granularité plus fin prend toute son importance dans le cas des logiciels en temps réel, puisque leurs processus contiennent un nombre variable de sous-processus, chacun ayant une contribution individuelle à la mesure de leur taille fonctionnelle. (Desharnais *et al.*, 1998)

La méthode permet de déterminer tous les groupes de données envoyés, lus, écrits et reçus, ce qui génère un résultat plus grand en mesure fonctionnelle et reflète le nombre des fonctionnalités réellement fournies à l'utilisateur du logiciel.

Contrairement à d'autres méthodes telles que IFPUG, l'utilisateur de COSMIC-FFP peut être un humain, un autre logiciel ou un appareil qui interagit avec le logiciel.

La méthode COSMIC-FFP est fondée sur une théorie solide et des décennies d'expérience internationale (*Software Measurement Services*, 2001).

COSMIC-FFP a été conçue dès le début pour être conforme à la norme ISO 14143 pour la MTF et pour être compatible avec les techniques modernes de spécification des exigences comme UML et le prototypage (*Software Measurement Services*, 2001).

Cette méthode COSMIC-FFP reconnaît que le développement des logiciels modernes se sert des composants à différents niveaux dans une architecture de logiciel, ce qui permet d'atteindre des niveaux que d'autres méthodes ne tiennent pas en compte en termes de fonctionnalité (*Software Measurement Services*, 2001).

L'utilisation des étapes de la mesure fonctionnelle proposée dans le Manuel de mesure COSMIC-FFP (A. Abran *et al.*, 2002) fait partie de la démarche méthodologique. D'ailleurs, le design d'un étalon est basé sur les différentes parties de la méthode COSMIC-FFP.

La méthode de mesure COSMIC-FFP (ISO 19761), une méthode de MTF des logiciels, consiste à appliquer un ensemble de règles et de procédures sur les fonctionnalités livrées aux utilisateurs et qui sont décrites via les Fonctionnalités Utilisateurs Requises (FUR) afin d'aboutir à un nombre, son unité est le Cfsu (*COSMIC functional size unit*), représentant une valeur quantitative de la taille fonctionnelle du logiciel.

8.3.2 Réalisation d'un étalon

La réalisation d'un étalon, dans ce contexte de recherche, sera faite par l'application d'une méthode de mesure des logiciels, COSMIC-FFP, sur un ensemble des

fonctionnalités utilisateurs requises. La Fonctionnalité utilisateur requise (FUR) est un sous-ensemble des besoins de l'utilisateur. Le FUR représente les pratiques et les procédures de l'utilisateur que le logiciel doit accomplir pour répondre aux besoins de celui-ci. Le FUR exclut les besoins en matière de qualité et les besoins techniques (ISO/IEC 14143-1, 1998)

8.3.3 Ajout des diagrammes UML

En pratique, l'application de mesures fonctionnelles sur les logiciels n'est pas simple et demande un apprentissage et une expérience en méthode de mesure et en interprétation des artefacts des logiciels.

En fait, dans l'application de la méthode COSMIC-FFP, le mesureur doit déterminer à partir des artefacts disponibles (souvent de la documentation incomplète), les couches du logiciel à mesurer, sa frontière, ses utilisateurs, les événements déclencheurs, les processus fonctionnels, les groupes de données et les mouvements de données. Ce travail demande au mesureur de faire des interprétations sur plusieurs variables, telles que les fonctionnalités utilisateurs requises (FUR), qui ne sont pas toujours claires. Dans ce cas, les interprétations sont difficiles à faire particulièrement pour les mesureurs novices ou les mesureurs qui n'ont pas une bonne connaissance du domaine fonctionnel du logiciel à mesurer.

La compréhension de ces différents concepts et les moyens de les appliquer pour arriver à les mesurer, font de la mesure fonctionnelle une tâche complexe et difficilement automatisable (J.M. Desharnais *et al.*, 2001).

Pour contribuer à uniformiser l'instanciation de la procédure de mesurage lors de la construction du référentiel, il a été choisi pour cette thèse de développer des diagrammes UML, tels que des cas d'utilisation et de diagrammes de séquences pour le logiciel avant de commencer le mesurage cela afin d'améliorer la compréhension de ses fonctions.

Ceci facilite, d'une part, la tâche de modélisation du mesureur et, d'autre part, lui permet d'avoir une documentation plus normalisée et plus précise pour lui permettre tant la répétabilité des résultats de mesure que la transparence et la documentation des étapes intermédiaires du processus de mesurage.

Dans cette recherche, avant de commencer le processus des mesures et après la compréhension des FUR, des diagrammes UML sont donc ajoutés afin de mieux comprendre les fonctionnalités fournies par le logiciel à mesurer. Les principaux diagrammes UML utilisés sont le diagramme de cas d'utilisation et le diagramme de séquence. Le cas d'utilisation traduit les fonctions du système. Un diagramme de séquence est une forme de diagramme comportemental qui permet de spécifier les interactions qui existent entre les objets du système.

8.3.4 Déterminer le gabarit du rapport d'étalon

Pour présenter des mesures de logiciels avec la méthode COSMIC-FFP, le Consortium COSMIC avait élaboré un gabarit initial du rapport pour la présentation des résultats de mesure (LRGL, 2000).

Le gabarit initial comprenait les sections suivantes (figure 13) :

Introduction
1 Application profile
2 Identify COSMIC-FFP software functional user requirement boundary
2 1 User
2 2 Boundary
3 Identify candidate COSMIC-FFP triggering events functional processes and data groups
3 1 Identify candidate triggering events
3 2 Identify candidate functional processes
3 3 Identify candidate data groups
4 Map identified candidate items into COSMIC-FFP software mode
5 Identify COSMIC-FFP functional sub-processes and apply COSMIC-FFP measurement function
6 COSMIC-FFP measurement summary
7 Questions and answers
Appendix specification of rice cooker application

Figure 13 Gabarit initial d'étalon

- L'introduction donne un bref aperçu sur l'étude de cas et sur les étapes de la mesure.
- Le profil de l'application définit le type de logiciel à mesurer et son domaine d'application.
- L'identification de la frontière COSMIC-FFP des fonctionnalités utilisateurs requises du logiciel à mesurer décrit la frontière et les utilisateurs du logiciel.
- L'identification des événements déclencheurs, les processus fonctionnels et les groupes de données COSMIC-FFP candidats
- Arrimage des items candidats identifiés dans le modèle logiciel de COSMIC-FFP pour associer chaque processus fonctionnel à un/des évènement(s) déclencheur(s) et pour évaluer si les items sont des items COSMIC-FFP
- L'identification des sous-processus fonctionnels et l'application de la fonction de mesure COSMIC-FFP permettent de déterminer les mouvements de données et d'accorder 1 Cfsu (unité de COSMIC-FFP) à chaque mouvement de données.
- Le sommaire de la mesure COSMIC-FFP résume la mesure en tableaux récapitulatifs.
- Les questions et les réponses présentent les questions et les réponses suggérées par le mesureur pendant le processus de mesure.
- L'annexe décrit les spécifications fonctionnelles du logiciel à mesurer.

Plus tard, le GÉLOG a amélioré ce gabarit (GELOG, 2004) de rapport de mesure.

Il est maintenant composé des sections suivantes (figure 14) :

1. Overview
1.1 Introduction
1.2 Measurement viewpoint, purpose and scope
2. Requirements – as documented in the software specification document
3. COSMIC-FFP measurement procedure
3.1 Identification of layers
3.2 Identification of users
3.3 Boundary
3.4 Identification of triggering events
3.5 Identification of data groups
3.6 Identification of functional processes
4. Identify data movements
4.1 Use case diagram
4.2 Sequence diagrams
4.3 List of data movements
4.4 Observations on the clarity of the documented requirements
5. Total functional size
6. Conclusions and observations
7. Questions & answers

Figure 14 Gabarit final d'étalon

Les différences principales par rapport au gabarit initial sont la structure des sections et l'ajout des diagrammes UML à savoir le cas d'utilisation et les diagrammes de séquence. Ce gabarit permet de mieux comprendre les fonctionnalités d'un logiciel par le développement des diagrammes UML et d'identifier plus aisément chaque partie de la mesure.

8.3.5 La sélection des experts

La conception des étalons est une activité qui doit être menée au niveau international par des groupes d'experts de plusieurs pays afin d'obtenir un consensus en la matière. L'organisation de normalisation ISO représente le cadre le plus adéquat à ce type d'activités.

La sélection des experts est faite à partir des contacts du laboratoire GÉLOG. Le laboratoire GÉLOG a des relations avec le groupe COSMIC (*Common Software Measurement International Consortium*), un groupe volontaire et informel d'experts

internationaux en mesure des logiciels. L'étudiant doctorant mandataire est un membre d'ISO qui représente le Canada dans le groupe de travail WG 12. Ce groupe WG12 est spécialisé en MTF au niveau mondial. Ceci a permis de tisser plusieurs relations avec des experts internationaux en MTF des logiciels et faciliter la présentation du projet dans la communauté ISO du génie logiciel. En outre, le laboratoire du génie logiciel de l'ÉTS possède des experts en mesure des logiciels qui ont montré de l'intérêt à ce travail. À travers des travaux antérieurs de recherche, le GÉLOG a aussi développé un réseau de contact avec des experts internationaux en mesure des logiciels.

8.4 Le développement d'un site web

Il est souhaitable de rendre disponible le référentiel sur Internet pour des expérimentations plus poussées auprès des mesureurs et des experts. À cet effet, un site web sera développé pour faciliter l'accès aux étalons de mesure pour les experts et pour les personnes intéressés par le développement du référentiel.

Le tableau 20 résume l'application du cadre de Basili dans le contexte de cette recherche :

Tableau XX

Cadre de Basili modifié pour la recherche exploratoire

Définition			
Motivation	Objet	Objectif	Perspective
Fournir une première version des références pour les mesures de la taille fonctionnelle des logiciels	La méthode de mesure COSMIC-FFP, la norme ISO 14143 et deux études de cas de RUP	Développer un référentiel pour les mesures de la taille fonctionnelle des logiciels	Mesureurs des logiciels Chercheurs en génie logiciel Les consultants en programmes de mesures des logiciels

Tableau XX (Suite)

Cadre de Basili modifié pour la recherche exploratoire

Définition			
Motivation	Objet	Objectif	Perspective
			Les organisations de production des étalons et des instruments de mesure des logiciels
Planification			
Étapes du projet	Intrants au projet	Livrables du projet	
Étude des mesures des logiciels	Revue et synthèse de la littérature théorique Pratiques utilisées pour mesurer les logiciels	Rapport sur les différentes mesures des logiciels et leurs propriétés	
Détermination de la méthode de mesure pour développer les étalons	Méthodes disponibles Exigences de mesure des logiciels	Caractéristiques et attributs des mesures des logiciels	
Application de COSMIC-FFP sur les FUR Développement des diagrammes UML Développement du site web	L'ensemble des FUR d'ISO, de <i>Rice cooker</i> application et des études de cas de RUP Revue de littérature sur l'application de la méthode COSMIC-FFP Gabarits du rapport de mesure Documents sur la méthodologie UML	Présentation du référentiel sur Internet	
Opération			
Documents analysés	Commentaires des experts	Ensemble proposé	
Manuel de la méthode COSMIC-FFP Normes ISO sur les mesures des logiciels Les FUR des documents de formation de processus RUP, ISO 14143 et de l'application Rice Cooker	Plusieurs résultats de mesure et commentaires sont fournis par les experts internationaux, professeurs spécialistes en mesures des logiciels et des étudiants des cycles supérieurs en génie logiciel de l'ÉTS.	Ensemble constitué de huit étalons de mesures. Réalisés par l'application de la méthode COSMIC-FFP sur huit FUR différentes.	

Tableau XX (Suite)

Cadre de Basili modifié pour la recherche exploratoire

Opération		
Documents analysés	Commentaires des experts	Ensemble proposé
Documents de la méthodologie UML	Leurs interventions sont enregistrées dans les rapports des étalons.	
Interprétation		
Contexte d'interprétation	Extrapolation des résultats	Autre Travail
Référentiel qui sera considéré comme une possibilité de développer un référentiel pour les mesures de la taille fonctionnelle des logiciels	Pour d'autres mesures des logiciels	A ajouter d'autres références et à expérimenter l'ensemble des étalons, le premier élément qui sera expérimenté est l'utilisation des étalons en industrie

8.5 Méthode de liaison d'une mesure au référentiel

En métrologie, un référentiel doit permettre de lier toute mesure à une ou plusieurs références. Le tableau 21 ci-dessous montre les quatre étapes de la méthode de liaison adaptée de Baize *et al.* (1995):

Tableau XXI

Méthode de liaison au référentiel, adapté de Baize et al. (1995)

étape	Description
Identification de la mesure	Descriptions et analyses des mesures existantes pour ce type de logiciel; description minimale de l'environnement du logiciel étudié.
Interprétation	Pour effectuer la méthode d'interprétation de la mesure, le mesureur puise dans le corpus des connaissances de son milieu professionnel et dans son expérience personnelle. L'interprétation d'une mesure ne peut être faite indépendamment du référentiel, ni de nombreux éléments du processus de mesure.

Tableau XXI (Suite)

Méthode de liaison au référentiel, adapté de Baize et al. (1995)

étape	Description
Liaison	<p>La liaison consiste à relier une mesure à une ou plusieurs références puis à lui donner le nom correspondant. Cela se fait par un raisonnement technique qui est du même ordre que celui effectué lors de l'interprétation des mesures.</p> <p>La liaison est un système souple qui nécessite l'étude de la ressemblance entre une mesure et les références. Pour analyser cette ressemblance on peut se fonder sur les concepts statistiques et employer des méthodes telles qu'analyses multidimensionnelles. On distingue des liaisons simples et multiples</p>
Ajout d'une nouvelle Référence	<p>Si une mesure se trouve très éloignée de toutes les références définies antérieurement, ce peut être l'occasion d'ajouter au référentiel une nouvelle référence. Cela est en effet toujours possible sans pour autant remettre en cause la cohérence du référentiel. De même il est possible de signaler l'existence de nouveaux types. Cependant, afin d'éviter la confusion, toute proposition de création d'une nouvelle référence ou d'un nouveau qualificatif devra faire l'objet d'une étude préalable détaillée et argumentée afin de maintenir la cohérence générale.</p>

8.6 Développement du référentiel

Le référentiel proposé sera composé de huit étalons de mesure appelés aussi références. Ces étalons seront réalisés à partir de l'application de la méthode MTF des logiciels COSMIC-FFP sur huit ensembles différents des fonctionnalités utilisateurs requises FUR des logiciels. Cinq des FUR proviennent de la norme ISO 14143-4, document ISO dont le but est justement de fournir aux chercheurs et aux praticiens des ensembles de fonctionnalités utilisateur requises (FUR) publiquement disponibles pour mesurer la taille fonctionnelle des logiciels :

1. Automatic Line Switching System;
2. Hotel Reservation System;
3. L-Euchre Application;
4. SAGA System;

5. Valve Control System;
6. Un autre ensemble de FUR appartient à l'application *Rice Cooker* (LRGL, 2004);
7. Course Registration System;
8. Collegiate Sports Paging System.

Les deux dernières FUR appartiennent aux deux logiciels fournis comme matériel de formation par le processus *Rational Unified Process*. Le *Rational Unified Process (RUP)* est un processus développé par la société IBM-Rational afin de bien maîtriser l'assignation des tâches et la responsabilisation des différents acteurs participant tout au long du cycle de développement du logiciel.

Le choix de ces ensembles de FUR de logiciels a été fait selon deux critères:

Disponibilité de la documentation : en tant que membres canadiens d'ISO/IEC SC7, l'équipe de recherche a accès aux exigences logicielles qui appartiennent au document ISO 14143-4. Pour les exigences logicielles de RUP sont disponibles sur Internet à des fins éducatives;

Crédibilité des organisations propriétaires des exigences logicielles : l'organisation internationale des standards (ISO). La compagnie IBM-Rational sont mondialement connues et contribuent dans la maturité du génie logiciel.

Il a été prévu que le projet se déroule en deux phases : préparation des mesures et vérification par des experts des résultats des mesures.

La première phase est l'obtention des résultats de mesure suite à l'application de la méthode COSMIC-FFP sur les huit ensembles des FUR des logiciels cités plus haut. La méthode de mesure COSMIC-FFP (ISO 19761), une méthode de MTF des logiciels, consiste à appliquer un ensemble de règles et de procédures sur les fonctionnalités livrées aux utilisateurs et qui sont décrites via les Fonctionnalités Utilisateurs Requises

(FUR) afin d'aboutir à un nombre, son unité est le Cfsu (*COSMIC functional size unit*), représentant une valeur quantitative de la taille fonctionnelle du logiciel.

La deuxième phase est la vérification des étalons de mesures par consensus entre experts en MTF. Le nombre d'experts qui ont participé dans le développement du consensus, varie d'un ensemble de FUR à un autre.

La figure 15 suivante présente les livrables prévus pour la première version du référentiel :

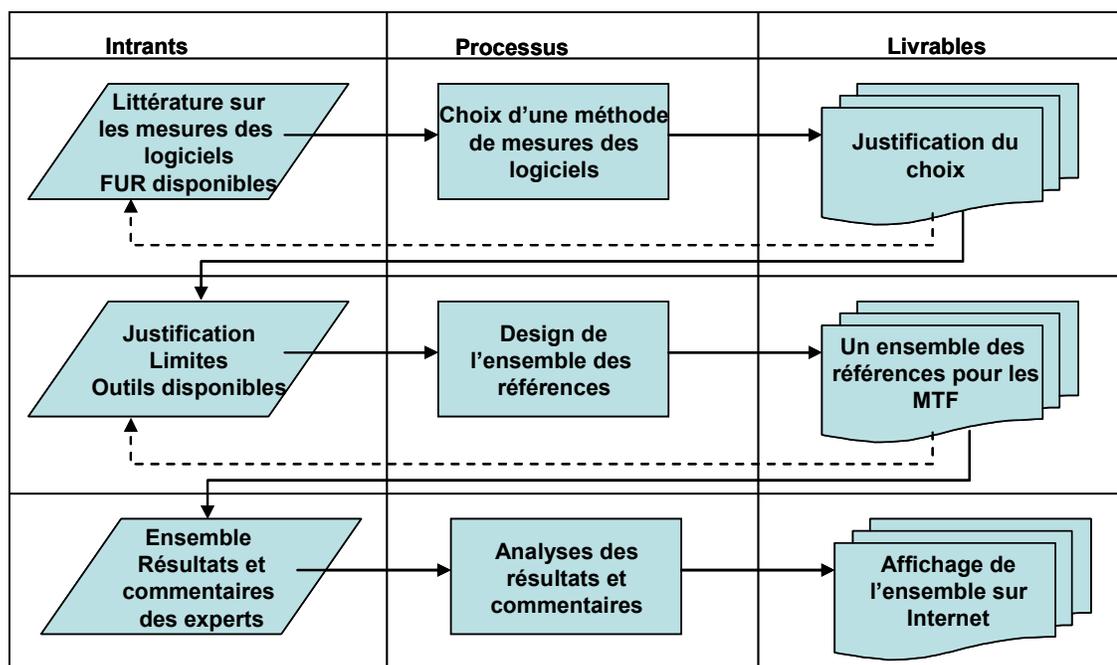


Figure 15 Processus, intrants et livrables

Cependant, la figure 16 suivante décrit les étapes de la méthodologie :

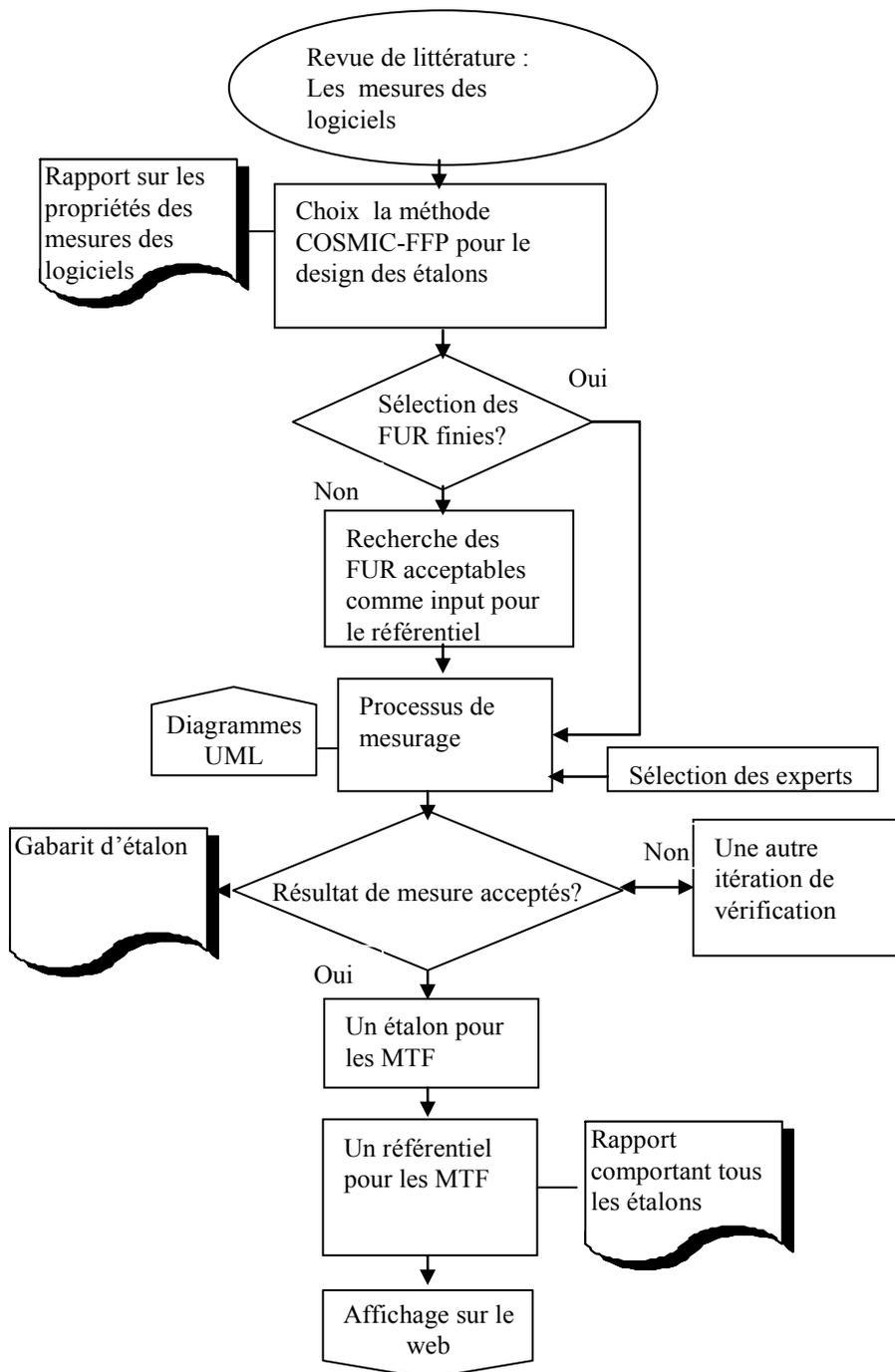


Figure 16 Les étapes principales de la méthodologie

8.7 Les partenaires de la recherche

C'est pour relever le double défi à la fois scientifique et industriel du développement du référentiel que plusieurs partenaires publics et privés seront contactés pour assurer la propagation de cette recherche dans la communauté des experts, aux organismes et aux étudiants des cycles supérieurs intéressés par les mesures de la taille fonctionnelle des logiciels. Les partenaires nationaux sont des employés de Bell Canada à Montréal, des professeurs du Département du génie logiciel et des TI de l'ÉTS et des étudiants du cycle supérieur du même département. Les partenaires internationaux sont des experts internationaux en mesure des logiciels et des professeurs universitaires du Royaume-Uni, d'Espagne, de Belgique, d'Irlande et d'Australie. Plusieurs de ces experts sont du groupe COSMIC (*Common Software Measurement International Consortium*) et du groupe de travail WG12 d'ISO.

8.8 Sommaire

La recherche exploratoire traite dans ce travail du développement d'un référentiel pour les mesures de la taille fonctionnelle des logiciels. Ce chapitre décrit la méthodologie de recherche pour réaliser cet objectif.

Ce chapitre présente la motivation de l'équipe de recherche pour le sujet abordé, les utilisateurs du référentiel des MTF et un rappel de l'objectif de la recherche, les outils nécessaires à sa réalisation et ses limites. Une bonne partie du contenu de ce chapitre a décrit les principales étapes de la méthodologie, à savoir : la détermination de la méthode de mesure pour concevoir le référentiel, la réalisation d'étalon, l'ajout des diagrammes UML, la conception du gabarit d'étalon et la sélection des experts pour vérifier la qualité des étalons.

Puis, le chapitre a présenté le développement du site web pour afficher le référentiel et la méthode de liaison d'une mesure au référentiel, et les partenaires dans cette thèse.

Cette recherche de type exploratoire peut donc être considérée comme un premier pas sur un terrain d'une grande complexité et diversité de génie logiciel qui est la conception des étalons de mesures. Du fait que la conception des étalons dans les génies traditionnels tels que le génie civil, mécanique ou électrique, est un travail de grande précision qui requerra par la suite diverses itérations afin de continuer à améliorer les étalons proposés. Nous savions par ailleurs, qu'il n'existait pas, à notre connaissance, de travaux antérieurs sur la construction d'étalons pour la mesure des logiciels. La démarche méthodologique suivie dans cette recherche peut donc être utile pour produire un référentiel des mesures de logiciels.

