

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

IDENTIFICATION DE PATTERNS DE LA MESURE DE LA TAILLE  
FONCTIONNELLE ET DÉVELOPPEMENT D'UN OUTIL LES  
SUPPORTANT

RAPPORT DE PROJET  
PRÉSENTÉ  
COMME EXIGENCE PARTIELLE  
DE LA MAÎTRISE EN GÉNIE LOGICIEL

PAR  
JIMMY CLOUTIER

JUIN 2019

UNIVERSITÉ DU QUÉBEC À MONTRÉAL  
Service des bibliothèques

Avertissement

La diffusion de ce document diplômant se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.10-2015). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

## REMERCIEMENTS

Je tiens tout d'abord à remercier ma directrice de projet, Mme Sylvie Trudel, de m'avoir suivi durant ce long cheminement entrecoupé de nombreuses cassures, ainsi que de m'avoir gracieusement proposé une alternative de projet lorsque l'initial n'a pu se poursuivre. Je tiens également à la remercier de m'avoir fait connaître le monde de la recherche en me joignant à M. Desharnais et à elle pour la rédaction d'un article scientifique, ce qui m'a permis de participer à la conférence du IWSM de 2016 regroupant plusieurs experts et chercheurs. Je me trouve privilégié d'avoir pu m'insérer dans ce monde de la recherche et découvrir ce que cela représentait.

Des remerciements spéciaux à Simone qui, à chaque discussion, réussissait à me remotiver pour compléter cette maîtrise en me rappelant l'importance que celle-ci avait pour enraciner mon parcours et surtout mon futur.

Enfin, et non la moindre, merci à ma conjointe Lucie de m'avoir appuyé dans ce parcours académique. Sans toi à mes côtés, j'aurais arrêté tout ceci il y a fort longtemps!



# TABLE DES MATIÈRES

Remerciements	iii
Liste des figures	ix
Liste des tableaux	xi
Liste des abréviations, sigles et acronymes	xiii
Résumé	xv
Abstract	xvii
Introduction	1
<b>1. Concepts et méthodes de la mesure de la taille fonctionnelle</b>	<b>3</b>
1.1. Origine	3
1.2. Cadre normatif	4
1.3. Différentes méthodes reconnues	4
1.3.1. IFPUG	4
1.3.2. MkII	5
1.3.3. NESMA	5
1.3.4. FiSMA	5
1.3.5. COSMIC	6
1.4. Processus de mesure sous COSMIC	6
1.4.1. Phase 1 - Stratégie du mesurage	6
1.4.2. Phase 2 - Arrimage	7
1.4.3. Phase 3 - Mesurage	7
1.5. Difficultés de la mesure	7
1.6. Identification des <i>FSM patterns</i> de la taille fonctionnelle via la rédaction d'un article	8
1.7. <i>FSM patterns</i>	10
1.7.1. <i>Micro FSM pattern</i>	10
1.7.2. <i>Basic FSM pattern</i>	11
1.7.3. <i>Composite FSM pattern</i>	11
1.7.4. <i>Multi-composite FSM pattern</i>	11
1.8. Objectifs recherchés par les <i>FSM patterns</i>	12
<b>2. Problématique de l'absence d'outils informatisés supportant les <i>FSM patterns</i></b>	<b>13</b>
2.1. Objectif 1 - Réduire l'effort de la mesure	13