Le prochain chapitre expose le déroulement de la recherche. Il explique l'exécution de la méthodologie pour développer le référentiel des mesures.

CHAPITRE 9

DÉROULEMENT DE LA RECHERCHE

Ce chapitre décrit le déroulement des travaux de recherche pour réaliser ce travail. Une description des étapes d'exécution des tâches est fournie pour résumer les principales activités faites dans cette thèse.

9.1 Introduction

Afin de réaliser ce travail, un plan a été développé au début de l'activité de recherche. Le plan a subi plusieurs changements et plusieurs ajustements pour répondre aux attentes et aux contraintes des partenaires de la recherche et pour tenir compte des ressources disponibles pour réaliser la thèse. Les sections suivantes décrivent sommairement ces différentes phases.

9.2 Sélection des FUR

Pour développer le référentiel, un ensemble de FUR a été sélectionné. Comme indiqué auparavant, l'ensemble de FUR est constitué de huit fonctionnalités utilisateurs requises.

Cinq des FUR proviennent de la norme ISO 14143-4 :

1. Automatic Line Switching System;
2. Hotel Reservation System;
3. L-Euchre Application;
4. SAGA System;
5. Valve Control System;
6. Un autre ensemble de FUR appartient à l'application Rice Cooker (LRGL, 2004);
7. Course Registration System;
8. Collegiate Sports Paging System.

Les deux dernières FUR appartiennent aux deux logiciels fournis comme matériel de formation par le processus Rational Unified Process.

9.3 Le processus de mesurage

Après la sélection des FUR, le mesureur en fait l'analyse. Ici, le mesureur est l'étudiant mandataire. Il ajoute les diagrammes des cas d'utilisation et de séquences d'UML pour une meilleure compréhension des fonctionnalités du logiciel à mesurer. Dans le cas où ces diagrammes existent déjà dans la documentation du logiciel à mesurer, le mesureur les étudie et les ajoute dans le rapport de mesure. Puis, il fait l'opération de la mesure. L'opération consiste en l'application de la méthode COSMIC-FFP sur les FUR choisies.

Les paragraphes suivants, tirés du Guide COSMIC d'application de la norme ISO 19761 (COSMIC-FFP) (A. Abran et al., 2003), décrivent la méthode. COSMIC-FFP offre une méthode normalisée pour mesurer la taille fonctionnelle du logiciel des domaines fonctionnels généralement désignés sous le nom de logiciel « d'application de gestion » (ou MIS) et logiciel « temps réel ». La méthode de mesure COSMIC-FFP consiste à appliquer un ensemble de règles et de procédures sur un logiciel donné, tel qu'il est perçu par ses utilisateurs, c'est-à-dire à travers les besoins fonctionnels. Le résultat de l'application de ces règles et de ces procédures est un nombre représentant la taille fonctionnelle du logiciel, c'est-à-dire une valeur quantitative. La figure 17 représente ce processus.

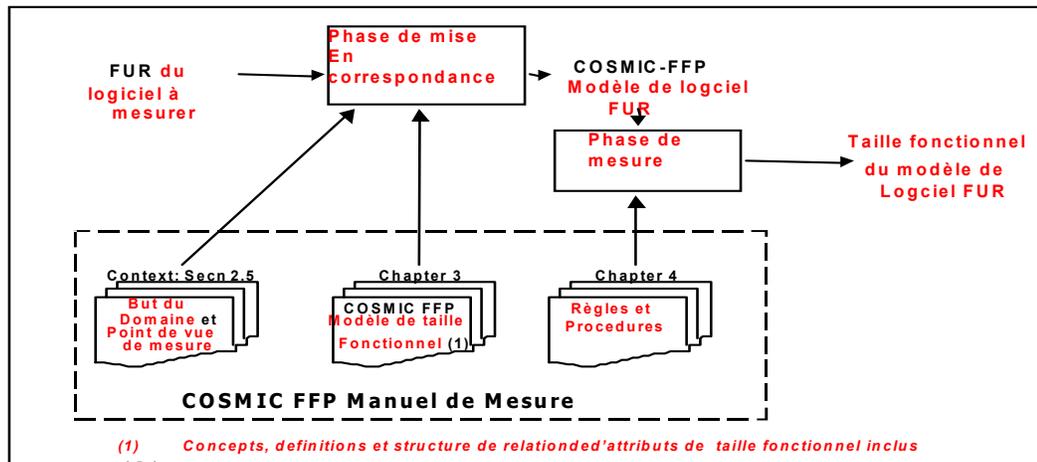


Figure 17 Modèle de mesure COSMIC-FFP (Abran et al., 2003)

La méthode considère la mesure de la taille fonctionnelle du logiciel à travers deux phases distinctes : la mise en correspondance du logiciel à mesurer avec le modèle de logiciel de COSMIC-FFP et la mesure des aspects spécifiques de ce modèle de logiciel.

La méthode générale du processus de mise en correspondance du modèle de logiciel générique COSMIC-FFP est résumée à la figure 18.

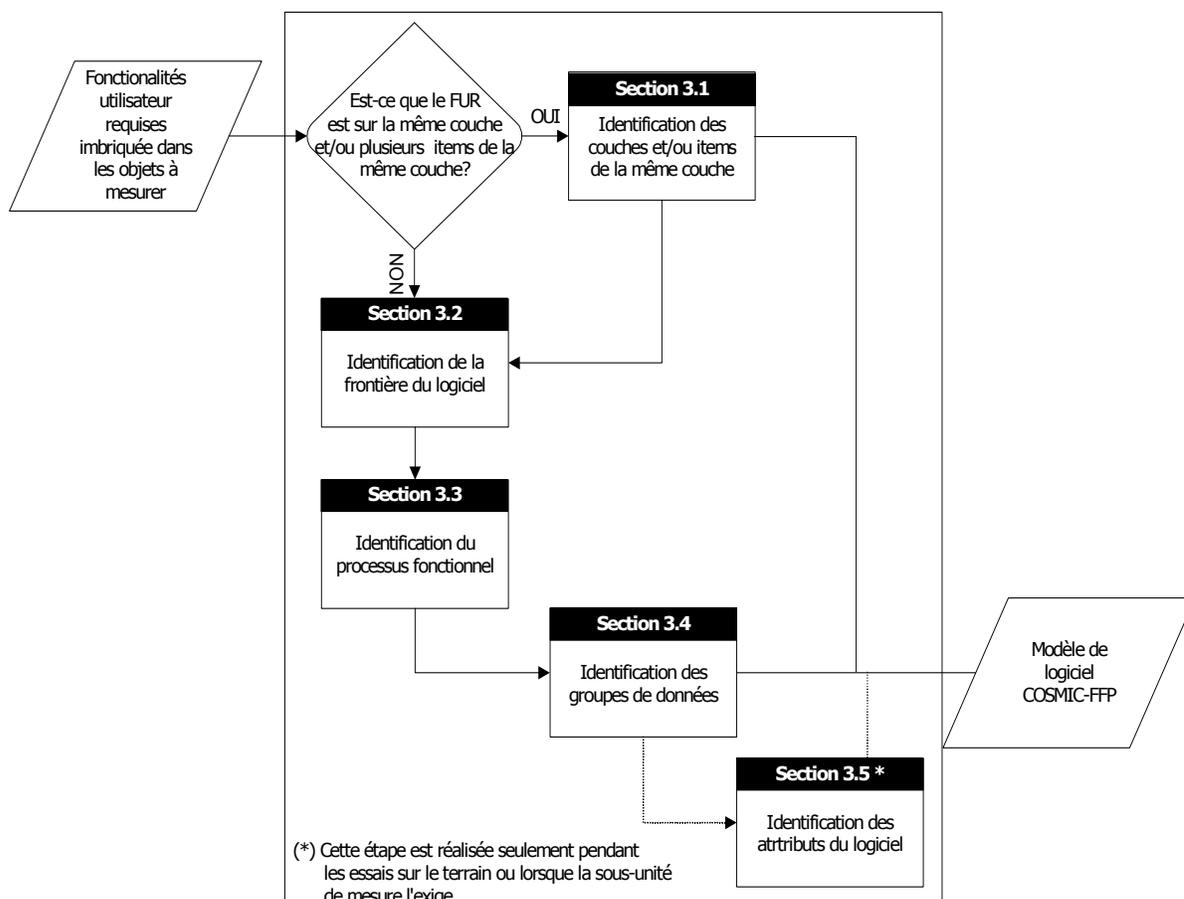


Figure 18 Méthode générale de mise en correspondance COSMIC-FFP

La dérivation de la taille fonctionnelle du logiciel mesuré est indépendante de l'effort requis pour développer ou maintenir ce logiciel. Elle est aussi indépendante de la méthode utilisée pour développer ou pour maintenir le logiciel ou encore indépendante de tout composant technologique ou physique de ce logiciel.

La procédure générale, pour mesurer une pièce de logiciel à partir du modèle générique de logiciel COSMIC-FFP, est résumée à la figure 19.

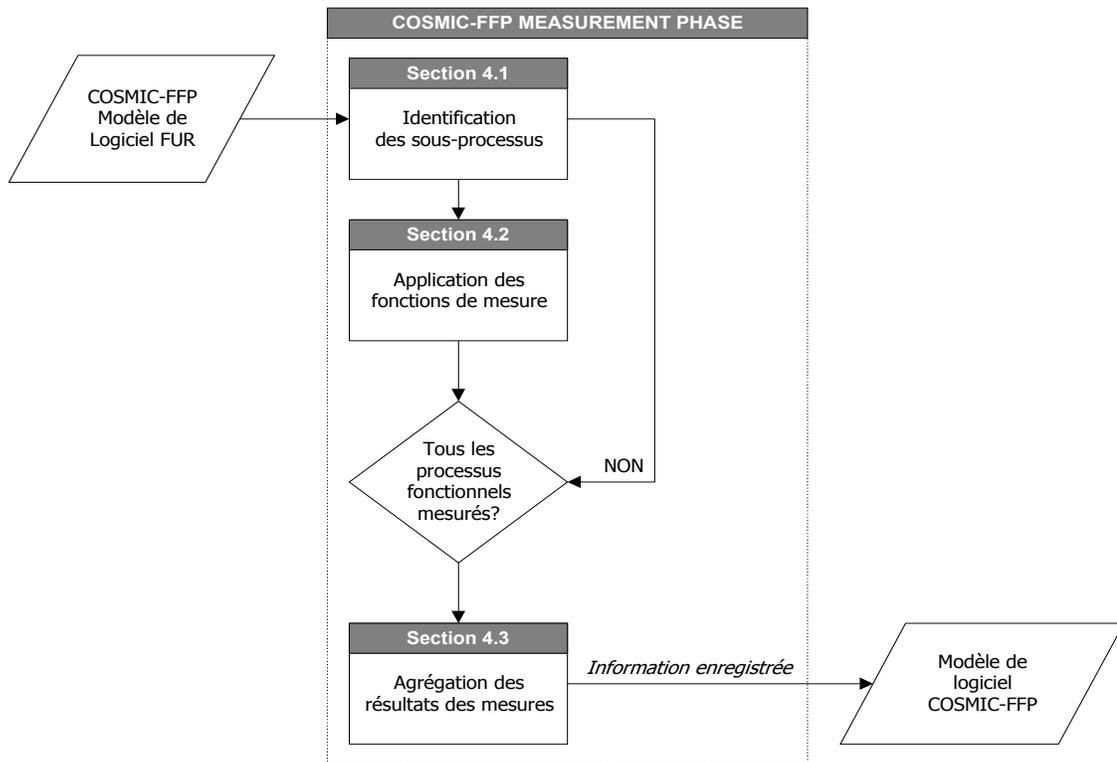


Figure 19 Procédure générale de la phase de mesure COSMIC-FFP

En parallèle, des étudiants en cycle supérieur, inscrits au cours gradué SYS861 La mesure : concept clef en ingénierie du logiciel (2003 et 2004), ont fait la mesure des mêmes FUR en 2003 et en 2004.

Le mesureur analyse ensuite les résultats des étudiants gradués pour situer ses interprétations des FUR par rapport à celles des étudiants, afin d'éliminer les éventuels ambiguïtés qui peuvent exister dans le résultat de mesure.

9.4 Gabarit d'étalon utilisé

Comme vu au chapitre précédent de méthodologie, l'élaboration finale du gabarit d'étalon est faite à partir des gabarits réalisés par le groupe COSMIC et le Laboratoire de recherche en génie logiciel (GÉLOG) de l'École de technologie supérieure. Le premier format du gabarit, développé en 2000, a été celui des *case studies* préparés par le groupe COSMIC et mis dans le domaine public sur le site web géré par le GÉLOG.

Le deuxième format a été ensuite développé en 2004 par le GÉLOG après l'adoption de la norme ISO 19761 (COSMIC-FFP) et dans le contexte de la préparation de matériel de cours pour la formation en Indes sur la norme ISO 19761.

Le tableau 22 suivant présente une comparaison entre les deux gabarits :

Tableau XXII

La comparaison entre les deux gabarits

No de section	Gabarit 1	Gabarit 2
1	L'introduction donne un bref aperçu sur l'étude de cas et les étapes de la mesure. Le profil de l'application définit le type de logiciel à mesurer et son domaine d'application.	L'aperçu a deux sections : L'introduction qui donne un bref aperçu sur le logiciel et définit le type de logiciel à mesurer ainsi que son domaine d'application; Le point de vue de la mesure, son but et sa portée qui détermine quel point de vue est utilisé pour faire la mesure, résume le but de la mesure et délimite sa portée.
2	L'identification de la frontière COSMIC-FFP des fonctionnalités utilisateurs requises du logiciel à mesurer décrit la frontière et les utilisateurs du logiciel.	Les exigences – comme documentées dans le document SRS (<i>Software Requirements Specifications</i>) : cette section présente les exigences et les spécifications fonctionnelles du logiciel à mesurer comme décrites par les documents SRS disponibles. Ces sont les documents utilisés pour faire la mesure.

Tableau XXII (Suite)

La comparaison entre les deux gabarits

No de section	Gabarit 1	Gabarit 2
	L'identification des événements déclencheurs, processus fonctionnels et groupes de données COSMIC-FFP candidats la méthode COSMIC-FFP.	L'équivalent de cette section pour le gabarit se trouve dans son annexe.
3	Identification des événements déclencheurs, processus fonctionnels et groupes de données candidats	<p>La procédure de mesure COSMIC-FFP a six sections :</p> <p>L'identification des couches permet de définir les couches du logiciel à mesurer ;</p> <p>L'identification des utilisateurs donne la liste des utilisateurs du logiciel à mesurer ;</p> <p>L'identification de la frontière COSMIC-FFP des fonctionnalités utilisateurs requises du logiciel à mesurer décrit la frontière et les utilisateurs du logiciel;</p> <p>L'identification des événements déclencheurs identifie les événements déclencheurs candidats, selon la méthode COSMIC-FFP, qui ont amorcé les processus fonctionnels du logiciel en mesure;</p> <p>L'identification des groupes de données définit les groupes de données candidats, selon la méthode COSMIC-FFP, qui interagissent pour produire les fonctionnalités du logiciel en question;</p> <p>L'identification des processus fonctionnels décrit les processus fonctionnels candidats, selon la méthode COSMIC-FFP, du logiciel en mesure. Cette section permet aussi d'évaluer les événements déclencheurs, les groupes de données et les processus fonctionnels candidats si sont des items de COSMIC-FFP.</p>

Tableau XXII (Suite)

La comparaison entre les deux gabarits

No de section	Gabarit 1	Gabarit 2
4	Arrimage des items candidats identifiés dans le modèle logiciel de COSMIC-FFP pour associer chaque processus fonctionnel à un/des événement(s) déclencheur(s) et d'évaluer si les items sont des items COSMIC-FFP.	L'identification des mouvements de données comprend l'application de la fonction de mesure COSMIC-FFP permet de déterminer les mouvements de données et d'assigner 1 Cfsu (unité de COSMIC-FFP) à chaque mouvement de données.
5	L'identification des sous-processus fonctionnels et application de la fonction de mesure COSMIC-FFP permet de déterminer les mouvements de données et d'accorder 1 Cfsu (unité de COSMIC-FFP) à chaque mouvement de données.	L'analyse des résultats de mesure fournit des tableaux récapitulatifs des résultats de mesure. Ces tableaux montrent la distribution des mouvements de données (E, R, W, X) par processus fonctionnel, la contribution de chaque processus fonctionnel à la taille totale et le pourcentage des mouvements de données par rapport à la taille totale du logiciel. Cette section montre aussi un graphique sur la proportion de chaque mouvement de données dans la taille totale du logiciel.
6	Le sommaire de la mesure COSMIC-FFP résume la mesure en tableaux récapitulatifs.	Les observations décrivent l'évaluation du mesureur de la qualité de la documentation fournie pour faire la mesure.
7	Les questions et les réponses présentent les questions et les réponses suggérées par le mesureur pendant le processus de mesure.	Les questions et les réponses présentent les questions et les réponses suggérées par le mesureur pendant le processus de mesure.

De plus, cette thèse ajoute au gabarit 2 les sections suivantes :

- Le diagramme UML - cas d'utilisation permet de traduire les fonctions du système en utilisant le même formalisme de documentation;
- Le diagramme de séquence UML permet de spécifier les interactions qui existent entre les objets du système.

La raison principale de choisir cette structure d'étalon est d'uniformiser le formalisme des modèles de rapport d'étalons et de faire l'arrimage avec les règles de la méthode de mesure COSMIC-FFP.

Il est essentiel de noter que c'est la première fois que cette structure est exposée dans ce genre de travail.

9.5 L'intervention des experts

L'intervention des experts se fait de trois façons : Il y a des experts qui sont impliqués dès le début du processus de mesurage avec l'équipe de mesureurs du GÉLOG; d'autres fournissent leurs résultats de mesure; enfin, d'autres font la vérification de la mesure.

En effet, quand ce projet a été présenté aux experts (17), ils ont tous affirmé leur intérêt et leur implication dans le projet (sauf un seul cas où l'expert, bien qu'ayant indiqué un grand intérêt pour cette recherche, ne pouvait pas participer faute de disponibilité de temps).

Actuellement la participation des experts se continue, cependant la période de leur participation dépasse la période du doctorat de l'étudiant mandataire. Des travaux ultérieurs à la période de doctorat sont envisagés pour continuer à améliorer le référentiel. Une hypothèse à considérer dans ce travail de recherche est que l'expertise des experts internationaux est correcte et exhaustive : des discussions poussées sur les résultats de mesure avec les experts à travers des forums de discussion sont en cours afin de clarifier toute interprétation qui pourrait prêter à des résultats de mesure non concordant d'un mesureur à l'autre et d'essayer d'établir des résultats de mesure avec une valeur conventionnellement vraie : « Étant donné qu'une valeur vraie ne peut pas

être déterminée, dans la pratique on utilise une valeur conventionnellement vraie » (ISO, 1993).

Selon l'organisation internationale de métrologie légale (OIML, 2002) la valeur conventionnellement vraie d'une grandeur est la valeur attribuée à une grandeur particulière et reconnue, parfois par convention, comme la représentant avec une incertitude appropriée pour un usage donné. Par exemple, en un lieu donné, la valeur attribuée à la grandeur réalisée par un étalon de référence peut être prise comme étant une valeur conventionnellement vraie. À noter que la valeur conventionnellement vraie est quelquefois appelée valeur assignée, meilleure estimation de la valeur, de la valeur convenue ou de la valeur de référence. « On utilise souvent un grand nombre de résultats de mesures d'une grandeur pour établir une valeur conventionnellement vraie » (OIML, 2002).

Ce travail de préparation des résultats de mesure (i.e. le mesurage) est fait de façon itérative. Voici des exemples de questions à se poser pour chaque itération :

- Est-ce que le résultat de la mesure obtenu correspond à celui de l'itération précédente?
- Est-ce qu'un autre expert pourrait vérifier ce résultat et être d'accord?
- Est-ce que je puis me servir de la démarche pour parfaire mes résultats dans les prochaines itérations?

9.6 Le développement d'un site web

Une fois que le processus de mesurage est réalisé par les différents partenaires dans cette recherche, un rassemblement des documents et un arrimage entre les opinions de l'équipe du GÉLOG avec ceux des experts sur toutes les parties de l'étalon sont faits pour développer un consensus sur le contenu et l'exactitude de l'étalon.

En outre, un forum de discussion a aussi été développé pour les personnes intéressées par ce projet afin de leur permettre d'enregistrer leurs commentaires et d'échanger des avis. Le forum se trouve à l'adresse suivante :

www.gelog.etsmtl.ca/FSMetalons/disc2_tocf.htm

Un exemple d'interaction par le forum est présenté à la figure 21.

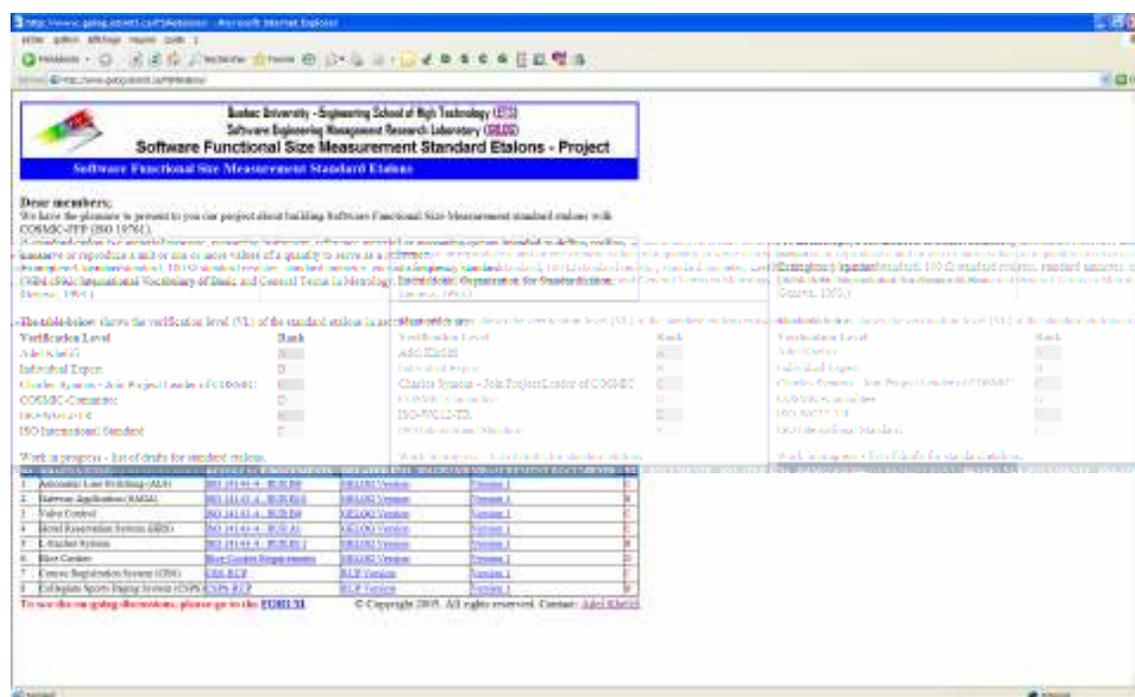


Figure 21 Forum de discussion sur les étalons de mesure

La mise en place de cette infrastructure légère a favorisé une bonne collaboration entre les différentes parties prenantes de ce projet et une bonne vérification de la qualité de la première version des étalons proposés dans cette thèse. Les partenaires principaux impliqués dans ce projet sont : le Laboratoire de recherche en génie logiciel (GÉLOG) de l'ÉTS, des gestionnaires de produits de Rational Unified Process, des étudiants des cycles supérieurs en génie logiciel, le co-directeur du groupe COSMIC – M. Charles

Symons, des membres du groupe COSMIC et des membres du groupe de travail ISO/IEC JTC1/SC7 WG12.

9.7 Personnes impliquées dans la construction des étalons

Les tableaux 23 à 30 suivants, tirés des rapports des étalons aux annexes, présentent les personnes (voir Annexe 8) impliquées dans la construction de chaque étalon.

Tableau XXIII

Étalon 1 - Automatic Line Switching System

Date	Réviseurs	Modifications / Additions
2004-08-26	Étudiant mandataire	Première version
2004-08-29	1 ^{er} expert international	Révision
2004-09-01	Étudiant mandataire	Révision
2004-09-11	2 ^e expert international	Révision
2005-03-11	2 ^e expert international	Révision
2005-06-01	3 ^e expert international	Révision

Tableau XXIV

Étalon 2 - SAGA System

Date	Réviseurs	Modifications / Additions
2004-03-12	Groupe d'étudiants de l'ÉTS de cycles supérieurs	Résultat de la mesure
2005-02-06	Étudiant mandataire	Premier Draft
2005-05-18	Expert international	Révision

Tableau XXV

Étalon 3 - Valve Control System

Date	Réviseurs	Modifications / Additions
2003-01-20	1 ^{er} expert international	Premier <i>draft</i>

Tableau XXV (Suite)

Étalon 3 - Valve Control System

Date	Réviseurs	Modifications / Additions
	2 ^e expert international	
2004-08-16	Étudiant mandataire	Révision du document, ajout des diagrammes UML
2004-08-16	1 ^{er} expert international	Révision et ajout des Références Utilisateurs Requisites ISO
2004-08-21	Étudiant mandataire	Révision des figures
2004-08-23	Secrétaire	Mise à jour du format du rapport de la mesure
2004-08-25	3 ^e expert international	Révision complète des sections 1 à 3.7
2004-09-01	Étudiant mandataire	Révision complète à partir de la section 3.8
2004-09-06	3 ^e expert international	Commentaire sur les révisions ultérieures
2004-09-07	Étudiant mandataire	Application des commentaires
2004-09-17	2 ^e expert international	Commentaires sur les révisions ultérieures
2004-09-19	Étudiant mandataire	Application des commentaires
2005-02-02	3 ^e expert international	Propositions des améliorations

Tableau XXVI

Étalon 4 - Hotel Reservation System

Date	Réviseurs	Modifications / Additions
2004-03-12	Groupe d'étudiants de l'ÉTS de cycles supérieurs	Résultat de la mesure
2005-02-02	Étudiant mandataire	Premier <i>draft</i>
2005-03-14	1 ^{er} expert international 2 ^e expert international 3 ^e expert international	Révision

Tableau XXVII

Étalon 5 - L-Euchre Application

Date	Réviseurs	Modifications / Additions
2004-03-12	Groupe d'étudiants de l'ÉTS de cycles supérieurs	Résultat de la mesure

Tableau XXVII (Suite)

Étalon 5 - L-Euchre Application

2005-02-02	Étudiant mandataire	Premier <i>draft</i>
2005-03-14	1 ^{er} expert international 2 ^e expert international 3 ^e expert international	Révision

Tableau XXVIII

Étalon 6 - Rice Cooker Application

Date	Réviseurs	Modifications / Additions
99-12-06	1 ^{er} expert international 2 ^e expert international	Premier <i>draft</i>
2000-01-11	1 ^{er} expert international 2 ^e expert international 3 ^e expert international 4 ^e expert international 5 ^e expert international 6 ^e expert international	Plusieurs changements et mises à jour des résultats de la mesure
2000-08-09	3 ^e expert international	Deux questions et réponses ajoutées sur la page p17. Correction mineure de l'exigence 3.
2002-03-13	3 ^e expert international	<i>Updates to include the Purpose and Scope step, and to Questions and Answers</i> <i>- Addition of Annex B</i>
2003-01-02	1 ^{er} expert international, 7 ^e expert international, 8 ^e expert international,	Mise à jour pour: - synchroniser avec le vocabulaire avec ISO 19761:2003; - clarifier quelles exigences fonctionnelles sont allouées au <i>hardware</i> et au <i>software</i> ; - séparer les exigences en trois prototypes itératifs
2005-03-02	Étudiant mandataire	Ajout des diagrammes UML

Tableau XXIX

Étalon 7 - C-Registration System

Date	Réviseurs	Modifications / Additions
2004-08-26	Étudiant mandataire	Premier <i>draft</i>
2004-09-07	1 ^{er} expert international	Révision
2004-09-16	Étudiant mandataire	Deuxième <i>draft</i>
2005-05-14	2 ^e expert international	Révision

Tableau XXX

Étalon 8 - Collegiate Sports Paging System

Date	Réviseurs	Modifications / Additions
2004-09-16	Étudiant mandataire	Premier <i>draft</i>
2004-09-29	1 ^{er} expert international	Révision
2005-05-17	2 ^e expert international	Révision

9.8 Sommaire

Ce chapitre a décrit les séquences de déroulement de l'activité de recherche pour faire ce travail. Le chapitre débute en expliquant la sélection des fonctionnalités utilisateurs requises à mesurer et le processus de mesurage. Il a décrit la façon d'intervention des experts et le développement du site web pour afficher le référentiel. Le chapitre s'est terminé avec un aperçu sur des réviseurs impliqués dans la construction de chaque étalon.

Le chapitre suivant aborde la réalisation des étalons. Il présente les différentes sections qui composent un étalon proposé dans cette thèse. Il donne aussi un exemple des huit étalons conçus dans cette recherche qui est le Valve Control System Etalon.

CHAPITRE 10

DISCUSSION DES RÉSULTATS

Ce chapitre discute les différentes sections qui composent un étalon proposé dans cette thèse. Il donne aussi un exemple des huit étalons conçus dans cette recherche.

10.1 Introduction

Un étalon contribue à la standardisation des mesures. On définit un étalon et on « compare » l'étalon à ce que l'on veut mesurer afin de standardiser les pratiques du processus de mesurage. Une bonne comparaison doit toucher tout le contenu d'un étalon. Cette recherche propose un contenu d'étalon pour servir comme référence de mesure pour les MTF.

10.2 Le contenu d'un étalon

Pour la mesure de la taille fonctionnelle des logiciels, un étalon est un document contenant les sections suivantes (voir figure 22) :

1. Overview
1.1 Introduction
1.2 Measurement viewpoint, purpose and scope
2. Requirements - as documented in SRS document
3. COSMIC-FFP measurement procedure
3.1 Identification of layers
3.2 Identification of users
3.3 Boundary
3.4 Identification of triggering events
3.5 Identification of data groups
3.6 Identification of functional processes
4. Identify data movements
4.1 Use case diagram
4.2 Sequence diagrams
4.3 Observations on the clarity of the documented requirements
5. Analysis of measurement results
6. Observations
7. Questions and answers

Figure 22 Le contenu d'un étalon

1- L'Overview a deux sections :

- i. L'introduction donne un bref aperçu sur le logiciel et définit le type de logiciel à mesurer ainsi que son domaine d'application;
- ii. Le point de vue de la mesure, son but et sa portée déterminent quel point de vue est utilisé pour faire la mesure et résumant le but de l'exercice de mesurage, la mesure et délimite sa portée.

2- Les Requirements – comme documentées dans le document SRS (Software Requirements Specifications) : cette section présente les exigences et les spécifications fonctionnelles du logiciel à mesurer comme décrites par les documents disponibles. Elle présente les documents utilisés pour faire la mesure.

3- La procédure de mesure COSMIC-FFP a six sections :

- i. L'identification des couches permet de définir les couches du logiciel à mesurer.
- ii. L'identification des utilisateurs donne la liste des utilisateurs du logiciel à mesurer.
- iii. L'identification de la frontière COSMIC-FFP des fonctionnalités utilisateurs requises du logiciel à mesurer décrit la frontière et les utilisateurs du logiciel.
- iv. L'identification des événements déclencheurs identifie les événements déclencheurs candidats, selon la méthode COSMIC-FFP, qui ont amorcé les processus fonctionnels du logiciel en mesure.
- v. L'identification des groupes de données définit les groupes de données candidats, selon la méthode COSMIC-FFP, qui interagissent pour produire les fonctionnalités du logiciel en question.
- vi. L'identification des processus fonctionnels décrit les processus fonctionnels candidats, selon la méthode COSMIC-FFP, du logiciel en mesure. Cette section permet aussi d'évaluer les événements déclencheurs, les groupes de données et les processus fonctionnels candidats si ce sont des items de COSMIC-FFP.

4- L'identification des mouvements de données comprend trois sections :

- i. Le diagramme UML - cas d'utilisation qui permet de traduire les fonctions du système en utilisant le même formalisme de documentation;
- ii. Le diagramme de séquence UML permet de spécifier les interactions qui existent entre les objets du système. Les ressources UML utilisées dans cette thèse sont fournies par le consortium OMG (2005). L'outil utilisé pour développer les diagrammes UML est le Rational Rose (Rational Software Corporation, 2002);
- iii. Observations sur la clarté des exigences documentées du logiciel qui résument les commentaires du mesureur sur l'interprétation des exigences documentées du logiciel à mesurer.

Cette section se termine par l'application de la fonction de mesure COSMIC-FFP, elle permet de déterminer les mouvements de données et d'assigner 1 Cfsu (unité de COSMIC-FFP) à chaque mouvement de données.

5- L'analyse des résultats de mesure fournit des tableaux récapitulatifs des résultats de mesure. Ces tableaux montrent la distribution des mouvements de données (E, R, W, X) par processus fonctionnel, la contribution de chaque processus fonctionnel à la taille totale et le pourcentage des mouvements de données par rapport à la taille totale du

logiciel. Cette section montre aussi un graphique sur la proportion de chaque mouvement de données dans la taille totale du logiciel.

Cette façon de documenter la mesure est bien conforme aux sections 3, 4 et 5 du manuel d'utilisation de la norme ISO 19761 (2002) qui est la méthode COSMIC-FFP. La section 3 du manuel est la phase de mise en correspondance, elle permet d'identifier les couches du logiciel, sa frontière, les processus fonctionnels, les groupes de données, les attributs. La section 4 du manuel est la phase de mesure, elle permet d'identifier les sous-processus (E, X, R, W) et d'appliquer la fonction de mesure. La section 5 du manuel fait la présentation des résultats de la mesure. Toutes ces sections représentent le cœur du gabarit d'étalon conçu dans ce travail.

6- Les observations décrivent l'évaluation du mesureur de la qualité de la documentation disponible pour faire la mesure.

7- Les questions et les réponses présentent les questions et les réponses suggérées par le mesureur pendant le processus de mesure.

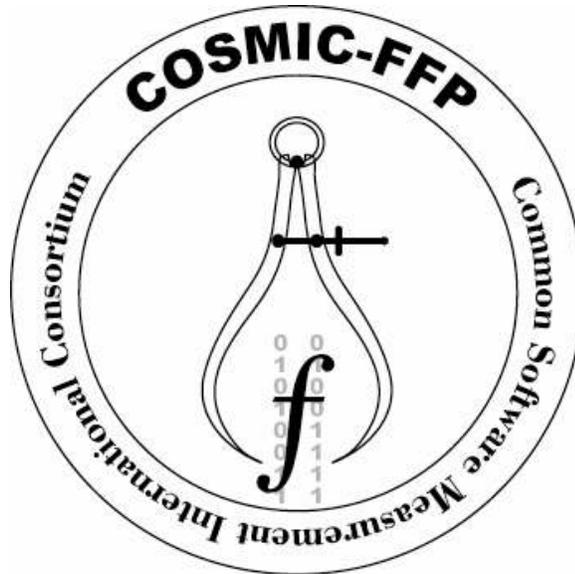
10.3 Le valve control system etalon

Les sections suivantes présentent le détail du Valve Control System Etalon. L'étalon est rédigé en anglais, car c'est la langue commune de tous les intervenants dans ce projet. Les critères de sélection de cet étalon comme exemple d'implantation du gabarit sont les suivants :

1. L'étalon est vérifié par plusieurs experts internationaux.
2. La documentation disponible est de bonne qualité.
3. L'étalon est le moins volumineux et le plus simple du référentiel.

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

UNIVERSITÉ DU QUÉBEC



Software Functional Size with

ISO 19761: 2003

COSMIC-FFP Measurement Method

Proposed Measurement Etalon: Valve Control System

Source of the Documented Requirements:

ISO TR 14143-4: 2000 Set RUR B.9

02 February 2005

DOCUMENT CONTROL

Date	Reviewers (s)	Modifications / Additions
04/11/2000		First draft, issued to Cosmic-Core Team for review
2003-01-20	Abran, O'Neill	
2004-08-16	Adel Khelifi	Whole document revisited, and sections added including: Improvements to specifications, Use Case Diagram and Sequence Diagram
2004-08-16	Alain Abran	Revision and reformatting + Addition of the ISO RUR at the beginning of the document
2004-08-21	Adel Khelifi	Revising Figures and other items
2004-08-23	Chantal Roy	Update of formatting style
2004-08-25	Charles Symons	Full revision of sections 1 to 3.7
2004-09-01	Adel Khelifi	Full revision of section 3.8 and on
2004-09-06	Charles Symons	Comments on revision
2004-09-07	Adel Khelifi	Implementation of comments
2004-09-17	Marie O'Neill	Comments on revision
2004-09-19	Adel Khelifi	Implementation of comments
2005-02-02	Charles Symons	Proposed editorial improvements

Copyright 2005. All Rights Reserved. Software Engineering Research Laboratory of École de Technologie Supérieure - Université du Québec (Canada). To copy all or part of this material requires a written permission of the Software Engineering Research Laboratory of École de Technologie Supérieure.

10.4 Overview

10.4.1 Introduction

The Valve Control system is documented in the ISO technical report: ISO/IEC TR 14143-4 (Version 2000). This ISO document provides various sets of Reference User Requirements (RUR), described usually in a textual format. The purpose of this ISO document is to provide publicly available sets of requirements to researchers and practitioners to be used as input for measuring functional sizes of software.

The Valve Control system used in this proposed measurement etalon corresponds to set RUR B.9 of this ISO document. This set RUR B.9 documents the requirements for a real-time system to control the behaviour of a control valve: the control valve is open by default and closes to engage the gear change mechanism of an automatic transmission installed in a land vehicle.

General information about the application to be measured

- Application domain: Control
- Application type: Real-time

Available information about the application

- Document: Requirements in a text format, as documented in from ISO 14143-4: 2000 RUR B.9

10.4.2 Measurement viewpoint, purpose and scope

For the purposes of the proposed measurement etalon, the following is given.

Measurement viewpoint:

The measurement viewpoint in this proposed measurement etalon is that of the software developer who is interested in quantifying the functionality of the software he has to develop.

Measurement purpose:

The measurement purpose is to measure all of the Functional User Requirements (FUR) of the software requirements documented in the set of Reference User Requirements selected for this proposed measurement etalon using the COSMIC-FFP functional sizing method. FUR are a sub-set of the RUR.

Measurement scope:

The measurement scope is all of the software Functional User Requirements within the set RUR B.9, and only these. The measurement scope is therefore a subset of the system requirements documented in this ISO proposed measurement etalon, that is, only those related to software, and not those related to the hardware.

10.5 Requirements – as documented in ISO 14143-4:2000

10.5.1 Context

The requirements below describe the behaviour of the solenoid control valve on a hydraulic circuit valve controlling a mechanical device for changing gear on an automatic transmission installed in a land vehicle. The valve can be open or closed: it is open by default and closed to engage the gear change mechanism. The process controls the amount of time the valve is closed during an operating cycle of several thousand microseconds. A clock supplying the operating cycle reference triggers the process.

10.5.2 Input

The process uses as input:

- A sensor signal (G_c) indicating gear change is in progress (value 1) or not (value 0),
- A sensor signal (S_u) indicating, during gear change, if shifting to upper gear (value 1) or lower gear (value 0),
- A sensor signal (I_{dl}) indicating whether the transmission is under stress (value 0) or idling (value 1),
- A binary flag “A” whose value is stored in the processor ROM memory or
- A binary flag “B” whose value is stored in the processor ROM memory.

Binary flags “A” and “B” describe some general configuration characteristics of the automatic transmission.

10.5.3 Output

The process produces as output:

Time (T), during one operating cycle, during which the control valve must be closed.

10.6 Requirements

10.6.1 Part a – determine general operating condition

Determine whether operating slowly or quickly from the closed state of the hydraulic valve.

```

IF (   Gc = 1
      AND Idl = 1
      AND A = 0
      AND B = 0 )
THEN, operating under normal condition, perform PART B
IF (   Gc = 1
      AND Idl = 0
      AND Su = 1

```

AND A = 0
 AND B = 0)
 THEN, operating during gear change, perform PART C

10.6.2 Part b – control to open hydraulic valve slowly from its closed state

Reset T to the smaller value of either INIT or the value of T during the last process cycle, where INIT is a constant stored in the computer ROM memory,

Compute the new value of T: $T = T - (Cst_X * ET)$

Where Cst_X is a constant stored in the processor ROM memory and ET is the elapsed time since an action that opens the hydraulic valve slowly from its closed state has been activated.

Condition for completion:

If the following conditions are met then valve control is passed to another process:

T is smaller or equal to LT

OR

Slp is greater or equal to Uslp

Where LT is a lower threshold of time and Uslp is an upper threshold of the amount of slip stored in the processor ROM memory. Slp is the current amount of slip, which denotes the difference of number of revolutions between the engine output shaft and the power train shaft. The value is computed and updated according to the following formula and stored in the processor RAM memory.

$$Slp = |E_{rev} - PS_{rev}|$$

where E_{rev} is the engine's output shaft revolutions and PS_{rev} is the power train shaft revolutions. Both variables' values are supplied by concurrent processes using input from separate sensors and placing the calculated result in the processor RAM memory.

10.6.3 Part c – control to open hydraulic valve quickly from its closed state

- Reset T to the smaller value of either $INITS(V_s)$ or the value of T during the last processing cycle, where $INITS$ is a table of initial values stored in the processor ROM memory and V_s is the vehicle speed which is computed and updated by another process and stored in the computer RAM memory.
- Compute the new value of T: $T = T - (INCR(V_s) * ET)$ where $INCR$ is a table of increments which depend on the speed of the vehicle stored in the processor ROM memory and ET is the elapsed time since an action to close the hydraulic valve quickly from its closed state has been activated,
- Condition for completion: if the following conditions are met then valve control is passed to another process:

T is smaller or equal to LT. LT is a lower time threshold stored in the ROM memory.

NOTE: From a functional size measurement perspective, the data attributes and data structures preserved in the processor ROM and RAM memory are considered to reside within the software boundary.

10.7 COSMIC-FFP measurement procedure

10.7.1 Identification of layers

There is a single software layer for this set of requirements

10.7.2 Identification of users

The users that interact with this software are the following mechanical devices:

1. Send information to the software: Clock and Sensors (GC, Su and IDL);
2. Receives information from the software: A Control Valve.

From the requirements, as written, there are no human users, nor are there other software applications interacting with this software.

10.7.3 Boundary

Based on the written requirements, we can identify the following software boundary.

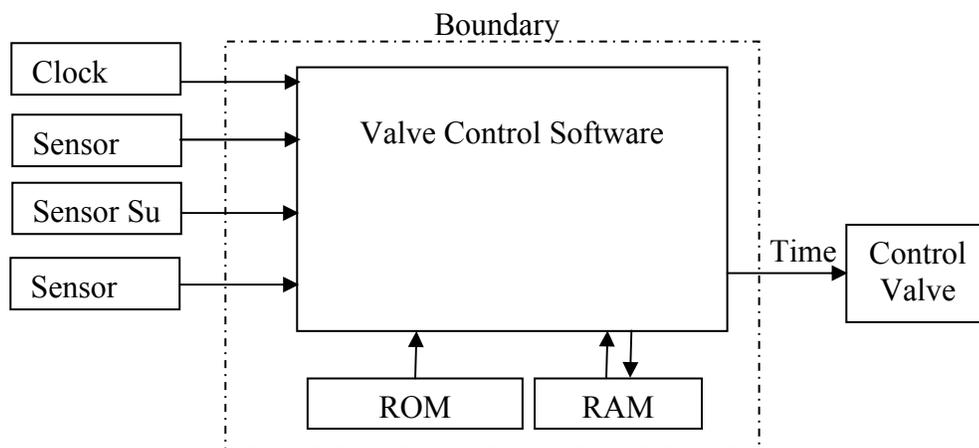


Figure 23 Software Boundary

10.7.4 Identification of triggering events

From the documented requirements a single triggering event is identified:

-A Clock supplying the operating cycle reference which triggers the process.

10.7.5 Identification of data groups

From the documented requirements, the following data groups are identified:

Data Sources/ Destinations	Objects of Interest	Data groups	Data attributes
Clock	Op cycle trigger event	Clock signal	Clock signal
Sensors	Gear-change status Shift direction Stress/idle status	Gc Su Idl	Gc Su Idl
ROM	Valve-type 'X'	Valve-type X's fixed parameter set for the Valve Control process	Flag A or Flag B INIT Cst_X LT Uslp INITS 1, 2, 3 etc INCR 1, 2, 3, etc
RAM	Engine Power train Vehicle Period since last action	E _{rev} PS _{rev} Vs ET	E _{rev} PS _{rev} Vs ET
Control Valve	Period valve-to-be-closed	T	T

Note 1: Much data needed by the process must be obtained from ROM. We assume that all these data are attributes of one Object of interest, namely the 'fixed parameter set for the Valve Control process'; this parameter set may be unique to this Valve-type and even to this automatic transmission, perhaps even to this vehicle. This assumption is justified as follows:

- Physically, it is likely that the data needed for any one cycle is obtained at one time from the ROM and not as a succession of Reads
- The COSMIC-FFP 'de-duplication' rule assumes that all data needed for any one Object of interest is obtained in only one data movement.

N.B. This assumption may be incorrect. The ROM(s) may store other data for other functional processes. If we had this wider knowledge, we might find that the data is

organized into groups for more than one Object of interest. But we do not have this wider knowledge and so we make this simple assumption (based on the RUR as documented).

Note 2: There are three parameters that are, explicitly according to the requirements, obtained from RAM and provided by other processes. The requirements do not state if these other processes are hardware or software but this is not relevant for our measurement purpose as defined in the measurement scope. These three parameters have to be Read from RAM by the Valve Control process.

Note 3: The requirements, as documented, do not specify whether the Elapsed times (which are defined differently for Parts B and C) are given by the hardware, or whether they are calculated by the software, nor do the requirements state where the ET is obtained from. For the purposes of this proposed measurement etalon, the following system decision was taken as an assumption: the ET is provided by another process and the Valve Control process obtains it from the RAM. Should another system decision be made, that is of allocating to the hardware the calculation of elapsed time or to a function within this functional process, another functional measurement would have to consider this added function to be developed and integrated within the software.

Note 4: S_{lp} is calculated on each cycle and is not made persistent between cycles. It is therefore the result of data manipulation and is not involved in any data movement according to the COSMIC-FFP method.

10.7.6 Identification of functional processes

From the documented requirements with a single triggering event, there is one candidate functional process which is:

- The Control of time during the operating cycle of the control valve.

The measurement procedure must assess whether a candidate functional process is a COSMIC-FFP one or not.

Each candidate process must satisfy the following questions in order to be validated as a COSMIC-FFP functional process:

- Does it operate on a unique and ordered set of data movements performing a set of FURs?
- Is it triggered by an event (triggering event)?
- Does the triggering event occur outside the boundary of the software?

These questions are investigated next:

Time-based control of the valve

Question	Answer	Comments
Does it operate on a unique and ordered set of data movements performing a set of FURs?	Yes	
Is it triggered by an event?	Yes	Clock signal event
Does the triggering event occur outside the boundary of the software?	Yes	The clock is outside of the software – see Figure 1

10.8 Identify data movements

For the single functional process in the previous steps, all data movements of a data group must be identified.

In this proposed measurement etalon, the Use Case (figure 24) and the Sequence Diagrams (figures 25 and 26) have been prepared to facilitate the identification of the data movements, and to ensure that all data movements have been identified.