2.2.	Objectif 2 - Obtenir une taille dans la phase initiale d'un projet . . . . .	15
2.3.	Approche proposée . . . . .	16
<b>3.</b>	<b>Revue systématique de la littérature sur les outils informatisés supportant la prise de mesure de la taille fonctionnelle selon la méthode <i>COSMIC</i></b>	<b>17</b>
3.1.	Questions de recherche d'articles . . . . .	18
3.2.	Critères de sélection des articles . . . . .	18
3.2.1.	<i>COSMIC</i> . . . . .	19
3.2.2.	<i>Tool</i> . . . . .	19
3.2.3.	<i>Functional Size Measurement</i> (FSM) . . . . .	19
3.3.	Sélection des bases de données . . . . .	19
3.4.	Requête de recherche pour chaque base de données . . . . .	20
3.5.	Requête de recherche dans d'autres sources d'information . . . . .	21
3.6.	Résultats de la sélection - Littérature scientifique . . . . .	21
3.6.1.	Échantillonnage . . . . .	21
3.6.2.	Interprétation des résultats de la RSL . . . . .	23
3.7.	Résultats de la sélection - Industrie . . . . .	26
3.8.	Conclusion de la RSL . . . . .	28
3.8.1.	Qu'existe-t-il comme outil informatisé supportant la prise de mesure de la taille fonctionnelle selon la méthode <i>COSMIC</i> ? . . . .	29
3.8.2.	Est-ce possible de faire évoluer ces outils afin d'y implémenter le concept de <i>FSM pattern</i> ? . . . . .	29
3.8.3.	Existe-t-il des outils implémentant déjà ce concept de <i>FSM pattern</i> ? . . . .	30
3.8.4.	Constat . . . . .	30
<b>4.</b>	<b>Méthodologie de développement d'un outil supportant les <i>FSM patterns</i></b>	<b>31</b>
4.1.	Démarrage du développement d'un outil supportant les <i>FSM patterns</i> . . . . .	31
4.1.1.	Analyse des besoins . . . . .	31
4.1.2.	Choix de la technologie . . . . .	32
4.1.3.	Choix de l'architecture . . . . .	39
4.1.4.	Choix de l'hébergement . . . . .	40
4.1.5.	Planifier le développement . . . . .	41
4.2.	Développement itératif . . . . .	42
<b>5.</b>	<b>Mises en oeuvre de l'outil de mesure <i>COSMIC-Sizing-Tool</i></b>	<b>45</b>
5.1.	Première tentative - Implémentation de fonctionnalités par des étudiants du baccalauréat . . . . .	45
5.1.1.	<i>Web Application Site Map</i> . . . . .	45
5.1.2.	Hébergement . . . . .	47
5.1.3.	Modèle de base de données . . . . .	47
5.1.4.	<i>GitHub</i> . . . . .	47
5.1.5.	Aperçu des fonctionnalités (partiellement) implémentées . . . . .	49
5.2.	Deuxième tentative - Prototype de saisie de la mesure avec les <i>FSM patterns</i>	55
5.2.1.	Maquettes d'interfaces . . . . .	55

5.2.2. Modélisation . . . . .	58
5.2.3. <i>Web API</i> du prototype . . . . .	58
5.2.4. Interface utilisateur du prototype . . . . .	60
5.3. Évaluation des objectifs énoncés dans la problématique . . . . .	65
5.3.1. Objectif 1 - Réduire l'effort de la mesure . . . . .	65
5.3.2. Objectif 2 - Obtenir une taille dès la phase initiale d'un projet . . . . .	68
<b>6. Discussion sur les obstacles et défis rencontrés lors de la réalisation du projet de maîtrise</b> . . . . .	<b>69</b>
6.1. Rédaction de l'article <i>Functional Size Measurement Patterns : A Proposed Approach</i> (Trudel et al., 2016) . . . . .	69
6.2. Réalisation d'une revue systématique de la littérature . . . . .	70
6.3. Première tentative de développement du prototype <i>COSMIC-Sizing-Tool</i> . . . . .	72
6.4. Deuxième tentative de développement du prototype <i>COSMIC-Sizing-Tool</i> . . . . .	74
6.5. État du <i>COSMIC-Sizing-Tool</i> . . . . .	75
<b>Conclusion</b> . . . . .	<b>79</b>
Retour sur les objectifs . . . . .	80
<b>A. Article <i>Functional size measurement patterns : a proposed approach</i></b> . . . . .	<b>81</b>
<b>B. Résultat des requêtes de la RSL</b> . . . . .	<b>95</b>
<b>C. Aperçu du Excel résultant de la RSL</b> . . . . .	<b>99</b>
<b>D. Carte conceptuelle des outils supportant la mesure</b> . . . . .	<b>101</b>
<b>E. Classification des articles</b> . . . . .	<b>103</b>
<b>F. <i>COSMIC-Sizing-Tool Backlog</i></b> . . . . .	<b>105</b>
<b>G. <i>Developer's Guide for the COSMIC-Sizing-Tool</i></b> . . . . .	<b>129</b>
G.1. Get sources . . . . .	129
G.2. Folder structure . . . . .	129
G.2.1. backend folder . . . . .	130
G.2.2. <i>Frontend</i> folder . . . . .	130
G.3. How to build . . . . .	131
G.3.1. <i>Back end</i> . . . . .	131
G.3.2. <i>Front end</i> . . . . .	134
G.4. Architecture . . . . .	135
G.4.1. Overview . . . . .	135
G.4.2. <i>Back end</i> . . . . .	135
G.4.3. <i>Front end</i> . . . . .	135
G.5. Coding standards . . . . .	135
G.5.1. <i>Back end</i> . . . . .	135