10.8.1 Use case diagrams

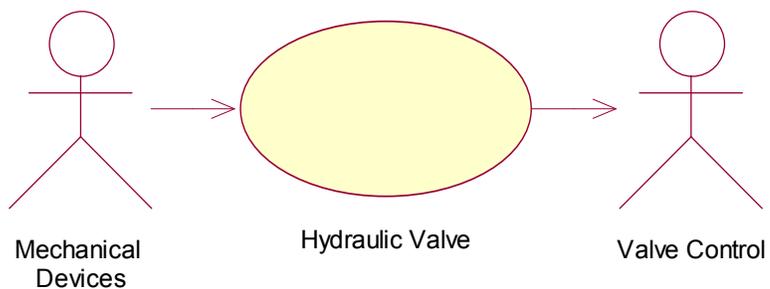


Figure 24 Valve Control System - Use Case Diagram

10.8.2 Message sequence diagram

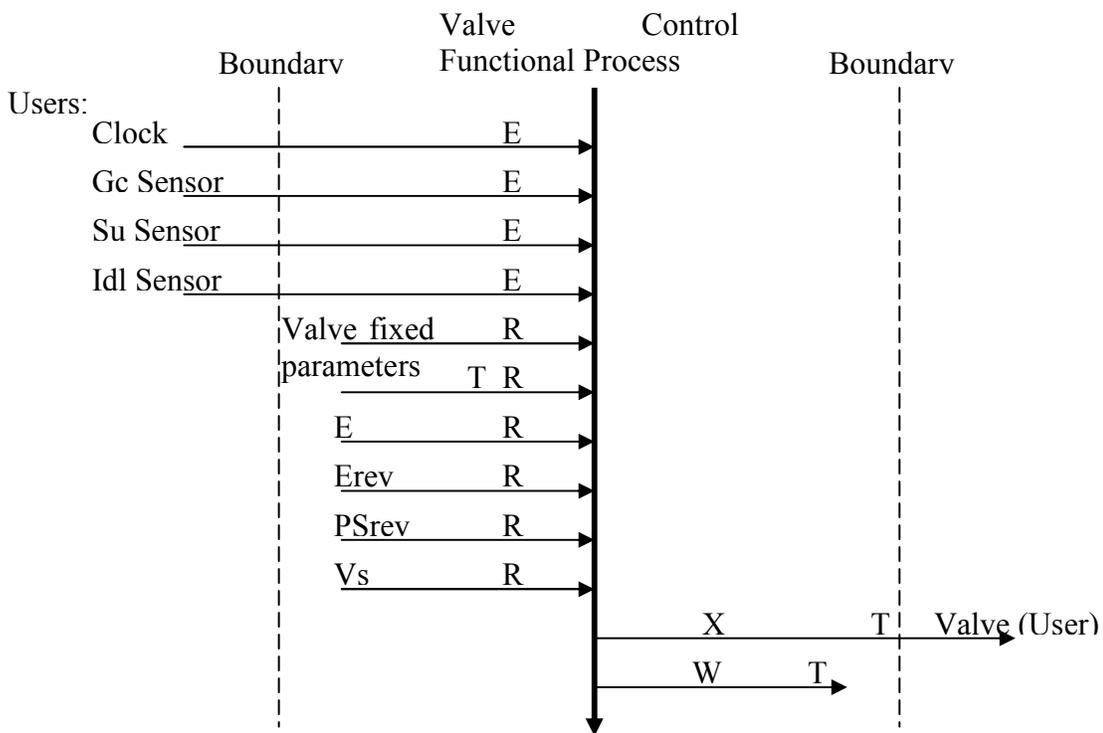


Figure 25 Valve Control Application - Message Sequence Diagram

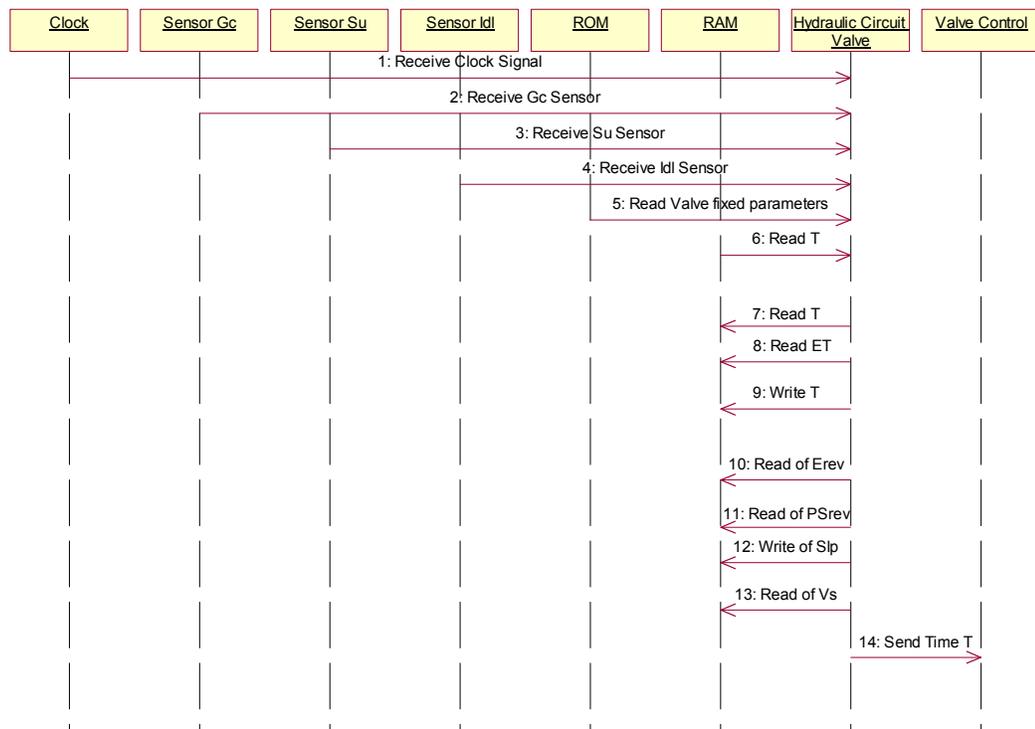


Figure 26 Valve Control System - Sequence Diagramme

10.8.3 List of data movements

The detailed list of the data movements identified is presented in table 31.

The following labelling convention is used: the functional process is assigned an ID number with the following format: x.y where x is the layer ID and y is the ID number of the functional process within the layer. Here this is a single layer and a single functional process that is 1.1

Tableau XXXI

List of Cosmic-FFP data movements

No	ID Process	Process description	Triggering event	Data movements Identification	Data Group	Data movement Type	Cfsu	
1	1.1	Time-based control of the valve	Clock Cycle Signal	Receive Clock Cycle Signal	Clock	E	1	
				Receive signal of Gc Sensor	Gc	E	1	
				Receive signal of Su gear change	Su	E	1	
				Receive signal of Idl Sensor	Idl	E	1	
				Read Valve fixed parameters	Valve fixed parm.	R	1	
				Read of T from RAM	T	R	1	
				Read ET from RAM	ET	R	1	
				Read Erev from RAM	E _{rev}	R	1	
				Read PSrev from RAM	PS _{rev}	R	1	
				Read Vs from RAM	Vs	R	1	
				Send T to the Control Valve	T	X	1	
				Write T to RAM	T	W	1	
		Total functional size in Cfsu =				ΣCfsu		12

10.8.4 Observations on the clarity of the documented requirements

Even though the documented requirements used for this proposed measurement etalon are coming from an ISO technical report, there is no mention in this report about the quality of these requirements.

IEEE Std 830-1998 recommends that requirements meet the following quality criteria:

- Correct;
- Unambiguous;
- Complete;
- Consistent;
- Ranked for importance and/or stability;
- Verifiable;
- Modifiable;
- Traceable.

In the ISO technical report, there is no claim that their sets of documented requirements meet the quality criteria specified in IEEE 830.

The following ambiguities have therefore been noted:

1. From the documented requirements, it is not clear neither where the control of the valve is performed nor where to send the time T after the calculation. For this measurement, the following assumptions was made: T is sent to the Control Valve
2. It is not clear when the application interacts with the ROM and the RAM. The specifications do not make clear the sequence in which any of the data movements occur. This observation does not modify the result of measurement of the functional size.
3. In the parts B & C of the requirements, there is no indication for actions to take if the conditions for completion are not met.
4. It is not documented where the variable ET (Elapsed time) comes from (that is, the elapsed time since an action that opened the hydraulic valve slowly or closed it quickly). For the measurement of this proposed measurement etalon, this assumption was made: ET is read from the RAM.
5. The parameters Erev, PSrev and Vs are all stated to be supplied into RAM by 'other concurrent functional processes'. Since these are seemingly asynchronous with the Valve Control functional process being measured it seems correct to assume that they are all attributes of separate Objects of interest, thus requiring separate Reads.
6. Note that the parameter T must be made persistent in RAM between each cycle of the Valve Control functional process. Therefore, each cycle of this process must Read the value from the last cycle and Write the latest value for use by the next cycle.

10.9 Analysis of measurement results

For this software application, the total functional size according to the COSMIC-FFP method is obtained with the addition of all data movements within its single functional process, that is 12 Cfsu (ISO 19761: 2003), as indicated at the bottom of table 32. The distribution of size units by data movement types is presented in table 33. Of course 100% of the data movements are within this single functional process – see table 34.

Table XXXII

Distribution of size units

No	Process ID	Process Identification	Data Movement Types				Cfsu
			E	X	R	W	
1	1.1	Time-based control of the valve	4	1	6	1	12
Summary		1 Functional Process	4	1	6	1	12

Table XXXIII

Contribution of functional process to total Cfsu size

No	Process ID	Process Identification	Cfsu	Application Percentage (%)
1	1.1	Time-based control of the valve	12	100
Total:			12	100 %

The percentages of size by data movement types are presented in table 34 and figure 27.

Table XXXIV

Percentage of the Cosmic-FFP data movement types

COSMIC-FFP Data movement types	Cfsu	Percentage (%)
Entry (E)	4	34
Exit (X)	1	8
Read (R)	6	50

Table XXXIV (Suite)

Percentage of the Cosmic-FFP data movement types

COSMIC-FFP Data movement types	Cfsu	Percentage (%)
Write (W)	1	8
Total :	12 (Cfsu)	100 (%)

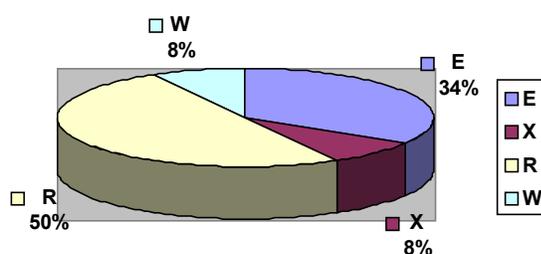


Figure 27 Proportion of COSMIC FFP data movement types

10.10 Conclusions and observations

This proposed measurement etalon has provided an illustration of the functional size measurement with ISO 19761: 2003 COSMIC-FFP.

The measurement is based on the requirements of RUR B.9 of ISO TR 14143-4, as documented in a textual format.

During the measurement process, uncertainties and ambiguities about the documented requirements have been noted. It was also observed that these requirements do not meet all of the quality criteria listed in IEEE 830.

The measurement solution presented takes into account some clarifications that have been required to allocate for instance the Elapsed time calculation function to hardware.

If the assumptions that have been made as a result of the uncertainties and ambiguities in the requirements should prove to be incorrect, then the measured functional size could change.

10.11 Questions and answers

Question 1: The following is written in the specifications: 'Reset T to the smaller value of either INIT or the value of T during the last process cycle'. Why is a Read data movement on T valid?

Answer to Question 1:

Because T is always updated according to the hydraulic valve operating state.

Question 2: It is written in the documented requirements that the Flags A and B are 'Inputs' to the software. When the software accesses the information in Flags A and B, why in COSMIC-FFP are these data movements considered as Reads and not considered as Entries?

Answer to Question 2:

The vocabulary in the documented requirements is not standardized. Even though Flags A and B are mentioned as 'inputs', it is also written that they are coming from a ROM memory. In COSMIC-FFP, information in a ROM is considered as persistent data within the software boundary. Any access to these data is then considered as a Read, and not as an Entry. Furthermore, the flags A and B are not considered as 'users' that interact with this software, and therefore data groups coming from the ROM are not considered as Entries, in the COSMIC-FFP method.

Question 3: What is the impact if the system requirements are changed and the elapsed time is to be implemented as a hardware function?

Answer Question 3:

If ET comes from hardware, then a ‘user’ (hardware source) should be shown on the diagram for ET and ET should NOT be shown in the RAM.

Then: ET comes as input from hardware (count one Entry instead of one Read). The overall size is unchanged.

Question 4: What is the impact if the software functional requirements are changed and the functional process is to provide itself with the elapsed time?

Answer Question 4:

We have assumed in the given solution that ET is provided by another process and the Valve Control process obtains it from RAM (count one Read)

Alternatively we might assume that this Valve Control process calculates and keeps track from one cycle to another. To do this the Valve Control process would need to Write a time-stamp to RAM between cycles (so count one Read and one Write per cycle of the time-stamp). The overall size would increase by one Cfsu from the solution given in 5. above.

10.12 Sommaire

Ce chapitre a présenté le contenu d’un étalon en décrivant toutes ses sections. Il a exposé en détail le Valve Control System Etalon afin de donner une illustration pratique de chaque section qui existe dans le gabarit d’étalon de mesure de la taille fonctionnelle des logiciels.

Le chapitre suivant décrit tous les étalons proposés dans le référentiel. Il fournit une fiche technique pour chaque étalon afin de montrer son contenu.

CHAPITRE 11

LE RÉFÉRENTIEL POUR LA MTF

Ce chapitre présente un résumé des résultats trouvés dans cette recherche. Il décrit les résultats obtenus au cours de cette étude qui vise à développer un référentiel pour la mesure de la taille fonctionnelle des logiciels.

Le résumé de chaque étalon est d'abord rapporté en langue anglaise car plusieurs des partenaires dans cette recherche ne parlent pas français. La fiche résumé décrit les sections les plus importantes de l'étalon en format abrégé. Elle présente le nom de l'étalon, son abréviation et sa date de réalisation. La fiche donne un aperçu sur le logiciel, des informations sur l'origine de ses fonctionnalités utilisateurs requises (FUR), son domaine d'application, son type d'application, ses utilisateurs et ses couches. De plus, elle résume le résultat de la mesure et elle indique les diagrammes UML développés pour chaque étalon ainsi que les observations relevées sur la clarté de la documentation du logiciel à mesurer. Après la présentation des résultats obtenus dans cette thèse, le chapitre apporte une brève évaluation des résultats obtenus par rapport à la problématique principale traitée dans cette recherche.

11.1 Introduction

La réalisation du système universel de mesure est aussi importante que l'élaboration du système judiciaire et du système monétaire pour tous les pays du monde. Il contribue à la gestion des aspects les plus fondamentaux de nos vies à tous.

Dans cette thèse, le référentiel peut constituer une aide à la décision qui pourra être utilisée pour organiser, pour choisir, pour communiquer et pour évaluer les attributs requis de la MTF pour un logiciel.

11.2 Description du référentiel

La figure 28 présente le processus de construction du référentiel pour la mesure de la taille fonctionnelle des logiciels :

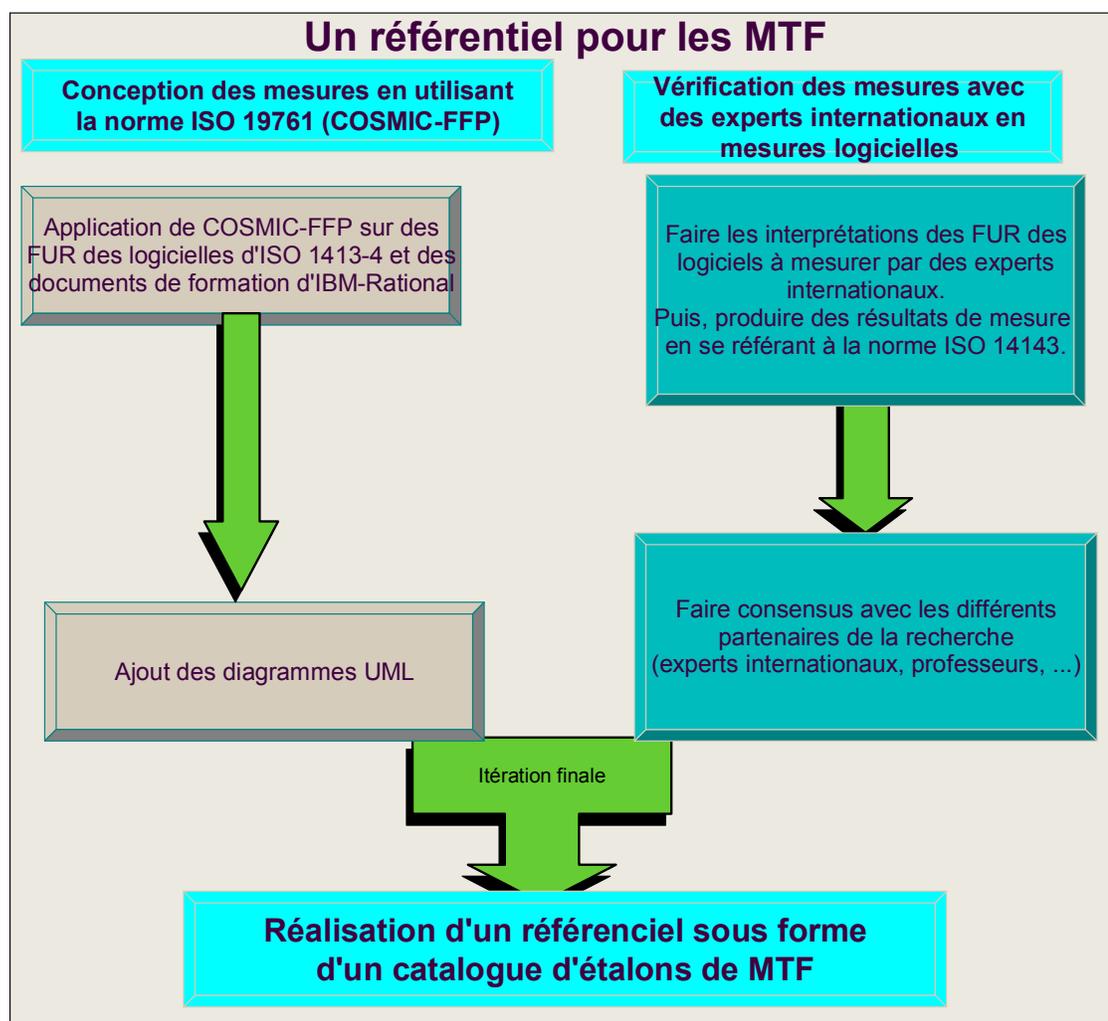


Figure 28 Le référentiel

L'objectif du projet est de construire un référentiel composé de huit étalons de la MTF des logiciels, mais bien entendu cela ne veut pas dire que la première version des étalons sera parfaite (e.g. mature). En effet, pour une mesure mature, comme la distance, la

masse ou le temps, il n'y a qu'un seul étalon pour toute la nation, appelé étalon national. Ce dernier est développé à partir d'un étalon international.

Cet étalon est développé avec la plus grande précision possible selon les connaissances et les technologies disponibles, d'après les expériences et les observations des personnes chargées de sa détermination. L'étalon évolue avec le temps et le progrès des connaissances et des technologies. L'histoire du mètre en fait un bon exemple de l'évolution de l'étalon de mesure. Il est déposé auprès des autorités concernées (normalement le BIPM : Bureau International des Poids et Mesures) avec tous les rapports des actes qui ont servi à le déterminer, afin qu'on puisse le vérifier en tout temps. À partir de l'étalon national, des unités gouvernementales s'occupent de faire des modèles conformes. Ces modèles serviront à la fabrication de toutes les sortes de mesures destinées aux usages des citoyens.

Vu l'aspect novateur de cette recherche exploratoire dans la mesure du logiciel, il n'est pas possible d'atteindre dès le départ une parfaite précision des étalons du système international d'unités. Cette thèse propose plutôt des résultats de mesure de la taille fonctionnelle des logiciels, vérifiés par des experts internationaux du domaine et suffisamment exacts pour être acceptés comme une première génération d'étalons de la MTF des logiciels en utilisant la norme ISO 19761 comme norme pour le mesurage. Un des moyens d'action dans ce travail est la collaboration des experts internationaux et des industriels qui opèrent dans le domaine de la mesure des logiciels.

D'ailleurs, il faudra s'attendre à produire des versions successives (imparfaites), puis progressivement des versions améliorées du référentiel en éliminant à chaque itération des causes successives d'erreur. Le but de cette recherche exploratoire est de produire un ensemble d'étalons qui peut servir comme référence pour le génie logiciel, comme le mètre est le standard pour les mesures de longueur, telles que le fermi (10^{-5} m), l'angstrom (10^{-10} m), le mile nautique (1852 m), l'astron ($1,496 \cdot 10^{11}$ m) et le parsec ($3,0857 \cdot 10^{16}$ m).

La stratégie particulière pour choisir quelle cause d'erreur nous devons éliminer à chaque itération afin d'améliorer les étalons consiste en l'analyse des résultats de chaque itération et, à partir de ces analyses, l'évaluation de la prochaine source d'erreurs à éliminer.

Il est à noter que la base des premières expérimentations pour construire le référentiel a été choisie initialement comme fixe : il s'agit d'une partie du contenu du document ISO 14143-4, soit cinq FUR (*Functional User Requirements*) qui sont *Automatic Line Switching System*, *Hotel Reservation System*, *L-Euchre Application*, *SAGA System* et *Valve Control System*. Une FUR appartient à l'application *Rice Cooker*. Les deux autres appartiennent aux documents de formation du processus RUP : ce sont *C-Registration System* et *Collegiate Sports Paging System*.

L'application de la méthode COSMIC-FFP sur les FUR des logiciels à mesurer, pour développer les étalons, a permis de porter un jugement sur la clarté de la documentation des logiciels à mesurer.

Bien que les FUR d'ISO soient documentés dans un document officiel ISO, il n'y a aucune indication sur leur qualité. En effet, ces FUR ont été donnés par divers intervenants à ISO pour servir de base commune à des étapes de vérification des MTF des logiciels, mais chacune de ces FUR qui est d'origine diverse, n'a pas été écrite selon des normes communes. Il n'y a même aucune information connue sur leur qualité, ni de traçabilité à des critères spécifiques de qualité. Actuellement, leur principale qualité est qu'elles existent, qu'elles sont documentées (telles quelles) et qu'étant un document officiel ISO, elles sont disponibles à tout chercheur (et ingénieur logiciels) qui en achète une copie à ISO.

Les exigences logicielles fournies par la compagnie IBM-Rational dans leur documentation de formation du processus RUP sont illustrées par des diagrammes UML.

L'application de la méthode de mesure COSMIC-FFP (ISO 19761) a donné les résultats présentés dans la section qui suit.

11.3 Description des résultats trouvés

Pour présenter les résultats de façon concise, une « fiche résumé » a été conçue. La «fiche résumé» récapitule en anglais, pour conserver le texte d'origine, les différentes sections de la mesure :

- Le nom de l'étalon permet d'identifier l'étalon;
- L'abréviation permet de donner un nom abrégé à l'étalon;
- La Date présente la date de réalisation de l'étalon;
- L'aperçu sur le logiciel mesuré donne un bref résumé sur sa fonction principale;
- L'information du logiciel décrit l'origine du document des fonctionnalités utilisateurs requises (FUR), le type de logiciel et son domaine d'application;
- L'identification des couches et des utilisateurs détermine le nombre de couches et les utilisateurs du logiciel;
- Le résumé de la mesure liste les processus fonctionnels identifiés du logiciel avec leur taille ainsi que le nombre de processus fonctionnels total du logiciel;
- Les Diagrammes UML énumèrent les diagrammes des cas d'utilisation et de séquence développés pour l'étalon;
- Les observations sur la clarté des exigences donnent une brève évaluation de la qualité de la documentation fournie.

Le même gabarit de fiche est utilisé pour tous les étalons.

Dans cette thèse, les étalons sont présentés selon l'ordre chronologique de leur réalisation par l'équipe impliquée dans ce travail.

11.3.1 Premier étalon

Le premier étalon de mesure s'appelle *Automatic Line Switching Etalon*. Les FUR du logiciel *Automatic Line Switching* sont documentées dans le rapport technique ISO/IEC TR 14143-4 (version 2000). L'*Automatic Line Switching* utilisé correspond à l'ensemble RUR B.8 du document ISO14143-4. Cet ensemble RUR B.8 documente les FUR d'un système de commutation, en temps réel, entre une ligne de travail et une ligne de secours qui sont données pour une chaîne de transmission : si la ligne de travail se dégrade ou échoue la ligne de secours est employée à la place.

Fiche résumé du premier étalon

<i>Etalon Name : Automatic Line Switching</i>	<i>Abbreviation :ALS</i>	<i>Date : March 2005</i>
Measured Software Overview	Software Information	
<p>The Automatic Line Switching system is documented in the ISO technical report: ISO/IEC TR 14143-4 (Version 2000). This ISO document provides various sets of Functional User Requirements (FUR), described usually in a textual formal. The purpose of this ISO document is to provide to researchers and practitioners with sets of requirements to be used as publicly available documents for measuring the functional size of software.</p> <p>The Automatic Line Switching system used in this case study corresponds to set RUR B.8 of this ISO document. This set RUR B.8 documents the requirements for a real-time system to switch between a Work Line and a Backup Line which are provided for a communication channel: if the Work Line degrades or fails the Backup Line is used instead.</p>	<p><i>FUR Origine : ISO TR 14143-4: 2000</i> Set RUR B.8 <i>Application domain : Control</i> <i>Application type : Real Time</i></p>	
	Layers and Users Identification	
	<p><i>Layers Number : 1</i> <i>Users :</i> - Technician - Work line – Quality level change monitor - Backup Line - Quality level change monitor - Switch to the other Line <i>Triggering events Number: 14</i></p>	
Measurement Summary	Etalon UML Diagrammes	

Process Name in Cfsu	Size	<ol style="list-style-type: none"> 1. Automatic Line Switching System - Use Case Diagram 2. Automatic Line Switching - Remove Work Line Sequence Diagram 3. Automatic Line Switching Application – Forced Switch to Work Line Sequence Diagram 4. Automatic Line Switching Application – Work Line degraded Sequence Diagram.
Remove Work Line	5	
Remove Backup Line	5	
Restore Work Line	5	
Restore Backup Line	5	
Forced switch to Work Line	3	
Forced switch to Backup Line	3	
Conditional switch to Work Line	5	
Conditional switch to Backup Line	5	
Work Line degraded	5	
Backup Line degraded	5	
Work Line failed	5	
Backup Line failed	5	
Work Line cleared	5	
Backup Line cleared	5	
Total Number of processes Size in Cfsu	Total	Observations on requirements' clarity
14	66	<ol style="list-style-type: none"> 1. It is not clear if the status and the quality of the lines are recorded. Neither is it specified where or how they should be saved. 2. There is no indication of how messages are sent or received between processes or between users and processes. 3. There is no information about the exit from the process when it is finished. 4. The Technician doesn't know the currently selected line. This will be confusing to him when he wants to return back to the beginning of the session. 5. The Requirements do not ensure the security of having acceptable quality if one line is out of service. 6. The Requirements do not cover the case where a line of 'normal' quality receives a signal that it has 'failed', without first passing through the 'degraded' quality level.

<p>The AUS-ONLINE organisation provides the interface between the user's videotext terminal and the SAVAT Corporation's mail order application. The AUS-ONLINE interface system is called Presentation/9 videotext system.</p> <p>AUS-ONLINE plan to decommission the Presentation/9 videotext system and replace it with a multi-media system which will allow the user to use a variety of different types of terminals. When AUS-ONLINE implement their new presentation software they will need to provide an alternative means of public access to the current Presentation/9 Service Providers, such as SAVAT.</p>		
Measurement Summary		Etalon UML Diagrammes
Process Name	Size in Cfsu	
Logon to the SAVAT System	7	1. Gateway System - Use Case Diagram
Change Password	11	2. Logon to the SAVAT System - Sequence Diagram
Logoff from SAVAT System	6	3. Change Password - Sequence Diagram
Order Goods	7	4. Logoff from SAVAT System - Sequence Diagram
List Orders for Account	10	5. Order Goods - Sequence Diagram
Order Account Statement	7	6. List Orders for Account - Sequence Diagram
Display Statement Summary	9	7. Order Account Statement - Sequence Diagram
Ad hoc Catalogue Order	3	8. Display Statement Summary - Sequence Diagram
Freeze Logon ID Request	4	9. Ad hoc Catalogue Order - Sequence Diagram
Register Product Category	8	10. Freeze Logon ID Request - Sequence Diagram
Delete Product Category	12	11. Register Product Category Selection - Sequence Diagram
Product Category List	4	12. Delete Product Category from Selection - Sequence Diagram
Users Product Category List	5	13. Product Category List - Sequence Diagram
Get Account Balance	4	14. Users Product Category List -
Get Account Serial Number	4	
Get Last 'n' transactions	4	
Get Serial Number	4	
Error Handling	3	
Full Screen Errors	5	
Total Number of processes	Total Size in Cfsu	
	19	117

	<p>Sequence Diagram</p> <p>15. Get Account Balance - Sequence Diagram</p> <p>16. Get Account Serial Number - Sequence Diagram</p> <p>17. Get Last 'n' transactions for the User's Account - Sequence Diagram</p> <p>18. Get Serial Number for Product Category - Sequence Diagram</p>
	<p>Observations on requirements' clarity</p> <p>Many ambiguities have therefore been noted in the Requirements. For example:</p> <ol style="list-style-type: none"> 1) The paragraphs of the B.10.5 section present the functionalities of the system mixed together without reference to software layers. Specially the B.10.5.3.2 section, which refers directly to requests for X.25 connection as well as videotext terminal windows. It would be preferable to present the layers directly. 2) The figures and tables have no names neither numbers. An identification of them would facilitate the reference. 3) The nomenclature is confusing, for instance, in the document many phrases used to mean the same thing. For example, there are « Call request », «Connection request» and «a connection is established» to mean a connection request. 4) In the Logon function, they make no difference between network error and system error. 5) In B.10.5.3.2, second sentence we can't know the "On receiving ..." refers to what? In the same section,

	there are many “IF” there’s no “Then”. So, we don’t know what happens if the condition is not satisfied.
--	--

11.3.3 Troisième étalon

Le troisième étalon de mesure s'appelle *Valve Control System Etalon*. Les FUR du logiciel *Valve Control System* sont documentées dans le rapport technique ISO/IEC TR 14143-4 (version 2000). Le *Valve Control System* utilisé, correspond à l'ensemble RUR B.9 du document ISO14143-4. Cet ensemble RUR B.9 documente les FUR d'un système en temps réel pour contrôler le comportement d'une soupape de contrôle : la soupape de contrôle est ouverte par défaut et se ferme pour engager le mécanisme de changement de vitesse d'une transmission automatique installée dans un véhicule de terrain. Le détail de cet étalon était présenté dans le chapitre précédent afin de décrire le contenu de toutes les sections du gabarit d'étalon.

Fiche résumé du troisième étalon

<i>Etalon Name</i> : Valve Control System	<i>Abbreviation</i> : VCS	<i>Date</i> : February 2005
Measured Software Overview	Software Information	
<p>The Valve Control system is documented in the ISO technical report: ISO/IEC TR 14143-4 (Version 2000). This ISO document provides various sets of Reference User Requirements (RUR), described usually in a textual formal. The purpose of this ISO document is to provide publicly available sets of requirements to researchers and practitioners to be used as input for measuring functional sizes of software. The Valve Control system used in this case study corresponds to set RUR B.9 of this ISO document. This set RUR B.9 documents the requirements for a real-time system to control the behaviour of a control valve: the control valve is open by default and closes to engage the gear change mechanism of an automatic transmission installed in a land vehicle.</p>	<p><i>FUR Origine</i> : ISO TR 14143-4: 2000 Set RUR B.9</p> <p><i>Application domain</i> : Control</p> <p><i>Application type</i> : Real Time</p>	
	Layers and Users Identification	
	<p><i>Layers Number</i> : 1</p> <p><i>Users</i> :</p> <ul style="list-style-type: none"> - Clock - Sensors: GC, Su and IDL - A Control Valve <p><i>Triggering events Number</i>: 1</p>	
Measurement Summary	Etalon UML Diagrammes	
Process Name Cfsu	Size in	1. Valve Control System - Use Case Diagram

Time-based control of the valve 12	2. Valve Control System - Sequence Diagram
Total Number of processes Cfsu	Observations on requirements' clarity
1 12	<ol style="list-style-type: none"> 1. From the documented requirements, it is not clear neither where the control of the valve is performed nor where to send the time T after the calculation. For this measurement, the following assumptions were made: T is sent to the Control Valve 2. It is not clear when the application interacts with the ROM and the RAM. The specifications do not make clear the sequence in which any of the data movements occur. This observation does not modify the result of measurement of the functional size. 3. In the parts B & C of the requirements, there is no indication for actions to take if the conditions for completion are not met. 4. It is not documented where the variable ET (Elapsed time) comes from (that is, the elapsed time since an action that opened the hydraulic valve slowly or closed it quickly). For the measurement of this case study, this assumption was made: ET is read from the RAM. 5. The parameters Erev, PSrev and Vs are all stated to be supplied into RAM by 'other concurrent functional processes'. Since these are seemingly asynchronous with the Valve Control functional process being

	<p>measured it seems correct to assume that they are all attributes of separate Objects of interest, thus requiring separate Reads.</p> <p>6. Note that the parameter T must be made persistent in RAM between each cycle of the Valve Control functional process. Therefore, each cycle of this process must Read the value from the last cycle and Write the latest value for use by the next cycle.</p>
--	--

11.3.4 Quatrième étalon

Le quatrième étalon de mesure s'appelle *Hotel Reservation System Etalon*. Les FUR du logiciel *Hotel Reservation System* sont documentées dans le rapport technique ISO/IEC TR 14143-4 (version 2000). Le *Hotel Reservation System* utilisé correspond à l'ensemble RUR A.1 du document ISO14143-4. Cet ensemble RUR A.1 documente les FUR d'un système de réservation qui fait partie d'un système général de logement d'hôtel. Le système de réservation offre la maintenance et la confirmation des réservations.

Fiche résumé du quatrième étalon

<i>Etalon Name : Hotel Accommodation System</i>	<i>Abbreviation</i> HAS	<i>Date</i> : February 2005
Measured Software Overview	Software Information	
<p>The Hotel Accommodation System (Reservation) is documented in the ISO technical report: ISO/IEC TR 14143-4 (Version 2000). This ISO document provides various sets of Functional User Requirements (FUR), described usually in a textual formal. The purpose of this ISO document is to provide to researchers and practitioners sets of requirements to be used as publicly available documents for measuring the functional size of software.</p> <p>The Hotel Accommodation System (Reservation) used in this case study corresponds to set RUR A.1 of this ISO document. This set RUR A.1 documents are the requirements for a reservation system which is part of an accommodation system of a general hotel system. The reservation system supports the following business functions related to the letting of hotel rooms:</p> <ul style="list-style-type: none"> - maintain reservations - confirm reservations. 	<p><i>FUR Origine</i> : ISO TR 14143-4: 2000 Set RUR A.1</p> <p><i>Application domain</i> : Business Application</p> <p><i>Application type</i> : MIS</p>	
	Layers and Users Identification	
	<p><i>Layers Number</i> : 1</p> <p><i>Users</i> :</p> <ul style="list-style-type: none"> - Agent - Other Hotel System parts - Screen - Printer - Files or Tables - Memory <p><i>Triggering events Number</i>: 5</p>	
Measurement Summary	Etalon UML Diagrammes	

Process Name	Size in Cfsu	Size in
Create Reservation	8	
Update Reservation	12	
Accept Reservation	8	
Confirm Reservation	10	
Select Reservation	8	
Cancel Reservation	12	
Room Type Report	8	
Total Number of processes		Total Size in
7	66	Cfsu
<ol style="list-style-type: none"> 1. Hotel Reservation System - Use Case Diagram 2. Create Reservation Sequence Diagram 3. Update Reservation Sequence Diagram 4. Confirm Reservation Sequence Diagram 5. Accept Reservation Sequence Diagram 6. Select Reservation Sequence Diagram 7. Select Reservation Sequence Diagram 8. Room Type Report Sequence Diagram. 		
Observations on requirements' clarity		
<p>The difficulties of interpretation of this statement of FUR all arise from lack of clarity in the specification, which is badly written. The text is badly structured, descriptions of 'create' and 'change' functions are inter-mixed at random and even the Scope of the measurement is very unclear.</p> <p>When a specification is so bad and obviously could never work sensibly in practice, the FSM analyst is tempted to make his own interpretations of how it should work. This adds to the variations you will get in measurements of functional size.</p> <p>Examples of lack of clarity.</p> <p>The main heading says 'Hotel Accommodation System (Reservation), which would point you towards measuring only the functionality shown in the second Menu, Option1 ('Reservation'). But under para</p>		

A.1.2.2.1 which describes the Reservation function, there is a statement 'If necessary more room types can be stored for the same period.' This could be interpreted as a function of Option 5 'Maintain Room Data'. However, I have taken it to mean that the customer can ask for more than one room-type for the same period. It is not shown how a clerk can continue to enter room types and numbers after the first type/number

The spec says that 'all data' must be entered in the reservation screen before the system will tell the customer if there are rooms available to satisfy his request. This includes a name and address and other details which it is crazy to enter at this stage. My guess is that this screen, as shown is also the output of an enquiry on a Reservation, since it also allows a Confirm or Change transaction

The relationships between 'confirm' Reservation on the RES screen and 'accept' a Reservation on the Accept Reservation (Details) screen and what happens 'when an accepted reservation is confirmed' according to para A.1.2.2.5 is quite unclear. The text says "If the request can be met, the acceptance screen ACP-RES stores the reservation and a confirmation of the reservation (CON-RES) is produced for the billing address. If the request cannot be met, room type report (RT-REP) is called to look up an alternative choice." The wording seems to suggest that these steps take place automatically (implying one complex functional process resulting from a single triggering event) but they could also take place as a result

of the clerk pressing buttons (implying conscious break points and hence separate functional processes). In the analysis below, because some buttons are provided, I have assumed some separation of the functional processes. Also, common sense says that the software design should allow for separate human-controlled decisions on what to do next at certain stages. But mine is not the only possible interpretation

Adding to the argument of the previous point, neither the Accept Reservation screen, nor the Confirmation of Reservation report contains any price information. Any normal system would have to provide this information, at least on the Accept Reservation screen. The point is important because the decision to Accept would not normally be automatic and therefore not part of the overall procedure of creating a reservation.

Naming is inconsistent (room-type/room class; invoice/billing; screen/report with 'report' used to mean both screen and report. In A.2.1, it mentions a 'reservation report'. This is either not mentioned in A.1 or it is called 'Acceptance report'. Descriptions do not match names – a 'billing address' is actually a 'person or institution'. 'Country' seems to be used for the country of the person making the reservation and also for 'Language code'.

The relationship between A.1 and A.2 is unclear and the content inconsistent. A.2 does not list a function to enquire on or to retrieve a reservation, which is

	<p>needed before you can change a reservation. Common-sense says it will be needed anyway</p> <p>The system would be grossly inefficient in operation because the only way of finding out if a room-type is available on a given date is to scan the entire list of reservations. The system does not keep track of availability of room-type by date.</p> <p>For all these reasons, it is IMPOSSIBLE to decide exactly what the functional processes are. This is independent of the FSM Method and this specification is not a fair test for checking consistency of sizing using ANY FSM Method. My analysis below is a 'best guess'. It is not the only possible interpretation. I changed my mind on the 'best guess' several times during my analysis, which is a sure indication of a lousy specification.</p>
--	---

11.3.5 Cinquième étalon

Le cinquième étalon de mesure s'appelle *L-Euchre Card Game System Etalon*. Les FUR du logiciel *L-Euchre Card Game System* sont documentées dans le rapport technique ISO/IEC TR 14143-4 (version 2000). Le *L-Euchre Card Game System* utilisé, correspond à l'ensemble RUR B.11 du document ISO14143-4. Cet ensemble RUR B.11 documente les FUR d'un système en temps réel. Le *L-Euchre* est un jeu de cartes d'euchre de quatre joueurs. Le jeu relie à distance quatre joueurs à travers Internet par le protocole de TCP/IP. Le même noyau de jeu doit s'exécuter sur l'ordinateur de chaque joueur. Un joueur est choisi pour contrôler le jeu et les trois joueurs restants se connectent au jeu par l'adresse IP du serveur central.

Fiche résumé du cinquième étalon

<i>Etalon Name</i> : L-Eurche Card Game System	<i>Abbreviation</i> : LECGS	<i>Date</i> : February 2005
Measured Software Overview	Software Information	
<p>The Euchre game system is called L-Euchre. The "L" stands for the LeTourneau version of the game. It is documented in the ISO technical report: ISO/IEC TR 14143-4 (Version 2000). This ISO document provides various sets of Functional User Requirements (FUR), described usually in a textual formal. The purpose of this ISO document is to provide to researchers and practitioners sets of requirements to be used as publicly available documents for measuring the functional size of software.</p> <p>The L-Euchre card game system used in this case study corresponds to set RUR B.11 of this ISO document. This set RUR B.11 documents are the requirements for a real-time system. The L-Euchre is a four-player euchre card game. The game remotely connects four players across the Internet through the TCP/IP protocol. The same game kernel must be running on each</p>	<p><i>FUR Origine</i> : ISO TR 14143-4: 2000 Set RUR B.11</p> <p><i>Application domain</i> : Games</p> <p><i>Application type</i> : Real Time</p>	
	Layers and Users Identification	
	<p><i>Layers Number</i> : 1</p> <p><i>Users</i> :</p> <ul style="list-style-type: none"> - Host Player - Players <p><i>Triggering events Number</i>: 14</p>	

<p>player's computer. One player chooses to host the game and the remaining three players connect to game by entering the host's IP address.</p> <p>After all three players have connected, teams are chosen by the host's selection. The host kernel proceeds to shuffle the cards and deal them out to each player. Bidding for trump and points begins on the first game with the host. Playing for tricks begins with the player virtually to the left or clockwise.</p> <p>The host kernel keeps track of the tricks won by each team and then determines the points won after each hand is played. The host kernel deals a new hand until a team wins.</p> <p>After the game is won/lost teams may be re-selected and a new game started, or the game is dropped. A chat window is provided in the program to allow real-time "table talk" between all players.</p>																																					
Measurement Summary	Etalon UML Diagrammes																																				
<table border="0"> <thead> <tr> <th data-bbox="300 1083 730 1119">Process Name</th> <th data-bbox="730 1083 922 1119">Size in Cfsu</th> </tr> </thead> <tbody> <tr> <td data-bbox="300 1119 730 1155">Host a Game</td> <td data-bbox="730 1119 922 1155">6</td> </tr> <tr> <td data-bbox="300 1155 730 1190">New Game</td> <td data-bbox="730 1155 922 1190">5</td> </tr> <tr> <td data-bbox="300 1190 730 1226">Choosing a Seat</td> <td data-bbox="730 1190 922 1226">4</td> </tr> <tr> <td data-bbox="300 1226 730 1262">Dealing of Cards</td> <td data-bbox="730 1226 922 1262">4</td> </tr> <tr> <td data-bbox="300 1262 730 1297">Choosing trump suit</td> <td data-bbox="730 1262 922 1297">3</td> </tr> <tr> <td data-bbox="300 1297 730 1333">Going Alone</td> <td data-bbox="730 1297 922 1333">5</td> </tr> <tr> <td data-bbox="300 1333 730 1369">Playing a Hand</td> <td data-bbox="730 1333 922 1369">4</td> </tr> <tr> <td data-bbox="300 1369 730 1404">Error message</td> <td data-bbox="730 1369 922 1404">3</td> </tr> <tr> <td data-bbox="300 1404 730 1440">Winning a Trick</td> <td data-bbox="730 1404 922 1440">3</td> </tr> <tr> <td data-bbox="300 1440 730 1476">Ending a hand</td> <td data-bbox="730 1440 922 1476">4</td> </tr> <tr> <td data-bbox="300 1476 730 1512">Scoring a Hand</td> <td data-bbox="730 1476 922 1512">3</td> </tr> <tr> <td data-bbox="300 1512 730 1547">Keeping Score</td> <td data-bbox="730 1512 922 1547">5</td> </tr> <tr> <td data-bbox="300 1547 730 1583">Win/Lose a Game</td> <td data-bbox="730 1547 922 1583">3</td> </tr> <tr> <td data-bbox="300 1583 730 1619">Lost IP Connection</td> <td data-bbox="730 1583 922 1619">5</td> </tr> <tr> <td data-bbox="300 1619 730 1654">Host a Game</td> <td data-bbox="730 1619 922 1654">4</td> </tr> <tr> <td data-bbox="300 1654 730 1690">Total Number of processes</td> <td data-bbox="730 1654 922 1690">Total Size in Cfsu</td> </tr> <tr> <td data-bbox="300 1690 730 1726">15</td> <td data-bbox="730 1690 922 1726">61</td> </tr> </tbody> </table>	Process Name	Size in Cfsu	Host a Game	6	New Game	5	Choosing a Seat	4	Dealing of Cards	4	Choosing trump suit	3	Going Alone	5	Playing a Hand	4	Error message	3	Winning a Trick	3	Ending a hand	4	Scoring a Hand	3	Keeping Score	5	Win/Lose a Game	3	Lost IP Connection	5	Host a Game	4	Total Number of processes	Total Size in Cfsu	15	61	<ol style="list-style-type: none"> 1. L-Euchre card game System - Use Case Diagram 2. L-Euchre card game - Host a Game Sequence Diagram 3. L-Euchre card game System – join a game Sequence Diagram 4. L-Euchre card game System – New Game Sequence Diagram 5. L-Euchre card game - Choosing a seat Sequence Diagram 6. L-Euchre card game System – Dealing of cards Sequence Diagram 7. L-Euchre card game System – Choosing trump suit Sequence Diagram 8. L-Euchre card game System –
Process Name	Size in Cfsu																																				
Host a Game	6																																				
New Game	5																																				
Choosing a Seat	4																																				
Dealing of Cards	4																																				
Choosing trump suit	3																																				
Going Alone	5																																				
Playing a Hand	4																																				
Error message	3																																				
Winning a Trick	3																																				
Ending a hand	4																																				
Scoring a Hand	3																																				
Keeping Score	5																																				
Win/Lose a Game	3																																				
Lost IP Connection	5																																				
Host a Game	4																																				
Total Number of processes	Total Size in Cfsu																																				
15	61																																				

	<p>Error message Sequence Diagram</p> <p>11. L-Euchre card game System – Winning a trick Sequence Diagram</p> <p>12. L-Euchre card game System – Ending a hand Sequence Diagram</p> <p>13. L-Euchre card game System – Scoring a hand Sequence Diagram</p> <p>14. L-Euchre card game System – Keeping a score Sequence Diagram</p> <p>15. L-Euchre card game System – Winning/losing a game Sequence Diagram</p> <p>16. L-Euchre card game System – Lost IP Connection Sequence Diagram</p> <p>Observations on requirements' clarity</p> <p>At the time of doing the measurement, Researchers and international experts didn't bring any observations.</p>
--	---

11.3.6 Sixième étalon

Le sixième étalon de mesure s'appelle *Rice Cooker System Etalon*. Les FUR du logiciel *Rice Cooker System* sont documentées dans son rapport de spécifications et d'exigences. Ces exigences et ces spécifications documentent les FUR d'un système de contrôle de cuiseur de riz.

Fiche résumé du sixième étalon

<i>Etalon Name : Rice Cooker System</i>	<i>Abbreviation</i> : <i>Date</i> : August RCS 2004
Measured Software Overview	Software Information
<p>The Rice Cooker must be able to cook rice in three Modes: fast, normal and gruel.</p> <p>The Rice Cooker starts cooking rice when the START button is pressed, normally after the operator selects a Mode.</p> <p>If the START button is pressed without the operator having first selected a Mode, the Rice Cooker automatically starts cooking in normal Mode.</p> <p>After the rice cooking is completed, the Rice Cooker automatically enters warming status (in all Modes).</p> <p>The appropriate indicator lamps must be lit during cooking and warming to inform the operator of the Rice Cooker's status.</p> <p>The heater must be controlled according to the 'target temperature/mode/elapsed time' data.</p> <p>The Cooking and Warming indicator lamps will be switched on or off dependent on the elapsed time since start and the selected Mode. For each Mode, there is a pre-defined elapsed time that is used as a reference to switch the indicator lamps. The Cook lamp shall be lit for a pre-defined elapsed time after the start of the cooking time. At all other times it shall be unlit. The Warm lamp shall be lit after a pre-defined elapsed time from the start of the cooking start time, and will remain lit while the Rice Cooker</p>	<p><i>FUR Origine</i> : Rice Cooker System</p> <p><i>Application domain</i> : Control</p> <p><i>Application type</i> : Real Time</p>
	Layers and Users Identification
	<p><i>Layers Number</i> : 1</p> <p><i>Users</i> :</p> <ul style="list-style-type: none"> - the Timer, - the Mode switch, - the Temperature sensor - Cooking and Warming indicator lamps, - the heater <p><i>Triggering events Number</i>: 5</p>

is powered. At all other times it shall be unlit.		
Measurement Summary		Etalon UML Diagrammes
Process Name	Size in	Rice Cooker System - Use Case
Cfsu		Control Heater - Sequence Diagram
Set Target Temperature	4	Control Indicator Lamps
Control Heater	4	Select Cooking Mode
Control Indicator Lamps	3	Set Target Tempuarture
Total Number of processes	Total Size in	Observations on requirements' clarity
Cfsu		
3	12	If the requirements had been to get, at the time of the failure, the data from the software rather than from the sensors, would this have had an impact on the size measurement?

Create a Schedule	8		CRS – Register for Courses (Delete Schedule) Sequence Diagram
Modify a Schedule	8		CRS – Register for Courses (Submit Schedule) Sequence
Delete a Schedule	5		CRS – Register for Courses (Update Schedule) Sequence
View a Schedule	4		CRS – Register for Courses (Save Schedule) Sequence Diagram.
Save a Schedule	5		
Add a student	4		
Modify a student	4		
Delete a Student	4		
View a Student	4		
Select Courses to Teach	8		
Submit Grades	6		
View Report Card	5		
Monitor for Course Full	4		
Total Number of processes		Total Size in	
Cfsu			
19	96		
			Observations on requirements' clarity
			The following ambiguities have been noted:
			There are several inconsistencies between Vision document and Use Cases, for instance:
			The Department responsible for a Course is mentioned in the Vision but not in the Use Cases;
			There is a requirement in the Vision to enable a student to enquire on Course Offerings by Department, which is not mentioned in the Use Cases;
			It is stated somewhere that all error messages must be logged, but this is ignored in the Use Cases. Similarly the Vision document refers to the need for Help functionality, but this is ignored;
			There are several enquiries mentioned in the Vision document, but no Use Cases for them, and the 'retrieve-before-update' enquiries are not shown as separate functional processes in the ETS document;
			The Vision says a Student can enquire on his score for attendance at a single Course Offering, as well as on his own Report Card (these sound like separate

	functional processes), but the former is not recognized in the Use Cases. In order to remove these ambiguities, the result measurement in this document is carried out from the Use Cases.
--	--

11.3.8 Huitième étalon

Le huitième étalon de mesure s'appelle *Collegiate Sports Paging System Etalon*. Les FUR du logiciel *Collegiate Sports Paging System* sont documentées dans le manuel de formation du processus RUP. Cet ensemble documente les FUR d'un système qui permet aux abonnés d'être avisés des événements collégiaux ou des événements des équipes de sports auxquels ils sont souscrits. Le système permet aussi aux abonnés de regarder le contenu de ces événements.