G.5.2. <i>Front end</i> . . . . .	136
G.6. Dev. Environnement . . . . .	136
<b>Bibliographie</b>	<b>137</b>

## LISTE DES FIGURES

1.1. <i>Best Paper</i> IWSM-MESURA . . . . .	9
1.2. Hiérarchie des <i>FSM patterns</i> (tiré de Trudel <i>et al.</i> , 2016) . . . . .	10
2.1. Solution du <i>FSM pattern Composite CRUDL-3DG</i> tiré de Trudel <i>et al.</i> (2016) . . . . .	14
4.1. Aperçu de la méthodologie . . . . .	31
4.2. Tendance des recherches <i>Python</i> (en bleu) versus <i>Java</i> (en rouge) . . . . .	34
4.3. Tendance des recherches <i>React</i> (en bleu) versus <i>Angular</i> (en rouge) . . . . .	35
4.4. Tendance des recherches modèle relationnel (en bleu) versus <i>NoSQL</i> (en rouge) . . . . .	38
4.5. Tendance des SGBD relationnels . . . . .	38
5.1. <i>Site Map</i> du <i>COSMIC-Sizing-Tool</i> . . . . .	46
5.2. Modèle relationnel du <i>COSMIC-Sizing-Tool</i> . . . . .	48
5.3. Kanban <i>COSMIC-Sizing-Tool</i> dans waffle . . . . .	49
5.4. <i>COSMIC-Sizing-Tool</i> : Page d'accueil . . . . .	50
5.5. <i>COSMIC-Sizing-Tool</i> : Inscription . . . . .	51
5.6. <i>COSMIC-Sizing-Tool</i> : Authentification . . . . .	52
5.7. <i>COSMIC-Sizing-Tool</i> : Création d'un projet . . . . .	53
5.8. <i>COSMIC-Sizing-Tool</i> : Liste des <i>patterns</i> . . . . .	54
5.9. <i>COSMIC-Sizing-Tool</i> : Édition d'un <i>pattern</i> . . . . .	54
5.10. <i>COSMIC-Sizing-Tool</i> : Sommaire de la mesure . . . . .	55
5.11. Prototype : Arborescence composée de contrôles multiples . . . . .	56
5.12. Prototype : Arborescence avec grille . . . . .	57
5.13. Prototype : Similaire à Excel . . . . .	57
5.14. Prototype : Liste des organisations . . . . .	61
5.15. Prototype : Liste des projets et processus fonctionnels du premier projet . . . . .	61
5.16. Prototype : Liste des groupes de données du processus fonctionnel sélectionné . . . . .	62
5.17. Prototype : Processus fonctionnels - boutons d'action . . . . .	62
5.18. Prototype : Groupes de données - boutons d'action . . . . .	63
5.19. Prototype : Édition au niveau de la liste . . . . .	63
5.20. Prototype : Liste des <i>patterns</i> . . . . .	64
5.21. Prototype : Spécification des groupes de données . . . . .	64
5.22. Prototype : Mesure de la taille fonctionnelle en appliquant un <i>pattern</i> . . . . .	65
C.1. Excel RSL : onglet Métrique . . . . .	99

C.2. Excel RSL : onglet <i>Papers</i> . . . . .	99
C.3. Excel RSL : onglet SCO1 . . . . .	99
C.4. Excel RSL : onglet SCO-Merge . . . . .	100
C.5. Excel RSL : onglet SPL01 . . . . .	100
C.6. Excel RSL : onglet SPL-Merge . . . . .	100
D.1. Outils supportant la mesure selon COSMIC : carte conceptuelle . . . . .	102
G.1. <i>Back end</i> running in command line . . . . .	133
G.2. Architecture overview . . . . .	135

## LISTE DES TABLEAUX

3.1. Résultat des requêtes fusionnées . . . . .	22
5.1. Codes <i>HTTP</i> . . . . .	60
5.2. Temps requis - <i>COSMIC-Sizing-Tool</i> . . . . .	66
5.3. Temps requis - Excel - <i>Basic Create</i> . . . . .	66
B.1. Résultat des requêtes . . . . .	95



## LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

**API** Application Programming Interface  
**BPM** Business Process Modeling  
**CFP** COSMIC Function Point  
**COSMIC** COmmon Software Measurement International Consortium  
**CRUD-L** Create, Retrieve, Update, Delete et List  
**CSS** Cascading Style Sheets  
**CSV** Comma Separated Value  
**DBA** Database Administrator  
**DG** Data Group (groupe de données)  
**DRY** Don't Repeat Yourself  
**DSRM** Design Science Research Methodology  
**ÉTS** École de Technologie Supérieure  
**FiSMA** Finnish Software Measurement Association  
**FP** Functional Processus (processus fonctionnels)  
**FPA** Function Point Analysis  
**FSM** Functional Size Measurement  
**FUR** Functional User Requirement  
**HTML** Hyper Text Markup Language  
**HTTP** Hypertext Transfer Protocol  
**ID** Identifiant  
**IDE** Integrated Development Environment  
**IEC** International Electrotechnical Commission  
**IEEE** Institute of Electrical and Electronics Engineers  
**IFPUG** International Function Point Users Group  
**ISBSG** International Software Benchmarking Standards Group  
**ISO** International Organization for Standardization  
**IWSM** International Workshop on Software Measurement  
**JSON** JavaScript Object Notation  
**KLM** Keystroke-level Model  
**M. Ing.** Maître en Ingénierie  
**MGL** Maîtrise en Génie Logiciel  
**MkII** Mark II  
**MVC** Model-View-Controller  
**NB** Nombre  
**NESMA** Netherlands Software Metrics Users Association  
**NLP** Natural Language Processing  
**ORM** Object Relational Mapping

**PO** Product Owner  
**PRiM** Process Reengineering i\* Method  
**REST** Representational State Transfer  
**RSL** Revue Systématique de la Littérature  
**SGBD** Système de Gestion de Base de Données  
**SI** Système d'Information  
**SRS** Software Requirements Specification  
**UML** Unified Modeling Language  
**UQAM** Université du Québec à Montréal  
**XML** eXtensible Markup Language

## RÉSUMÉ

La mesure de la taille fonctionnelle selon la méthode *COSMIC* fait face à divers enjeux, entre autres, en lien avec la formation des nouveaux mesureurs, l'effort nécessaire à la mesure, l'obtention d'une mesure en début de projet et l'assurance d'une mesure précise. Ces divers enjeux ont motivé la création des *FSM Patterns* tels que décrits par les auteurs Sylvie Trudel, Jean-Marc Desharnais et Jimmy Cloutier dans l'article *Functional Size Measurement Patterns : A Proposed Approach*. Des hypothèses y sont émises selon lesquelles les *FSM patterns* permettraient de répondre favorablement à ces enjeux.

Partant de cette prémisse, ce projet se penche plus particulièrement sur les enjeux de l'obtention d'une mesure avec peu d'effort ainsi que l'obtention de celle-ci en début de projet. Ces deux enjeux spécifiques nécessitent un outil informatisé afin d'être validés.

Ce rapport de projet documente les diverses activités effectuées afin de valider les deux objectifs suivants dans le contexte de la mesure fonctionnelle : 1) réduire l'effort de la mesure et 2) obtenir une taille fonctionnelle dans la phase initiale d'un projet. Pour ce faire, une revue systématique de la littérature a été effectuée afin d'identifier les divers outils informatisés supportant la mesure de la taille fonctionnelle selon *COSMIC*, dans le but d'implémenter les *FSM Patterns* dans l'un d'entre eux. Cette recherche s'étant avérée infructueuse, des recherches supplémentaires ont été réalisées, portant sur les outils de mesure payants proposés par l'industrie. Les résultats étant similaires, deux tentatives ont été effectuées afin de développer un outil de mesure de la taille fonctionnelle selon *COSMIC* implémentant le concept de *FSM Patterns*. Ceci a permis de créer un prototype portant le nom de *COSMIC-Sizing-Tool*. Dans l'état actuel du *COSMIC-Sizing-Tool*, les deux objectifs semblent atteignables, mais devront être réévalués dans l'éventualité où un tiers poursuivrait ce développement.