Fiche résumé du huitième étalon

Etalon Name : Collegiate Sports Paging System	Abbreviation : CSPS	Date : October 2005																											
Measured Software Overview	Software Information																												
The Collegiate Sports Paging System is documented in the Rational Unified Process (RUP Version 2003.06.00.65) document as an example of a fictional project to develop a system to page subscribers when specified collegiate sporting events take place. This system will allow subscribers to be notified of events relating to collegiate sports events or teams to which they subscribe, and will enable them to view the content they have subscribed to.	<i>FUR Origine</i> : IBM-Rational training documents <i>Application domain</i> : Web site <i>Application type</i> : MIS																												
	Layers and Users Identification																												
	<i>Layers Number</i> : 1 <i>Users</i> : -User (advertiser or subscriber) -Pager through pager gateway -User (advertiser or subscriber) -Existing Web Server <i>Triggering events Number</i> : 10																												
Measurement Summary	Etalon UML Diagrammes																												
<table border="0"> <tr> <td>Process Name</td> <td></td> <td>Size in</td> </tr> <tr> <td>Cfsu</td> <td></td> <td></td> </tr> <tr> <td>Approve Story</td> <td>6</td> <td></td> </tr> <tr> <td>Reject Content</td> <td>4</td> <td></td> </tr> <tr> <td>Modify Content</td> <td>7</td> <td></td> </tr> <tr> <td>Save Content</td> <td>5</td> <td></td> </tr> <tr> <td>View Content</td> <td>4</td> <td></td> </tr> <tr> <td>Approve Advertising</td> <td>5</td> <td>Content</td> </tr> <tr> <td>Reject Advertising Content</td> <td>4</td> <td></td> </tr> </table>	Process Name		Size in	Cfsu			Approve Story	6		Reject Content	4		Modify Content	7		Save Content	5		View Content	4		Approve Advertising	5	Content	Reject Advertising Content	4		Main Use Case Diagram Advertiser Use Case Diagram Current WebNewsOnLine System - Use Case Diagram Editor - Use Case Diagram Pager Gateway - Use Case Diagram Potential Subscriber - Use Case Diagram Subscriber – Use Case Diagram Approve Story Realization - Sequence	
Process Name		Size in																											
Cfsu																													
Approve Story	6																												
Reject Content	4																												
Modify Content	7																												
Save Content	5																												
View Content	4																												
Approve Advertising	5	Content																											
Reject Advertising Content	4																												

View Advertising Content	3	Diagram
Edit Profile	9	Edit Profile Realization - Sequence Diagram
Update Profile	4	Diagram
Pay Fee with Credit Card	7	Pay Fee with Credit Card Realization - Sequence Diagram
Subscriber renews subscription	5	Post Ad Content Realization - Sequence Diagram
Select Advertiser Reports	4	Diagram
Print Advertiser Reports	3	Print Ad Reports Realization - Sequence Diagram
Save Advertiser Reports	6	Diagram
Post Advertising Content	5	Read Content (on website) Realization - Sequence Diagram
Associate Content with ad	8	Send Content Realization - Sequence Diagram
Invalid Account Information	4	Diagram
Provide Feedback	3	Send Page Realization - Sequence Diagram
Email Option	5	Diagram
Read Content on Web site	7	Subscribe Realization - Sequence Diagram
User is not registered subscriber	4	Diagram
View a Story	3	Proportion of COSMIC FFP data movement types - Collegiate Sports Paging application.
Send Content	4	
Send Page	4	
Subscribe	6	
Update Subscriber Category	7	
Total Number of processes Cfsu	27	136
		Observations on requirements' clarity
		In the RUP documentation, there is no claim that their sets of documented requirements meet the quality criteria specified in IEEE 830.
		Both the Use Cases and the Sequence diagrams have been taken 'as in' from the RUP training section on RATIONAL web site.
		The measurement is based on the RUP documentation 'as is'.
		The following ambiguities have therefore been noted:
		In many use cases specifications there aren't clarifications about messages. To avoid any inconstancies between the Vision document and Uses Cases, the measurement result is carried out from

	Use Cases only.
--	-----------------

11.4 Affichage sur le web

Après avoir complété le développement de la première version des étalons, les étalons sont mis sur le site web (<http://www.gelog.etsmtl.ca/FSMetalons/>) avec leurs exigences, les cas d'utilisation et les diagrammes de séquence. Chaque membre du processus de vérification peut télécharger les documents et envoyer ses commentaires.

Un forum est également ajouté pour améliorer la qualité des étalons. Actuellement la participation des experts est en cours, cependant la période de leur participation dépasse la période du doctorat de l'étudiant mandataire. Des travaux ultérieurs à la période de doctorat sont envisagés pour parfaire le référentiel. Une hypothèse à considérer dans ce travail de recherche est que l'expertise des experts internationaux est correcte et exhaustive. Des discussions poussées sur les résultats de mesure avec les experts à travers des forums de discussion sont prévues pour éviter tout malentendu.

Le site web (figure 29 ci-dessous) se trouve à l'adresse suivante :

<http://www.gelog.etsmtl.ca/FSMetalons/>

http://www.gelog.etsmtl.ca/FSMetalons/ - Microsoft Internet Explorer

Fichier Edition Affichage Favoris Outils ?

Précédente Recherche Favoris

Adresse http://www.gelog.etsmtl.ca/FSMetalons/



Quebec University - Engineering School of High Technology (ETS)
Software Engineering Management Research Laboratory (GELOG)
Software Functional Size Measurement Standard Etalons - Project
Welcome to first drafts of Software Functional Size Measurement

Dear members;
 We have the pleasure to present to you our project about building Software Functional Size Measurement standard etalons with COSMIC-FFP (ISO 19761).
 A standard etalon is a material measure, measuring instrument, reference material or measuring system intended to define, realize, conserve or reproduce a unit or one or more values of a quantity to serve as a reference.
 Examples: 1 kg mass standard, 100 Ω standard resistor, standard ammeter, caesium frequency standard
 (VIM 1993: International Vocabulary of Basic and General Terms in Metrology, International Organization for Standardization, Geneva, 1993.)

The table below shows the verification level (VL) of the standard etalons in ascendant order are :

Verification Level	Rank
Adel Khelifi	A
Individual Expert	B
Charles Symons - Join Project Leader of COSMIC	C
COSMIC-Committee	D
ISO-WG12-TR	E
ISO International Standard	F

Work in progress - list of drafts for standard etalons.

No	DRAFT'S TITLE	REFERENCE DOCUMENTS	RELATED UML DIAGRAMS	MEASUREMENT DOCUMENTS	VL
1	Automatic Line Switching (ALS)	ISO 14143-4 - RUR B8	GELOG Version	Version 1	C
2	Gateway Application (SAGA)	ISO 14143-4 - RUR B10	GELOG Version	Version 1	B
3	Valve Control	ISO 14143-4 - RUR B9	GELOG Version	Version 1	C
4	Hotel Reservation System (HRS)	ISO 14143-4 - RUR A1	GELOG Version	Version 1	C
5	L-Euchre System	ISO 14143-4 - RUR B11	GELOG Version	Version 1	B
6	Rice Cooker	Rice Cooker Requirements	GELOG Version	Version 1	D
7	Course Registration System (CRS)	CRS-RUP	RUP Version	Version 1	C
8	Collegiate Sports Paging System (CSPS)	CSPS-RUP	RUP Version	Version 1	B

To see the on-going discussions, please go to the **FORUM**. © Copyright 2005. All rights reserved. Contact: [Adel Khelifi](mailto:Adel.Khelifi@etsmtl.ca)

Figure 29 Site web des étalons de mesure

Le forum se trouve à l'adresse suivante (figure 30):

http://www.gelog.etsmtl.ca/FSMetalons/disc1_welc.htm

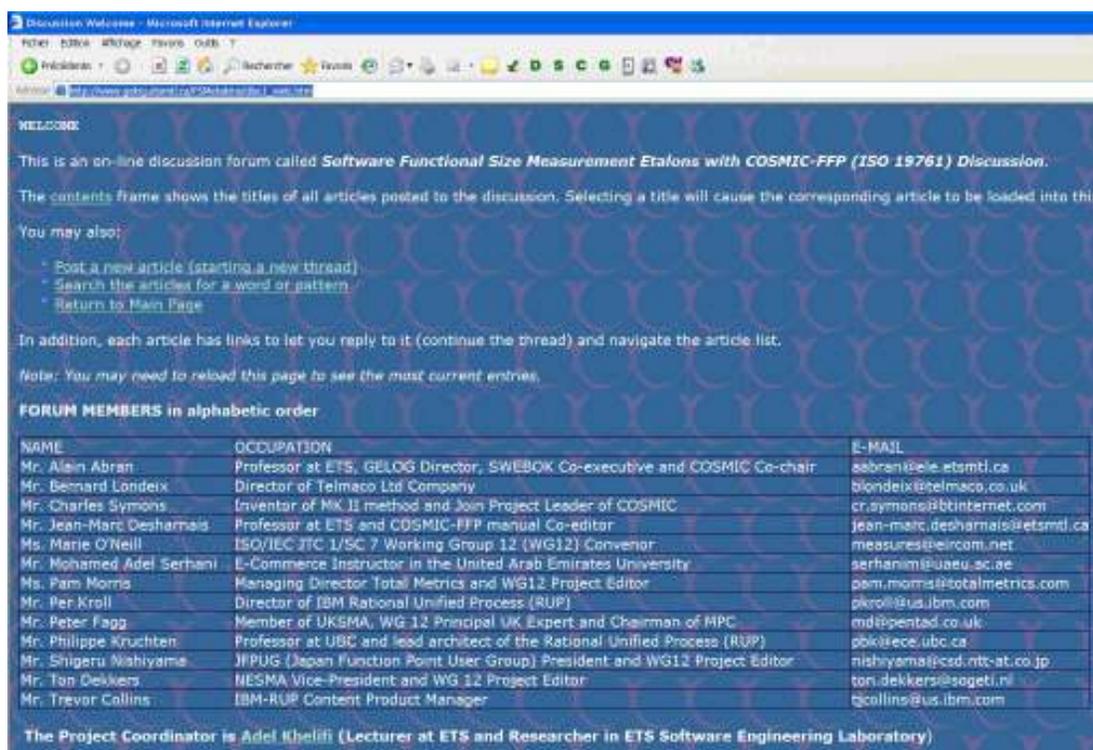


Figure 30 Forum de discussion sur les étalons de mesure

Tout cela a favorisé une bonne collaboration entre les différentes parties prenantes de ce projet. Les parties principales impliquées dans ce projet sont : le Laboratoire de Recherche en Génie Logiciel (GÉLOG) de l'ÉTS, des gestionnaires de produits de *Rational Unified Process*, des étudiants des cycles supérieurs en génie logiciel, des membres du groupe COSMIC et des membres du groupe de travail JTC1/SC7 WG12 d'ISO.

11.5 Évaluation des résultats

Question 1 : Comment aborder le design d'étalonnage en génie logiciel?

Dans cette thèse, une étude de la taille fonctionnelle des logiciels est faite, suivie par le choix de la méthode COSMIC-FFP pour le design du référentiel vu les avantages, cités

au quatrième chapitre, qu'a cette méthode de mesure par rapport à d'autres qui se trouvent sur le marché.

Les étalons conçus illustrent les caractéristiques d'un référentiel recommandées par ce travail, pour faciliter la mesure de la taille fonctionnelle des logiciels. Ces étalons sont produits en appliquant la méthode COSMIC-FFP qui est une norme internationale (ISO/IEC 19761, 2002) sur des fonctionnalités utilisateurs requises provenant de la norme ISO 14143-4 (2000) et des documents de la compagnie IBM-Rational, et vérifiés par des experts internationaux en mesure des logiciels qui ont une bonne connaissance de la norme ISO 14143-3 (2002) de vérification de la mesure de la taille fonctionnelle des logiciels.

Question 2 : Comment développer une première version d'étalons de la mesure de la taille fonctionnelle des logiciels qui puisse être utilisée comme référence?

Le cadre de réalisation des étalons est celui de la normalisation. En fait, l'utilisation des normes ISO pour l'application de la méthode de mesure (ISO 19761) sur des FUR d'ISO (ISO 14143) et du processus RUP ainsi que la vérification des résultats par des experts connaisseurs de la norme ISO 14143-3, donne à l'ensemble des étalons le caractère d'un référentiel.

11.6 Conclusion

Ce chapitre a présenté un référentiel pour la mesure de la taille fonctionnelle des logiciels. L'approche permet, pour la première fois en génie logiciel, de présenter des résultats de mesure des logiciels vérifiés par des experts internationaux et suffisamment précis pour être considérés comme une première génération des étalons de la MTF des logiciels. Ainsi, dans sa première version, le référentiel représente des références au début de leur cycle de vie, décrivant la mesure des logiciels par des rapports récapitulatifs qui montrent l'application de la MTF sur les FUR des logiciels. Cette

mesure est vérifiée selon une approche utilisant la comparaison avec les résultats de la même mesure réalisée par des experts du domaine.

Le référentiel élaboré dans le domaine du génie logiciel se place dans le mode international de pensée qui représente une expérience pertinente pour la MTF des logiciels. La méthode de vérification de ces étalons est présentée dans le prochain chapitre.

CHAPITRE 12

VÉRIFICATION DES ÉTALONS

Le but de ce chapitre est de fournir une méthode de vérification des étalons conçus. Au début, il montre l'absence de consensus ainsi que le besoin d'en avoir un pour la vérification de la mesure des logiciels. Puis il décrit les procédures de vérification adaptée dans cette thèse. Le chapitre évoque le choix et les rôles des experts dans le processus de vérification des étalons, il définit aussi le processus de vérification et les hypothèses prises. Il termine par la présentation de la conception de la vérification, son déroulement et son évaluation.

12.1 Introduction

La vérification consiste à s'assurer que l'on fait les choses correctement (*doing the things right* (B. Boehm, 1984)). Selon Industrie Canada (2004) « la vérification est l'examen et l'étude méthodique en vue de contrôler une mesure, une procédure, un procédé ou une activité ». Dans le cas de cette recherche, la mesure en vérification est la mesure de la taille fonctionnelle du logiciel. La vérification des étalons demande la consultation des experts en mesure des logiciels afin de nous assurer de la qualité des étalons. C'est ce qui devrait permettre de créer une première version des étalons de la MTF des logiciels.

L'objectif derrière ce processus de vérification, est d'obtenir, avec une certitude le moins biaisée possible, des résultats de mesure produits par des experts afin de les vérifier par rapport aux étalons. Pour ce faire, l'expert mesure, au choix, les FUR des logiciels et fait ses commentaires. Toutefois, il faut signaler que cette vérification dépend de l'intérêt des experts pour un étalon ou un autre et de leur disponibilité.

12.2 Absence de consensus

Le troisième chapitre a dressé un inventaire extrait du livre de Zuse *A Framework for Software Measurement* sur les propriétés des mesures logicielles. Cet inventaire montre bien qu'il n'y a pas de consensus sur les propriétés de la mesure des logiciels. Donc, on ne peut pas avoir pour le moment un consensus sur la vérification des propriétés de la mesure des logiciels.

D'un point de vue pratique, une vérification basée sur les propriétés des mesures serait très difficile car elle laisse au vérificateur la responsabilité de définir précisément les propriétés à mesurer. Par exemple, si on se réfère au rapport technique ISO 14143-3, il faudrait prouver que la mesure est répétable, reproductible, exacte, convertible, qu'elle ait un seuil de tolérance et qu'elle soit applicable aux domaines fonctionnels. Cette définition est trop rigoureuse sur le plan pratique pour une mesure dans le domaine du génie logiciel, ce qui la rend inappropriée dans le contexte de cette recherche. Au début, la recherche a abordé l'application de ces propriétés de mesure sur les étalons mais il a été observé que les outils, les méthodes et l'objectif de la vérification deviennent hétérogènes et ouvrent un grand éventail de procédures et d'activités peu concentré sur l'objectif principal de la thèse. D'ailleurs, après des discussions avec les experts impliqués dans cette recherche, il a été constaté qu'ils s'intéressent plus au contenu de l'étalon, à sa qualité et aux interprétations faites pour réaliser les étalons qu'à l'application des propriétés ci-dessus, malgré qu'elles proviennent d'un rapport technique ISO. La vérification de ces propriétés devient plus intéressante une fois que les étalons sont acceptés comme première génération des étalons de mesure logicielle.

12.3 Nécessité d'un consensus

L'inventaire sur les propriétés des logiciels dans le troisième chapitre n'est pas exhaustif mais montre bien la diversité de sens que peut revêtir le terme *propriété*. Il est important de se mettre d'accord dans la communauté scientifique et industrielle sur ce que signifie

une mesure vérifiée. Pour améliorer le crédit des mesures logicielles auprès des ingénieurs, il est important de construire une démarche de vérification sur des fondements solides. La métrologie est une base sérieuse qui devrait favoriser l'établissement d'un consensus parmi les experts (industriels et scientifiques). De plus, l'implication des experts internationaux dans le processus de vérification de la mesure des logiciels est de première importance. Vu l'absence de références, de standard et d'étalon dans le domaine de la mesure logicielle ainsi que le jeune âge du génie logiciel lui-même, l'implication des experts permet de déterminer des processus bien élaborés et de contribuer à la maturité du domaine.

12.4 Procédures de vérification

Pour tester la robustesse de la procédure de vérification, cette thèse se base d'abord sur la comparaison de nos résultats de mesure avec ceux des étudiants des cycles supérieurs en génie logiciel, puis avec les résultats de mesure produits par les experts internationaux. Une fois cette phase terminée, des discussions avec les experts du domaine sont engagées afin d'arriver à un consensus sur le résultat, le contenu et la qualité de la mesure. La vérification est basée sur les propriétés de la mesure des logiciels proposées dans le rapport technique ISO 14143-3 qui sont : Répétitivité, Reproductibilité, Exactitude, Convertibilité, Seuil de Tolérance et Applicabilité aux domaines fonctionnels.

12.4.1 Choix et rôles des experts

Dans cette recherche la vérification est limitée au contenu de l'étalon, à sa qualité et aux interprétations faites pour réaliser les étalons. Ce processus de vérification a besoin de plusieurs experts en mesure des logiciels. La sélection des experts internationaux est faite à partir de réseaux de contacts du laboratoire GÉLOG. Les contacts du GÉLOG sont développés à travers les relations avec le groupe de travail WG 12 d'ISO, le consortium COSMIC (*Common Software Measurement International Consortium*), un

groupe volontaire et informel d'experts internationaux en mesure des logiciels, et les nombreuses interventions, au plan national et international, des chercheurs du laboratoire en matière de MTF. L'Annexe 9 présente un résumé biographique de quelques experts. Le suivi avec ces partenaires se fait pendant la période de la recherche et va se poursuivre jusqu'à ce que les étalons de mesure soient vérifiés par l'organisation ISO. Ces experts ont une bonne connaissance théorique des règles de la méthode de mesure fonctionnelle ainsi qu'une solide expérience pratique de cette même mesure. Les procédures de mesurage sont répétées par des mesureurs distincts.

Dans ce travail, la vérification des étalons de mesure de la taille fonctionnelle des logiciels et leur analyse, relèvent de la compétence d'experts dans le domaine des MTF. Dans ce processus, les experts travaillent par consensus. Dans ce cadre, il n'est donc pas nécessaire de définir des procédures spécifiques pour les experts. Cette vérification s'appuie sur toute la connaissance que les experts ont de la méthode de mesure, des interprétations des exigences fonctionnelles du logiciel à mesurer, de leur influence sur le processus de mesurage et des propriétés des mesures logicielles (répétitivité, précision,...). La connaissance que partagent les experts sur la méthode de mesure est presque la même. Cette façon de travailler facilite la tâche de formalisation du processus de mesurage. Elle fournit un cadre de travail rigoureux et favorise l'établissement d'un consensus entre experts. Le travail des experts est d'appliquer la méthode COSMIC-FFP sur des exigences fonctionnelles des logiciels à mesurer, de déterminer le contenu de l'étalon, la taille du logiciel, l'exactitude des diagrammes UML et les interprétations faites pendant le processus de mesurage ainsi que la qualité générale de l'étalon.

12.4.2 Définition de la vérification

Objet de l'étude : la mesure de la taille fonctionnelle des logiciels est une mesure dérivée à partir de la quantité de fonctionnalités requises que le logiciel doit accomplir pour répondre aux besoins de l'utilisateur.

Objectif : L'objectif est de développer des étalons de mesure.

Perspective : La perspective adoptée est celle du chercheur.

Contexte : L'expérience a été réalisée sur huit FUR différentes.

Sélection du contexte : Le contexte de l'expérience est le Laboratoire de génie logiciel (GÉLOG) de l'ÉTS. L'expérience est réalisée en dehors du contexte industriel. L'expérience est spécifique puisque le sujet est le développement des étalons de la mesure des logiciels.

À notre connaissance, c'est la première fois que ce sujet est traité en génie logiciel. Il s'agit d'un problème à la fois méthodologique et pratique, à savoir l'absence de références et de standards pour la mesure des logiciels.

12.4.3 Hypothèses de vérification

L'hypothèse principale dans cette vérification est que les résultats de mesures (produits par les experts internationaux en mesure de logiciels, sélectionnés pour le processus de vérification de nos étalons) sont crédibles ou de bonne qualité. Leur opinion est considérée comme très pertinente dans le contexte de ce travail de recherche.

Du fait que ce travail traite de la conception des étalons de mesure de la taille fonctionnelle des logiciels, il est évident que les FUR proviennent d'abord de la communauté ISO et plus précisément de la métanorme qui définit les critères pour les méthodes de mesure de la taille fonctionnelle des logiciels, ISO 14143.

Aussi, pour avoir un nombre suffisant d'étalons, l'équipe de cette recherche a reçu l'autorisation de la compagnie IBM-Rational d'utiliser les FUR de deux logiciels de leur programme de formation éducative, et ce à l'échelle internationale (voir Annexe 10). L'hypothèse ici est que la qualité des FUR est acceptable pour pouvoir appliquer la

méthode COSMIC-FFP : il s'agit de documents qui ont été produits et vérifiés par des experts du domaine, et qui, de plus, sont utilisés à l'échelle internationale et du domaine public.

Le cas de *Rice Cooker* vient de compléter l'ensemble des FUR à mesurer ici. Le cas *Rice Cooker* est préparé par les experts mêmes qui ont développé la méthode COSMIC-FFP.

12.4.4 Conception de la vérification

Il a été décidé de présenter ce travail aux experts internationaux en mesure des logiciels. L'outil principal de communication est Internet. Plusieurs experts ont montré leur volonté à contribuer au processus de vérification des étalons.

Un site web est développé pour faciliter l'accès aux documents et aux discussions entre experts. Le nombre d'experts ayant participé varie de 2 à 9, selon l'étalon. Les experts sont spécialisés en mesure de logiciels, et en particulier avec la norme ISO 19761. Les FUR, et les résultats de mesures à vérifier, sont disponibles sur le site (<http://www.gelog.etsmtl.ca/FSMetalons/>).

Les activités exécutées pour mettre sur pied ce processus de vérification sont les suivantes :

- sélectionner un certain nombre d'experts sur la base de leur expérience de la mesure COSMIC-FFP, de la mesure fonctionnelle et de leur disponibilité;
- demander aux experts d'utiliser le référentiel qui est en ligne (<http://www.gelog.etsmtl.ca/FSMetalons/>) pour télécharger les FUR à mesurer, pour consulter nos étalons et leurs diagrammes UML et pour soumettre leurs commentaires. De plus, ils peuvent discuter à travers un forum développé à cet effet (http://www.gelog.etsmtl.ca/FSMetalons/disc1_welc.htm);
- demander aux experts de répondre à des questions ou d'en ajouter d'autres sur l'interprétation des FUR;
- demander aux experts s'ils sont d'accord avec les recommandations produites en rapport avec les étalons afin de faciliter le processus de mesurage;
- assurer le suivi auprès des experts;

- mettre le résultat par expert dans une nouvelle version;
- une fois qu'un accord est établi entre les différents partenaires sur le contenu et la qualité du résultat de la mesure, l'étalon est affiché sur le site web.

12.5 Déroulement de la vérification

Le référentiel est composé de trois types d'étalons.

Type A : étalons sans aucun historique antérieur de mesures.

Le tableau 35 suivant présente les étalons qu'on retrouve sous ce type.

Tableau XXXV

Étalons - Type A

Nom d'étalon	Nombre d'itérations	Nom de la personne	Contribution	Date
Automatic Line Switching	7	Étudiant mandataire	First Draft	2004-08-26
		1 ^{er} expert international	Revision	2004-08-29
		Étudiant mandataire	Revision	2004-09-01
		2 ^e expert international	Revision	2004-09-11
		2 ^e expert international	Revision	2005-03-11
Collegiate Sports Paging System	3	Étudiant mandataire	First Draft	2004-09-16
		Expert international	Revision	2005-05-16

Type B : étalons bâtis sur les mesures des mêmes logiciels réalisés par des membres du comité COSMIC.

Le tableau 36 suivant présente les étalons qu'on retrouve sous ce type.

Tableau XXXVI

Étalons - Type B

Nom d'étalon	Nombre d'itérations	Nom de la personne	Contribution	Date
Rice Cooker	2	1 ^{er} expert international, 2 ^{ème} expert international	First draft, issued to Cosmic-Core Team for review	1999-12-06
		1 ^{er} expert international, 2 ^{ème} expert international, 3 ^{ème} expert international, 4 ^{ème} expert international, 5 ^{ème} expert international and 6 ^{ème} expert international	Various changes to clarify specifications and updates to the measurement procedures and results	2000-01-11
		Expert international	Two questions and answers added on p17. Minor re-phrasing of Requirement 3.	2000-08-09
		1 ^{er} expert international, 2 ^{ème} expert international	Updates to include the Purpose and Scope step, and to Questions and Answers - Addition of Annex B	2002-03-13
		1 ^{er} expert international, 2 ^{ème} expert international, 3 ^{ème} expert international,	Updates to: - synchronize the vocabulary with ISO 19761:2003; - to clarify which functional requirements are allocated to the hardware, and which functional requirements are allocated to the software; - to segregate the requirements into 3 iterative prototypes	2003-01-02
		Étudiant mandataire	Adds UML Diagrams	2005-03-02

Tableau XXXVI (Suite)

Étalons - Type B

Nom d'étalon	Nombre d'itérations	Nom de la personne	Contribution	Date
Valve Control System	18		First draft, issued to Cosmic-Core Team for review	04/11/2000
		1 ^{er} expert international, 2 ^{ème} expert international,		2003-01-20
		Étudiant mandataire	Whole document revisited, and sections added including: Improvements to specifications, Use Case Diagram and Sequence Diagram	2004-08-16
		1 ^{er} expert international	Revision and reformatting + Addition of the ISO RUR at the beginning of the document	2004-08-16
		Étudiant mandataire	Revising Figures and other items	2004-08-21
		Secrétaire	Update of formatting style	2004-08-23
		3 ^{ème} expert international	Full revision of sections 1 to 3.7	2004-08-25
		Étudiant mandataire	Full revision of section 3.8 and on	2004-09-01
		3 ^{ème} expert international	Comments on revision	2004-09-06
		Étudiant mandataire	Implementation of comments	2004-09-07
		2 ^{ème} expert international	Comments on revision	2004-09-17
		Étudiant mandataire	Implementation of comments	2004-09-19
		2 ^{ème} expert international	Proposed editorial improvements	2005-02-02

Type C : les étalons sont comparés avec les mesures des étudiants de maîtrise faites sur les mêmes logiciels pendant les sessions d'hiver 2003 et 2004 dans le cours SYS861 *La mesure : concept clef de en ingénierie du logiciel*.

Le tableau 37 suivant présente les étalons qu'on retrouve sous ce type.

Tableau XXXVII

Étalons - Type C

Nom d'étalon	Nombre d'itérations	Nom de la personne	Contribution	Date
Hotel Accommodation System (Reservation)	6	Étudiant mandataire	First Draft	2005-02-02
		1 ^{er} Expert international	Revision	2005-03-12
		2 ^e expert international	Revision	2005-03-17
SAVAT Gateway (SAGA) System	2	Étudiant mandataire	First Draft	2005-02-06
L-Euchre card game system (minimal implementation)	3	Étudiant mandataire	First Draft	2005-02-14
		Expert international	Revision	2005-05-09
C-Registration System	5	Étudiant mandataire	First Draft	2004-08-26
		1 ^{er} expert international	Revision	2004-09-07
		Étudiant mandataire	Second Draft	2004-09-16
		2 ^e expert international	Revision	2005-05-04

Nos étalons passent par trois types de vérification :

1. Vérification individuelle;
2. Vérification par des experts internationaux en mesure de logiciels, incluant ISO 19761;
3. Vérification systématique par le comité COSMIC

Vérification individuelle

La figure 31 suivante illustre ce genre de vérification pour chaque type d'étalon.

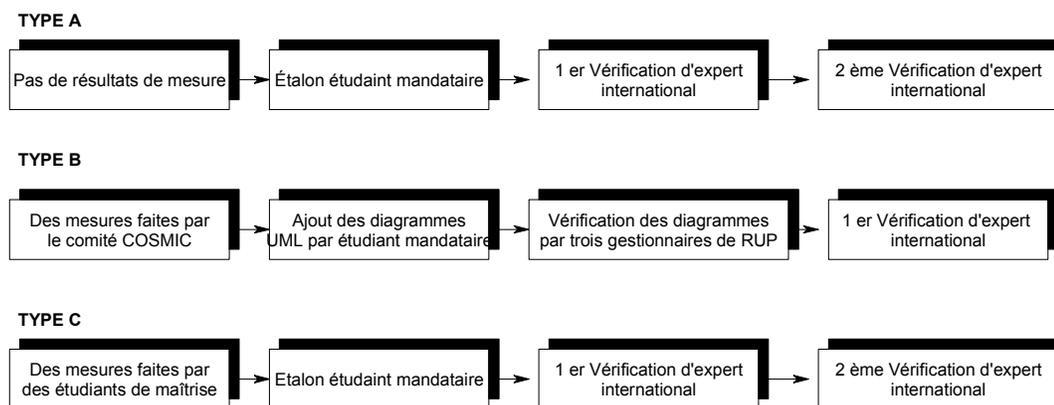


Figure 31 Vérification individuelle

Vérification par des experts internationaux en mesures logicielles

Un site web (figure 32) est développé pour permettre aux membres du processus de vérification de vérifier les versions intérimaires des étalons de mesures.

Le site contient 24 fichiers; huit documents d'exigences de logiciel, huit diagrammes UML et huit résultats de mesure. Chaque membre du processus de vérification peut consulter ces documents et mettre ses évaluations sur l'exactitude des étalons au forum de discussions. Les évaluations soumises au forum peuvent être commentées par les membres du processus de vérification.

La contribution des membres est laissée volontaire puisqu'ils ne sont pas obligés de fournir leur vérification pour chaque étalon de mesure.

http://www.gelog.etsmtl.ca/FSMetalons/ - Microsoft Internet Explorer

Quebec University - Engineering School of High Technology (ETS)
Software Engineering Management Research Laboratory (GELOG)
Software Functional Size Measurement Standard Etalons - Project
Welcome to first drafts of Sc

Dear members;
We have the pleasure to present to you our project about building Software Functional Size Measurement standard etalons with COSMIC-FFP (ISO 19761).
A standard etalon is a material measure, measuring instrument, reference material or measuring system intended to define, realize, conserve or reproduce a unit or one or more values of a quantity to serve as a reference.
Examples: 1 kg mass standard, 100 Ω standard resistor, standard ammeter, caesium frequency standard (VIM 1993: International Vocabulary of Basic and General Terms in Metrology, International Organization for Standardization, Geneva, 1993.)

The table below shows the verification level (VL) of the standard etalons in ascendant order are :

Verification Level	Rank
Adel Khelifi	A
Individual Expert	B
Charles Symons - Join Project Leader of COSMIC	C
COSMIC-Committee	D
ISO-WG12-TR	E
ISO International Standard	F

Work in progress - list of drafts for standard etalons.

No	DRAFT'S TITLE	REFERENCE DOCUMENTS	RELATED UML DIAGRAMS	MEASUREMENT DOCUMENTS	VL
1	Automatic Line Switching (ALS)	ISO 14143-4 - RUR B8	GELOG Version	Version 1	C
2	Gateway Application (SAGA)	ISO 14143-4 - RUR B10	GELOG Version	Version 1	B
3	Valve Control	ISO 14143-4 - RUR B9	GELOG Version	Version 1	C
4	Hotel Reservation System (HRS)	ISO 14143-4 - RUR A1	GELOG Version	Version 1	C
5	L-Euchre System	ISO 14143-4 - RUR B11	GELOG Version	Version 1	B
6	Rice Cooker	Rice Cooker Requirements	GELOG Version	Version 1	D
7	Course Registration System (CRS)	CRS-RUP	RUP Version	Version 1	C
8	Collegiate Sports Paging System (CSPS)	CSPS-RUP	RUP Version	Version 1	B

To see the on-going discussions, please go to the [FORUM](#). © Copyright 2005. All rights reserved. Contact: [Adel Khelifi](#)

Figure 32 Vérification des étalons par des experts à travers le site web

Processus de vérification systématique par le comité COSMIC

Une fois les processus précédents de vérification complétés, il est prévu de soumettre les propositions d'étalons à un processus de vérification systématique par le comité COSMIC. Ce dernier processus de vérification n'est toutefois pas sous le contrôle immédiat des chercheurs et la durée de cette vérification dépassera largement la période de doctorat de l'étudiant mandataire.

Pour tous les types de vérification, tous les étalons vont passer par un cycle d'amélioration. Ce cycle a, en entrée, l'ensemble des fonctionnalités utilisateurs requises (FUR), une fois cet ensemble passé par le processus de mesure, il produit une première version d'étalons. Le résultat de mesure des étalons est vérifié par des experts internationaux. Les diagrammes de cas d'utilisation et de séquence sont aussi vérifiés par des connaisseurs en méthodologie UML. À l'issue de cette première vérification, des discussions sont engagées pour améliorer la compréhension des exigences. Ce processus de vérification est itératif jusqu'à ce que les différents partenaires de cette recherche se contentent du contenu, de l'exactitude et de la qualité des étalons. La figure 33 suivante illustre ce cycle d'amélioration.

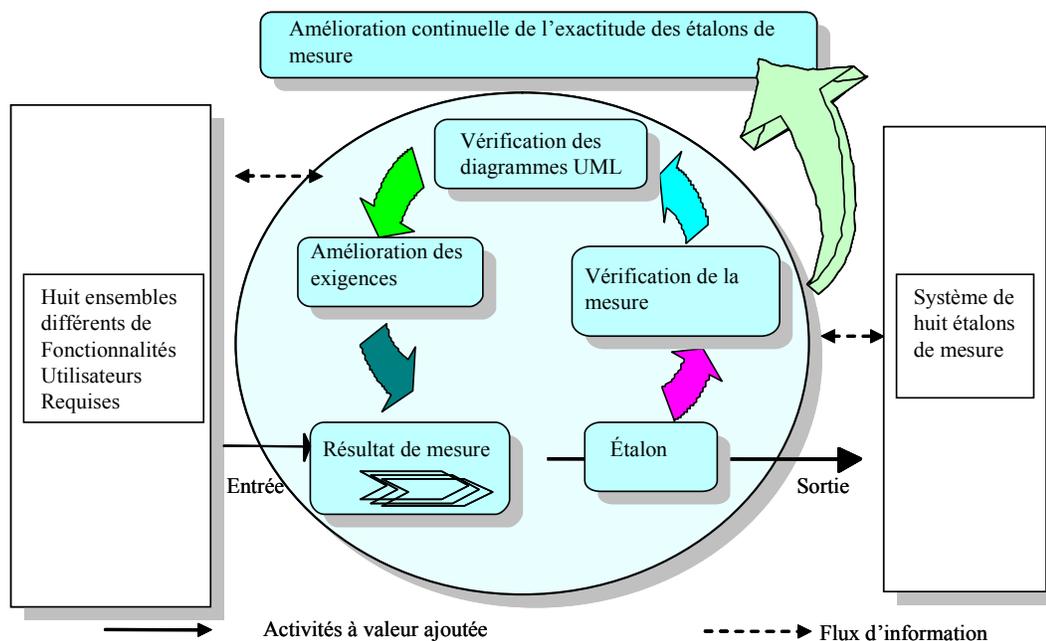


Figure 33 Cycle d'amélioration de la qualité des étalons

12.6 Portée de la vérification

Le processus de vérification proposé ici s'articule en sept étapes :

1. Définition de l'objectif de l'intervention des experts;
2. Choix des experts;
3. Développement des étalons de mesure;
4. Réception des résultats des mesures des experts;
5. Comparaison des résultats;
6. Discussion avec les experts internationaux sur la qualité des étalons;
7. Réalisation de la première version du référentiel.

La crédibilité des résultats pour les étalons se situe au niveau de la confiance que portent les utilisateurs des étalons sur l'expertise des experts qui ont contribué au processus de vérification. Comment savoir si la connaissance des experts reflète bien l'état de l'art dans la conception des étalons? Est-ce que leur savoir est pertinent et exhaustif? Ces interrogations sont aussi valables pour les utilisateurs des étalons.

Le résultat ne peut être généralisé au-delà de la portée du processus de vérification utilisé. Dans cette recherche, le nombre important d'interactions des experts internationaux assure une bonne vérification. Une vérification plus générale est à réaliser si ce travail vise une acceptation au niveau international, car le contexte de travail est académique et restreint à un nombre limité de FUR.

En métrologie, il est accepté que les étalons ne soient pas parfaits (surtout initialement) et que progressivement la communauté de la métrologie améliore la qualité des étalons. D'ailleurs, vu son importance, la préparation d'un étalon peut prendre tout le temps nécessaire afin d'être considérée comme référence de mesure. Par exemple, le mètre a pris presque deux cents ans pour se définir. Il a été conçu initialement en 1795 et redéfini finalement en 1983. En fait, de nos jours, les sciences expérimentales ne peuvent se développer et progresser qu'au prix d'évaluations quantitatives de plus en plus précises et poussées.

12.7 Sommaire

La vérification de la mesure des logiciels en est encore à ses débuts. Bien que cette étape soit primordiale pour assurer l'exactitude et la qualité d'une mesure, il n'y a pas de

consensus sur sa définition ni sur son utilisation dans l'arène de la mesure des logiciels. Ce chapitre a présenté la méthode de vérification des étalons conçus. La méthode de vérification utilisée repose principalement sur le consensus entre des experts internationaux en mesure des logiciels avec la norme ISO 19761 sur le contenu, le résultat de mesures et la qualité des étalons proposés.

Le prochain chapitre conclut cette thèse par les apports d'un tel travail en domaine du génie logiciel et par les travaux futurs qu'on peut amorcer à partir de ce travail de recherche.

CHAPITRE 13

CONCLUSION GÉNÉRALE

13.1 Introduction

Le présent projet de recherche a permis de concevoir un référentiel pour la mesure de la taille fonctionnelle des logiciels afin de contribuer à résoudre la problématique de manque de références pour la mesure des logiciels. Ce référentiel est composé de huit résultats de mesure, résultats appelés dans cette recherche « étalons de mesure ». Chaque étalon est réalisé par l'application d'une méthode spécifique pour la mesure des logiciels, soit la norme de mesure COSMIC-FFP (ISO 19761), sur des ensembles de Fonctionnalités Utilisateurs Requises (FUR).

La méthode COSMIC-FFP a été choisie dans cette recherche comme méthode de mesure pour le design du référentiel parce que cette méthode couvre plusieurs types de logiciels, qu'elle est plus facile à utiliser, qu'elle a été conçue selon la norme de mesure de la taille fonctionnelle des logiciels ISO 14143, qu'elle est compatible avec les méthodes modernes du génie logiciel telle que UML et qu'elle a été développée par des experts internationaux en mesure des logiciels et réalisée par le Laboratoire de recherche en génie logiciel de l'École de technologie supérieure (GÉLOG). C'est l'équipe de ce laboratoire qui a réalisé aussi ce référentiel. La sélection des ensembles de FUR a été faite à partir des FUR disponibles à ISO et à la compagnie IBM-Rational pour donner à ce travail un aspect de standardisation.

En effet, l'importance croissante des mesures en génie logiciel s'étend à présent à plusieurs attributs de logiciel tels que la taille, la qualité, l'utilisabilité, etc.

Du fait qu'« Une science a l'âge de ses instruments de mesure » (Louis Pasteur), le génie logiciel doit être doté d'instruments de mesure.

C'est dans la perspective de cette évolution que s'inscrit cette recherche dont l'objectif était de montrer, dans un premier temps, que la mesure en génie logiciel a besoin des standards et des références et, dans un deuxième temps, de proposer un exemple d'un référentiel pour la mesure des logiciels.

13.2 Ce qui a été fait

Pour ce faire, il a fallu tout d'abord mettre en évidence la place de la mesure dans le développement du logiciel et ses classifications proposées par plusieurs chercheurs du domaine tel que Fenton (1991) (Chapitre 1).

Puis, un inventaire a été dressé à partir des propriétés désirables des mesures des logiciels proposées par plusieurs chercheurs du domaine et exposé par Zuse (1998) dans son livre *A Framework for Software Measurement*, et des propriétés de mesure des logiciels recommandées par la norme ISO 14143-3. Cet inventaire a aidé à la sélection des propriétés de mesures des logiciels, comme la mesure de la taille fonctionnelle des logiciels (Chapitre 2).

Une vue d'ensemble est ensuite présentée sur la métrologie, sur le concept de standardisation, sur le contenu d'une norme et sur le rôle des normes et les types de normes. Cette revue de littérature a permis d'avoir une idée sur la conception des étalons de mesure (Chapitre 3).

Après une étude des principales méthodes de mesures existantes sur le marché, la méthode COSMIC-FFP (ISO/IEC 19761, 2002) a été choisie pour les avantages présentés en quatrième chapitre. De plus, cette méthode a été développée en grand partie par le Laboratoire de génie logiciel (GÉLOG) : par conséquent, le GÉLOG avait l'expertise nécessaire pour contribuer au développement des étalons de mesure avec la méthode COSMIC-FFP.

Une définition du terme étalon s'est avérée importante, par référence aux étalons universels et spécifiques à l'étalonnage, à la calibration et à la traçabilité des étalons (Chapitre 5).

Trois sous-problématiques ont été identifiées dans cette recherche : a) le logiciel est un produit difficile à mesurer; b) la mesure des logiciels n'a ni référence, ni standard ; et c) la documentation des logiciels est souvent incomplète ce qui rend la tâche de la mesure plus ambiguë. Ici, la problématique de recherche principale traite de manque de références en mesure des logiciels. La revue de littérature faite dans cette thèse a permis de cerner diverses facettes de cette problématique. En effet, il a été constaté que l'émergence de l'utilisation des concepts de métrologie classique en génie logiciel (A. Abran *et al.*, 2003; A. Khelifi *et al.*, 2004) est une bonne piste pour attaquer ce problème. Du fait que les mesures dans les domaines d'ingénierie classiques tels que civil, mécanique ou électrique se réfèrent toujours à des normes et à des standards dans leur conception et dans leur application, l'idée d'étudier les normes qui définissent la mesure des logiciels est importante. Mais le problème de choix de méthode de mesure s'est rapidement posé. En d'autres termes, devant l'existence de plus que 1 500 mesures de logiciels (Horst Zuse, 1998), il fallait en choisir une seule pour produire les étalons (Chapitre 6).

Afin de réaliser cet objectif, une méthodologie exploratoire et qualitative a été adoptée. Dans la méthodologie, la définition du projet, la motivation de l'équipe de recherche pour le sujet abordé, les utilisateurs du référentiel des MTF, les outils nécessaires à sa réalisation et ses limites ont été identifiés. Les principales étapes de la méthodologie conçue pour aborder cette problématique sont : la détermination de la méthode de mesure pour concevoir le référentiel, la réalisation de l'étalon, l'ajout des diagrammes UML, la conception du gabarit d'étalon et la sélection des experts pour vérifier la qualité des étalons. Pour afficher le contenu du référentiel à ses partenaires, un site web a été développé à cet effet.

Le terme étalon est intimement lié à la normalisation. Les outils utilisés dans cette recherche sont essentiellement des standards ISO. Les partenaires dans ce travail de recherche ont été des professeurs spécialisés en mesure des logiciels des universités locales et étrangères, des étudiants des cycles supérieurs et des experts nationaux et internationaux en mesure des logiciels (Chapitre 7).

Cela a permis d'obtenir des résultats intéressants (Chapitres 8 et 9), soumis à des discussions avec des experts afin d'en montrer leur pertinence (Chapitre 11). L'explication de la vérification des résultats a donné à ce travail plus de précision (Chapitre 11). À présent, la conclusion permet de revenir sur les conditions de réalisation de ce travail, les difficultés rencontrées, les résultats obtenus, les implications de cette recherche dans la pratique ainsi que les recherches à venir. Dans cette perspective, il est essentiel de décrire ce qui a été appris au terme de cette démarche.

13.3 Ce qui a été appris

Pour mieux décrire les différents apprentissages réalisés au cours de cette recherche, voici brièvement les deux résultats les plus importants :

1) La mesure des logiciels exige des améliorations :

La revue de littérature faite dans cette thèse a relevé que la mesure en génie logiciel a été traitée par plusieurs auteurs (A. Abran *et al.*, 2003; B. Kitchenham *et al.*, 1993; C. Symons, 1998; D. Kafura *et al.*, 1985; Horst Zuse, 1998, 1999; J. K. Kearney *et al.*, 1986; J. Navlaka, 1986; J. Pfanzgal, 1968; J.P. Jacquet *et al.*, 1998; M. Shepperd, 1992; N. Fenton *et al.*, 1997; N. Schneidewind, 1992; P. Suppes *et al.*, 1963; R. Watts, 1987; R.J. Rubey *et al.*, 1968; S. Wolff, 1999; S.L. Pfleeger, 1997; T. McCabe, 1976). Ces auteurs suggèrent entre autres que ce concept soit étudié selon plusieurs perspectives afin de les améliorer et de les rendre plus matures en tant que mesure.

Le travail de recherche décrit dans cette thèse apporte une pierre supplémentaire pour soutenir les hypothèses déjà émises par de nombreux auteurs. En effet, la majorité de ces chercheurs suggèrent que la mesure des logiciels nécessite des améliorations pour contribuer à réduire le fossé entre le concept de mesure dans les génies traditionnels tels que chimique, civil ou mécanique et dans le génie logiciel.

2) Le développement des standards et des références pour la mesure des logiciels est une nécessité pour hisser le génie logiciel au rang des génies traditionnels :

À notre connaissance, en dehors des travaux d'Abran *et al.* (2000; 2002; 2003) et de Khelifi *et al.* (2004) sur l'introduction des concepts métrologiques dans le design et dans l'utilisation des logiciels, cette recherche est l'une des rares à avoir introduit la conception des étalons de mesure en génie logiciel.

Cette recherche a présenté le développement d'un référentiel pour la mesure des logiciels, référentiel bâti sur une norme ISO (COSMIC-FFP – ISO 19761) et avec des rapports techniques ISO; de ce fait, les résultats de recherche obtenus s'appuient sur des références crédibles pour les mesureurs de logiciels.

13.4 Les contributions de la recherche

La réflexion menée sur la contribution d'un référentiel pour la mesure des logiciels a débouché sur le rappel de contributions théoriques importantes en mesure. La principale de ces contributions théoriques est que le développement des étalons de mesure est critique pour fournir des références, et même pour le domaine du génie logiciel.

Pour des raisons pratiques, la partie pratique de ce travail de recherche a été limitée à une seule méthode de MTF; ceci constitue néanmoins une avancée importante pour les mesureurs et les chercheurs en mesure des logiciels. En effet, les recherches récentes d'Abran *et al.* (2000; 2002; 2003) et de Khelifi *et al.* (2004) mettent en relief l'utilisation

des concepts de métrologie classique dans le design et dans l'application de la mesure des logiciels. Cette recherche va dans ce sens et introduit la notion d'étalonnage pour la mesure des logiciels. L'étalonnage est classiquement effectué avec des sources ponctuelles. Les sources ponctuelles de référence sont peu nombreuses en génie logiciel.

Cette thèse a amorcé une première initiative dans la réalisation et dans l'application de la mesure des logiciels par la préparation d'une première version d'un ensemble de références. En outre, le choix de la méthode COSMIC-FFP pour le design des étalons a permis d'avoir des étalons qui ont permis d'évaluer indirectement certains aspects de la qualité des artefacts disponibles pour faire la mesure, telle que la documentation,.

Ce travail a aussi souligné l'importance de la métrologie des logiciels qui est l'application directe de la théorie de mesure fondée par Krantz *et al.* (1971). Pour plusieurs chercheurs (N. Fenton et S.L. Pfleeger, 1997), cette théorie veut dire une assignation numérique. En génie logiciel, l'utilisation des valeurs numériques, pour l'évaluation des attributs d'un logiciel, n'est pas toujours adéquate car cette vision ignore de nombreux concepts métrologiques tels que l'utilisation des instruments de mesures, des standards et des références.

13.5 Conséquences des résultats

Dans la pratique, les résultats de cette recherche peuvent être utiles à la fois pour les gestionnaires de logiciels, pour les chercheurs en mesure de logiciels et pour les mesureurs de logiciels. Plus encore, les résultats obtenus contribueront éventuellement à répondre à un besoin spécifique de la communauté de mesure de la taille fonctionnelle des logiciels, soit la certification des mesureurs des MTF.

13.5.1 Utilité de ce travail pour la certification des mesureurs MTF

En novembre 2004, le groupe COSMIC (*Common Software Measurement International Consortium*) a établi un groupe d'étude pour aborder la problématique de la certification

avec la méthode de mesure COSMIC-FFP (*Certification Study Group*, 2005). Le groupe d'étude doit préparer des recommandations pour le groupe COSMIC. Ce dernier prévoit le besoin de certification des mesureurs qui utilisent la méthode COSMIC-FFP.

13.5.2 Les principes fondamentaux de certification

Les principes décrits dans les paragraphes suivants sont un résumé du texte de la norme ISO/IEC 17024 (2003). La caractéristique primaire d'un groupe de certification est de bien définir la politique et les procédures pour garantir, maintenir, renouveler, élargir et réduire la portée de la certification désirée et pour suspendre ou annuler la certification. Voici d'autres caractéristiques essentielles pour un groupe de certification :

- Un groupe de certification doit définir, développer et maintenir la méthode et les mécanismes à utiliser pour évaluer la compétence du candidat;
- Un groupe de certification doit établir, implanter et maintenir un système de gestion qui assure l'application appropriée des normes. Il doit maintenir des enregistrements appropriés afin d'assurer l'intégrité du processus de certification. La sécurité et la confidentialité doivent être assurées dans tout le processus;
- Un groupe de certification doit établir les procédures et les conditions pour la maintenance de certification. Il doit définir un processus de surveillance pour contrôler la conformité des détenteurs du certificat avec ces procédures et ces conditions;
- Un groupe de certification doit définir des exigences de re-certification et établir des procédures et des conditions pour renforcer ces exigences;
- Si un groupe de certification fournit une marque ou un logo de certification, il doit documenter les conditions d'utilisation et gérer convenablement les droits d'usage.

Des directives de certification des mesureurs des logiciels

Ces directives devraient être considérées comme un modèle de référence pour l'examen basé sur la certification du mesureur de logiciels.

Le groupe de certification fera un travail et analysera des tâches avec les buts suivants :

- Identifier les tâches importantes et critiques et les activités de travail à faire par les mesureurs compétents de logiciels. Cet effort montrera la place de la prise de mesure dans le cycle de vie, incluant la phase des exigences.
- Déterminer une spécification pour l'examen :
 - la liste des sujets à tester ;
 - le nombre de questions et leur distribution dans les sujets;
 - les buts des niveaux cognitifs des questions ;
 - le score de passage pour avoir la certification;
 - les spécifications de test doivent se référer à la norme appropriée incluant ISO/IEC 19761;
 - l'analyse du travail et de la tâche devrait être culturellement appropriée.
- Le groupe de certification devra créer et approuver un examen culturellement approprié qui se conforme à la spécification. L'examen peut inclure :
 - des items à choix multiples,
 - des études de cas,
 - de courts essais,
 - aussi bien que d'autres formats de question.

Utilisant ce modèle de référence, le groupe de certification peut travailler avec d'autres groupes analogues dans d'autres pays pour développer des mécanismes pour la reconnaissance mutuelle des mesureurs de logiciels utilisant la norme ISO 19761 (COSMIC-FFP), ceci pour assurer la portabilité d'une telle certification à travers les

frontières nationales. Cet effort devra tenir compte des critères d'opération des canevas de certification dans diverses nations.

Le référentiel de mesure de la taille fonctionnelle des logiciels développé dans cette recherche constitue un ensemble des études de cas intéressantes pour le processus de certification des mesureurs de logiciels utilisant la méthode COSMIC-FFP. En effet, les étalons de mesure réalisés dans cette thèse constituent un intrant intéressant pour l'examen des professionnels en mesure de la taille fonctionnelle des logiciels.

13.6 Les limites et les difficultés rencontrées

Cette recherche se limite au contexte académique. Malgré que le design du référentiel pour les MTF ait été fait avec la contribution de plusieurs experts internationaux du milieu industriel, les étalons ont été développés dans le Laboratoire de génie logiciel (GÉLOG) de l'ÉTS.

Étant donné les difficultés rencontrées et la démarche méthodologique suivie, il nous apparaît primordial qu'une recherche à plus grande échelle avec les mêmes variables soit entreprise dans le même contexte et, si possible, sur d'autres types de logiciel, afin d'enrichir l'éventail des connaissances sur la standardisation de la mesure des logiciels.

Étant donné que la recherche indique que les étalons développés constituent un bon référentiel pour les MTF, l'objectif des recherches à venir devrait viser à approfondir cette proposition. Des recherches qualitatives plus approfondies pourraient être menées dans la lignée des théories de mesure, pour tenter d'instaurer les pratiques de la métrologie classique dans le domaine du génie logiciel. Ces recherches pourraient permettre d'identifier et de comprendre les concepts les plus pertinents à retenir en termes de normalisation du processus de mesurage des logiciels.

Enfin, d'autres recherches pourraient porter sur l'impact du développement des étalons de mesure qui peuvent couvrir plus qu'un attribut de logiciel.

13.7 Un dernier mot

Ce travail pourrait ouvrir des horizons pour contribuer à la maturité du génie logiciel. Le référentiel constitue un cadre professionnel pour l'interprétation des mesures et pareillement pour un langage synthétique pour les mesureurs. L'ensemble des étalons peut être identifié comme étant un outil essentiel pour le développement de la mesure de la taille fonctionnelle des logiciels.

Le souhait principal est que ce référentiel contribue éventuellement au développement d'une norme et puisse s'étendre à un niveau international

L'impact des étalons de mesure de la taille fonctionnelle des logiciels peut être assez important. En effet, les organisations qui pratiquent la mesure des logiciels peuvent utiliser ces étalons pour faciliter la mesure de leurs produits logiciels, standardiser le processus de mesurage et évaluer la qualité de la documentation des logiciels à mesurer. Le design d'un référentiel pour la mesure de la taille fonctionnelle des logiciels pourrait servir comme input pour les travaux des comités ISO en génie logiciel. La réalisation des étalons permettra éventuellement de standardiser, en génie logiciel, l'utilisation des mesures. La suggestion d'amélioration des normes internationales aidera dans la maturité du domaine et amènera le domaine à un niveau de qualité meilleur.

RÉFÉRENCES

- A. Abran. (1994). *Analyse du processus de mesure des points de fonction*. Non publiée Thèse de doctorat, Ecole Polytechnique de Montréal, Montréal.
- A. Abran. (2000). *Analyzing and Improving Software Measurement Instruments based on Metrology and Engineering Concepts*. Montréal: Université du Québec à Montréal.
- A. Abran, et A. Sellami. (2002). *Initial Modeling of the Measurement Concepts in the ISO Vocabulary of Terms in Metrology*. Présenté au IWSM.
- A. Abran, Desharnais, J.-M., Oigny, S., St-Pierre, D., et Symons, C. (2003). *COSMIC FFP - Manuel de mesures - Version 2.2*. Montréal: *The Common Software Measurement International Consortium (COSMIC)*.
- A. Abran, J.M. Desharnais, S. Oigny, D. St-pierre, et C. Symons. (2002). *COSMIC-FFP Measurement Manual 2.2*. Montréal: UQAM.
- A. Abran, et J.P. Jacquet. (1999). *A Structured Analysis of the New ISO Standard on Functional Size Measurement-Definition of Concepts*. *Fourth IEEE International Symposium and Forum on Software Engineering Standards*, 230-241.
- A. Abran, Lucie Laframboise, et Bourque, P. (1999). *A Risk Assessment Method and Grid for Software Measurement Programs* (Rapport technique no 99-03). Montréal: Laboratoire de recherche en gestion des logiciels, Université du Québec à Montréal.
- A. Abran, Sellami A., et W., S. (2003). *Metrology, measurement and metrics in software engineering*. *IEEE Software Metrics Symposium, 2003. Proceedings. Ninth International*, 2-11.
- A. Carty. (2000). *La métrologie -- Un pilier invisible du marché mondial*. In C. n. d. l. d. métrologie (Ed.). Hôtel Westin Harbour Castle Toronto (Ontario).
- A. Idri. (2003). *Un modèle intelligent d'estimation des coûts de développement de logiciels*. Non publiée doctorat, Université du Québec à Montréal, Montréal.
- A. Khelifi, A. Abran, et L. Buglione. (2004). *A System of References for Software Measurements with ISO 19761 (COSMIC-FFP)*. Présenté au 14th International Workshop on Software Measurement (IWSM) IWSM-Metrikon 2004, Konigs Wusterhausen , Magdeburg, Germany.

- A.E. Giles, Daich, et T. Gregory. (1995, février 1995). Metric Tools. *CROSSTALK*, Vol. 8.
- A.J. Albrecht. (1979, octobre 1979). *Measuring Application Development Productivity*. Présenté au *IBM Application Development Symposium*, Monterey, CA.
- AFNOR. (1994). *Vocabulaire international des termes fondamentaux et généraux de métrologie*. Paris: AFNOR.
- Antoine, et Mathieu. (2003). *Les étalons de mesure*, 2005, tiré de <http://www.csdm.qc.ca/fseguin/scienceplus/mai2003/etalons.htm>
- B. Boehm. (1984). *Verifying and validating software requirements and design specifications*. *IEEE Software*, 1(1).
- B. Kitchenham, et K. Kansala. (1993, mai 1993). *Inter-item Correlations among Function Points*. Présenté au *Proceedings IEEE-CS International Software Metrics Symposium*, Baltimore.
- Banker, R., S. Datar, C. Kemerer, et D. Zweig. (2002). Software Errors and Software Maintenance Management. *Information Technology and Management*, 3(1/2), 25-41.
- BIPM, B. i. d. p. e. m.-. (2005a). *Le système international d'unités (SI)*, 2005, tiré de <http://www.bipm.fr/fr/si/>
- BIPM, B. i. d. p. e. m.-. (2005b). *Qu' est-ce que la métrologie ?*, 2005, tiré de <http://www1.bipm.org/fr/bipm/metrology/>
- Bureau de communications. (2004). *Traçabilité*, 2005, tiré de http://inms-ienm.nrc-cnrc.gc.ca/calibration_services_f.html
- Bureau national de la métrologie. (2003). *Histoire de la mesure*. Site visité le 6 Juin 2003, 2003, tiré de <http://www.bnm.fr/mesure-metrologie/me.histoire1.htm>
- C. Jones. (1978). *Measuring Programming Quality and Productivity*. *IBM Systems Journal*, 17.
- C. Jones. (1994). *Assessment and Control of Software Risks*: Yourdon Press.
- C. Jones. (1996). *Applied Software Measurement : Assuring Productivity and Quality* (2e ed.). New York: McGraw Hill.
- C. Symons. (1988). *Function Points Analysis: Difficulties and Improvements*. *IEEE Transactions on Software Engineering*, SE-14(1), 2-11.

- C. Symons. (1998). *MK II Function Point Analysis Counting Practices Manual Version 1.3.1* (Manuel). London: United Kingdom Software Metrics Associations (UKSMA).
- C. Symons. (1999). *Conversion between IFPUG 4.0 and MKII Function Points Version 3.0* (Note). London: Software Measurement Services Ltd.
- C.F. Kemerer. (1987). An Empirical Validation of Software Cost Estimation Model. *Communications of the ACM*, 30(5), 416-429.
- C.F. Kemerer. (1993). Reliability of function Points Measurement, A field experiment. *Communication of the ACM*, 36(2).
- Certification Study Group. (2005). *International Certification of measurers using ISO 19761: 2003 COSMIC Measurement Method* (Rapport de groupe d'étude). Berlin: Groupe COSMIC.
- CPM 2.0, N. (1997). *Definitions and counting guidelines for the application of function point analysis* (2 ed.).
- CPM 4.1, I. (1999). *Function Point Counting Practices Manual Release 4.1*.
- D. Baize, et M.C. Girard. (1995). *Référentiel pédologique* (INRA ed.). Paris: Association française d'étude des sols.
- D. Kafura, et J. Canning. (1985). *A Validation of Software Metrics Using Many Metrics and Two Resources*. Présenté à la 8th Intl. Conf. on Engineering, Washington, D. C.
- D. St-Pierre, M. Maya, A. Abran, J.M. Desharnais, et P. Bourque. (1997). *Full Function Points: Counting Practices Manual* (Rapport technique No. 04). Montréal, Canada.: Université du Québec à Montréal.
- D.H. Krantz, R. D. Luce, P. Suppes, et Tversky., A. (1971). Foundations of Measurement: Additive and Polynomial Representations. *Academic Press*, 1.
- D.J. Reifer. (1991, avril 1991). *Real-time Function Point Extensions*. Présenté à la IFPUG Spring Conference, Baltimore, Maryland.
- Desharnais, J.-M., Abran, A., Maya, M., et St-Pierre, D. (1998). *Mesure de la taille fonctionnelle des logiciels temps réel* (Technical Report No. 12). Montreal: Université du Québec a Montréal.
- E.J. Weyuker. (1988). Evaluating Software Complexity Measures. *14*(9).

Encyclopédie libre Wikipedia. (2005). *Métrologie*, 2005, tiré de <http://encyclopedie.snyke.com/articles/metrologie.html>

EOS. (2001). *Ancient Egyptian Standardization*, 2005, tiré de http://www.eos.org.eg/web_en/intro/en_story.html

Fenton, N., et Neil, M. (1999). *New directions in software metrics.*, 2005, tiré de http://www.dcs.qmw.ac.uk/~norman/papers/new_directions_metrics/HelpFileHistory_of_software_metrics_as_a.htm

G. Chartron. (2000). *Standards, normes, documents numériques.*, 2005

Galea, S. (1995). The Boeing Company: 3 D Function Point Extension. 2.0(1.0).

GELOG. (2003). *Rice Cooker application*, 2004, tiré de <http://www.lrgl.uqam.ca/cosmic-ffp/casestudies/>

GELOG. (2004). *Valve Control System Case Study*. Montreal: Ecole de Technologie Supérieure.

Grady Booch, James Rumbaugh, et Ivar Jacobson. (2000). *Le guide de l'utilisateur UML*.

H. Preben, et R. Fiona. (2003). *Metrology – in short* (2 ed.). Lyngby, Denmark: EUROMET.

Horst Zuse. (1998). *A Framework for Software Measurement*. Berlin: DeGruyter Publisher.

Horst Zuse. (1999). *Validation of measures and prediction models*. Présenté au International Workshop on Software Measurement.

I. Douglas, H. Mary, et U. Joan. (1999). *Mars climate orbiter team finds likely cause of loss*, 2005, tiré de <http://mars.jpl.nasa.gov/msp98/news/mco990930.html>

IBM. (2005). *Rational Unified Process*, 2005, tiré de <http://www-306.ibm.com/software/awdtools/rup/>

IEEE Computer Society. (1993). *IEEE Standard for a software Quality Metrics Methodology*. Piscawaty, NJ.

IEEE Std 610.12-1990. (1990). *IEEE Standard Glossary of Software Engineering Terminology, IEEE Std 610.12-1990*. New York, NY: The Institute of Electrical and Electronics Engineers, Inc.

IEEE Std 830-1998. (1998). *Recommended Practice for Software requirement Specification*. In IEEE (Ed.). Piscataway, N.J.

Industrie Canada. (2004). *Glossaire*, 2005, tiré de http://strategis.ic.gc.ca/epic/internet/instco-levc.nsf/fr/h_qw00037f.html

ISBSG. (2005). *ISO Standard for Functional Size Measurement*, 2005, tiré de http://www.isbsg.org/html/Iso_fsm.html

ISO. (1993). *Vocabulaire international des termes fondamentaux et généraux de métrologie*. Genève: Organisation internationale de normalisation - ISO.

ISO. (2004). *Notice de droit d'auteur*, 2005, tiré de <http://www.jtc1-sc7.org/>

ISO/IEC 14143-1. (1997). *Information technology - Software measurement - Functional size measurement - Definition of concepts*. Genève: Organisation internationale de normalisation - ISO.

ISO/IEC 14143-2. (2001). *Information Technology - Software Measurement - Functional Size Measurement - Part 2: Conformity Evaluation of Software Size Measurement Methods*. Genève: Organisation internationale de normalisation - ISO.

ISO/IEC 14143-3. (2002). *Information Technology - Software Engineering - Software Measurement - Functional Size Measurement -Part3: Verification of functional size measurement*. Genève: Organisation internationale de normalisation - ISO.

ISO/IEC 14143-4. (2000). *Information Technology - Software Measurement - Functional Size Measurement - Part 4: Reference Model*. Genève: Organisation internationale de normalisation - ISO.

ISO/IEC 14143-5. (2004). *Information Technology - Software Measurement - Functional Size Measurement - Part 5: Determination of Functional Domains for use with Functional Size Measurement*. Genève: Organisation internationale de normalisation - ISO.

ISO/IEC 14143-6. (2003). *Information Technology - Software Engineering -- Software measurement -- Functional Size Measurement -- Part 6: Guide for use of ISO/IEC 14143 series and related international Standards*. Genève: Organisation internationale de normalisation - ISO.

ISO/IEC 17024. (2003). *Conformity assessment -- General requirements for bodies operating certification of persons*. Genève: Organisation internationale de normalisation - ISO.

ISO/IEC 19761. (2002). *Software Engineering - COSMIC-FFP - A functional size measurement method*. Genève: Organisation internationale de normalisation - ISO.

ISO/IEC DIS 9126-1 to 4. (2001). *Information Technology - Software Product Quality*. Genève: Organisation internationale de normalisation - ISO.

ISO/IEC Document: Guide 25. (1990). *General Requirements for the Competence of Calibration and Testing Laboratories*. Genève: Organisation internationale de normalisation - ISO.

J. Bieman, N. Fenton, D. Gustafson, Melton, A., et Ott., L. (1996). *Fundamental issues in software measurement*. In A. Melton (Ed.), *Software Measurement: Understanding Software Engineering* (pp. 39-74): International Thomson Publishing (ITP).

J. K. Kearney, R. L. Sedlmeyer, W. B. Thompson, M. A. Gray, et M. A. Adler. (1986). Software complexity measurement. *Commun. ACM*, 29(11), 1044-1050.

J. Navlaka. (1986). *Software productivity metrics: Some candidates and their evaluation*. Présenté à la Proceedings of National Computer Conference.

J. Pfanzgal. (1968). *Theory of Measurement*: Wiley, New York.

J.B. Dreger. (1989). *Function Points Analysis*.

J.C. Brown, et Shawn, M. (1981). *Toward A Scientific Basic for Software Evaluation*. Cambridge, Massachusetts.

J.M. Desharnais. (2003). *Application de la mesure fonctionnelle cosmic-ffp : une approche cognitive*. Université du Québec à Montréal (UQAM), Montréal.

J.M. Desharnais, et A. Abran. (2001). *Applying a Functional Measurement Method: Cognitive Issues*. Présenté au *International Workshop on Software Measurement*, Montréal, Québec.

J.P. Jacquet, et Abran, A. (1998). *Metrics Validation Proposals: A Structured Analysis*. Présenté au IWSM.

James Rumbaugh, Ivar, J., et Grady, B. (1999). *The Unified Modeling Language Reference Manual*: Addison-Wesley.

James W. Moore. (1997). *Software Engineering: A User's Road Software Engineering: A User's Road Map*. Los Alamitos, CA.

- John Quincy Adams. (1821). *Report upon Weights and Measures*. Washington: Congress.
- Jones, C. (1991). *Applied Software Measurement - Assuring Productivity and Quality*: McGraw-Hill.
- Jones, C. (1996). *Applied Software Measurement : Assuring Productivity and Quality* (2nd ed.). New York: McGraw Hill.
- K. Paton, et A. Abran. (1995). *A Formal Notation for the Rules of Function Point Analysis* (Rapport de Recherche No. 247). Montréal: Université du Québec à Montréal.
- K.B. Lakshmanan, S. Jayaprakash, et P.K. Sinha. (1991). Proprieties of Control-Flow Complexity Measures. *IEEE, Transactions on Software Engineering*, 17, 1289-1295.
- Kurmanadham, V. V. G. B. G. (2004). *Function Points or Lines of Code? - An Insight* (Article): Global Microsoft Business Unit, Wipro Technologies.
- L. Ejiogu. (1991). *Software Engineering with Formal Metrics*. Boston, Massachusetts: QED Technical Publishing Group.
- LRGL. (2000). *Rice Cooker Case Study*. Montreal: Université du Québec à Montréal.
- LRGL. (2004). *Rice Cooker System*. Montréal: ETS-UQAM.
- M. Henry. (1997). Notion de modèle et modélisation dans l'enseignement. In *Enseigner les probabilités au Lycée* (pp. 77-84). Reims: IREM de Reims.
- M. Lange. (2005). *Histoire du mètre, 2005*, tiré de <http://www.col-camus-soufflenheim.ac-strasbourg.fr/Page.php?IDP=249&IDD=0>
- M. Shepperd. (1992). Algebraic Models and Metric Validation. In S. Publ. (Ed.), *Denvir et. al.: Formal Aspects of Measurment*. (pp. 157-175).
- M. Shepperd. (1994, 7-9 septembre). *Some Observations on Function Points*. Présenté à la 11th CSSR Conference on Software Evolution, Models and Metrics, Dublin, Ireland.
- M.Q. Patton. (1990). *Qualitative Evaluation & Research Methods*: SAGE Publication.
- Maya, M., Abran, A., et Bourque, P. (1996). *Mesure de la taille fonctionnelle des logiciels tempsréel: Revue critique de la littérature* (Rapport de recherche). Montréal: Laboratoire de recherche en gestion des logiciels, Université du Québec à Montréal (UQAM).

- N. Fenton. (1991). *Software Metrics: a Rigorous Approach*.
- N. Fenton, R. Whitty, et Y. Iizuka. (1995). *Software Quality - Assurance and Measurement - A world Wide perspective*. Thompson Computer Press.
- N. Fenton, et S.L. Pfleeger. (1997). *Software Metrics : A Rigorous & Practical Approach*: Chapman & Hall.
- N. Schneidewind. (1992). Methodology for Validating Software Metrics. *IEEE Transactions on Software Engineering*, 18(5), 410-422.
- NESMA. (2003). *FPA Counting Guidelines of NESMA and IFPUG*. en Septembre 2003, 2003, tiré de <http://www.nesma.nl/english/nesma&ifpug.htm>
- Object Management Group. (2005). *UML Resource Page*, 2005, tiré de <http://www.uml.org/>
- Office de la langue française, Q. P. (2001). *Le Grand dictionnaire terminologique*, 2005, tiré de http://www.granddictionnaire.com/btml/fra/r_motclef/index800_1.asp
- OIML. (2002). Sphygmomanomètres non invasifs mécaniques. *Journal trimestriel de l'Organisation, publié par le Bureau International de Métrologie Légale*, 32.
- P. Bourque, et Côté. (1991). *An Experiment in Software Sizing with Structured Analysis Metrics*. *J. Systems Software*, 15, 159-172.
- P. Suppes, et Zinnes, J. L. (1963). *Basic Measurement Theory: Handbook of Mathematical Psychology*, Edited by D.D. Luce, R.R. Bush E. Galanter, Volume I, Wiley, New York.
- Pam Morris. (2003). *COSMIC-FFP and IFPUG 4.1 Similarities and Differences*. Présenté à la IFPUG Fall Conference Scottsdale, Arizona, USA.
- Picchiottino, Jean-Dominique, et Hocquart, H. (2000). *Dictionnaire mathématiques*. Paris.
- R. Bache. (1990). *Graph theory Models of Software*. Non publiée doctorat, *South Bank University*, Londres.
- R. Banker, Chang H., et Kemerer, C. F. (1994). *Evidence on Economies of Scale in Software Development*. *Information and Software Technology*, 36(5).
- R. Watts. (1987). *Measuring Software Quality*: NCC Publications.

- R.C. Bogdan, et Bilken, S. K. (1992). *Qualitative Research for Education : An introduction to theory and methods*: Allyn et Bacon.
- R.J. Rubey, et R. Hartwick. (1968). *Quantitative Measurement Program Quality*. Présenté à la National Computer Conference.
- Rational Software. (2002a). Collegiate Sports Paging System [Process]: IBM-Rational Unified Process.
- Rational Software. (2002b). Course Registration System [Process]: IBM-Rational Unified Process.
- Rational Software Corporation. (2002). Rational Rose (Version 2002.05.00).
- S. Azzouz, et Abran, A. (2004, 28-30 janvier). *A proposed measurement role in the Rational Unified Process (RUP) and its implementation with ISO 19761: COSMIC FFP*. Présenté au *Software Measurement European Forum - SMEF 2004*, Rome, Italie.
- S. Galea. (1995). *The Boeing Company: 3 D Function Point Extension. 2.0(1.0)*.
- S. Nishiyama, et T. FURUYAMA. (1994). *The validity and applicability of function point analysis - as related to specification quality and ergonomics*. Présenté à la EOQ-SC'94, Suisse.
- S. Stevens. (1946). On the Theory of Scales and Measurement. *Science Journal*, 103, 677-680.
- S. Wolff. (1999). *La Place de la Mesure au Sein des Principes Fondamentaux du GénieLogiciel* (Rapport final d'activité de synthèse de la Maîtrise en informatique de gestion). Montréal: Université du Québec à Montréal.
- S.A. Whitmire. (1995). An Introduction to 3D Function Points. *Software Development*, 43-53.
- S.B. Merriam. (1988). *Case Study Research in Education: A Qualitative Approach*. San Francisco, CA.
- S.D. Conte, H.E. Dunsmore, et V.Y. Shen. (1986). *Software Engineering Metrics and Model*. Menlo Park: Benjamin/Cummings publishing Company.
- S.J. Taylor, et R. Bogdan. (1984). *Introduction to qualitative research methods : the search for meaning* (2e ed.): John Wiley & Sons.

- S.L. Pfleeger. (1997, mars/avril 1997). *Assessing Software Measurement*. Présenté à la IEEE Software.
- S.L. Pfleeger, et J. Palmer. (1990). *Software estimation for object-oriented systems*. Présenté à la International Function Point Users Group Fall Conference, San Antonio, TX.
- S.M. Ndagijimana. (2002). *étude du processus de mesures des points de fonction et analyse de la documentation fonctionnelle*. Facultés Universitaires Notre-Dame de la Paix, Namur.
- Serge d'Agostino, Patrice Pourcel, et Philippe Deubel. (2003). *Dictionnaire des auteurs en sciences économiques et sociales* (Bréal ed.).
- Software Measurement Services. (2001). *The Power of COSMIC FFP*. en mai 2004, 2004, tiré de http://www.gifpa.co.uk/news/8/p2_cosmic.html
- T. DeMarco. (1982). *Controlling Software Projects*. New York.
- T. McCabe. (1976). A Complexity Measure. *IEEE Transactions on Software Engineering*, SE-2(4), 308-320.
- T. McCabe, et C. Butler. (1998). Design Complexity Measurement and Testing. *Communications of the ACM*, 32.
- U. Sekaran. (1992). *Research Methods for Business : A Skill-Building Approach* (2e ed.): New York ; Toronto : J. Wiley.
- V. Kurmanadham. (2004). *Function Points or Lines of Code? - An Insight* (Article): Global Microsoft Business Unit, Wipro Technologies.
- V. Prince. (1996). *Vers une informatique cognitive dans les entreprises: Le rôle central du langage*. Paris.
- V.R. Basili, et Musa, J. D. (1991). *The Future Engineering Software: A Management Perspective*. IEEE Publications.
- V.R. Basili, R W Selby, et Hutchens, D. H. (1986). *Experimentation in software engineering*. *IEEE Transactions on Software Engineering*, 12(7), 733-743.
- V.R. Basili, et Reiter, R. (1979). *Evaluating Automatable Measures of Software Development*, New York.

Whitmire, S. A. (1992, June 1). *3D Function Points : Scientific and Real-Time Extensions to FunctionPoints*. Présenté au 1992 Pacific Northwest Software Quality journal.

World Standards Services Network. (2003). *General information*, 2005, tiré de http://www.wssn.net/WSSN/gen_inf.html

Y. Lincoln, Y., et E.G. Guba. (1985). *Naturalistic Inquiry*: SAGE Publication.