## ABSTRACT

Functional size measurement using the *COSMIC* method addresses various issues, among others, training of new measurers, effort required to measure, possibility of having a measure at the beginning of the project and production of an accurate measure. These various issues motivated the creation of the *FSM Patterns*, as described by the authors Sylvie Trudel, Jean-Marc Desharnais, and Jimmy Cloutier in the article *Functional Size Measurement Patterns : A Proposed Approach*. Assumptions are made that the *FSM patterns* would respond positively to these issues.

Based on this, the current project focuses on the issues involved in obtaining a measure with little effort and at the beginning of a project. These require a software to be validated.

This project report documents the various activities required to validate the two following objectives : 1) reducing the measurement effort and 2) obtaining a functional size in the initial phase of a project. To do this, a systematic review of the literature was performed to identify the various computerized tools supporting *COSMIC* functional size measurement, and to implement the *FSM Patterns* in one of them. This research was unsuccessful. Additional researches were performed on measurement tools proposed by the industry. Since the results were similar, two attempts were made to develop a functional size measurement tool according to *COSMIC* implementing the concept of *FSM Patterns*. A prototype called *COSMIC-Sizing-Tool* was created. In the current state of the *COSMIC-Sizing-Tool*, the two objectives seem achievable, but will have to be reassessed assuming a third party continues the tool development.

## INTRODUCTION

« *Do or do not. There is not try.* »

- *Yoda*

Ce projet de synthèse s'inscrit dans le cadre de la maîtrise en génie logiciel à l'Université du Québec à Montréal (UQAM). L'objectif académique de cette synthèse est de mettre en pratique les savoirs acquis durant la formation afin de consolider l'apprentissage reçu et démontrer la maîtrise de ce qui a été enseigné.

Plus particulièrement, dans le présent contexte, les notions apprises au niveau de la gestion de projet, de l'élicitation des exigences, de la conception, de la réalisation, de la vérification et de l'assurance qualité logicielle ont été essentielles. En effet, l'artefact résultant du projet de synthèse a permis d'instaurer les bases d'un logiciel libre (*open source*) spécialisé dans la prise de mesure de la taille fonctionnelle en accord avec la méthode *COSMIC* (ISO/IEC19761 :2011, 2011).

Par ailleurs, cette synthèse comporte également un volet recherche non négligeable dans le sens où l'outil réalisé intègre la notion de « *pattern* de la mesure de la taille fonctionnelle ». Cette notion de *pattern* correspond à un nouveau concept qui a été proposé dans l'article « *Functional Size Measurement Patterns : A Proposed Approach* » (Trudel *et al.*, 2016). L'un des buts recherchés par l'outil développé par l'implémentation des *patterns* est de valider l'utilisation de ce nouveau concept dans un outil informatisé afin de, entre autres, faciliter et accélérer la prise de la mesure selon *COSMIC*.

Afin de synthétiser l'ensemble, ce rapport de projet est structuré comme suit. Le chapitre 1 décrit les concepts et les méthodes de la mesure de la taille fonctionnelle du logiciel. Quant au chapitre 2, il contient la problématique de l'absence d'outils informatisés supportant les *FSM patterns*. À cela s'ajoute, au chapitre 3, une revue systématique de la littérature sur les outils informatisés supportant la prise de mesure de la taille fonctionnelle selon la méthode *COSMIC*. La méthodologie de développement d'un outil supportant les *FSM patterns* compose le chapitre 4. Puisqu'un développement logiciel a été effectué dans le cadre de ce travail, l'outil développé est présenté au chapitre 5. Enfin,

le rapport se termine avec un chapitre 6 présentant une discussion sur les obstacles et défis rencontrés lors de la réalisation du projet de maîtrise dans son ensemble.

## CHAPITRE 1.

# CONCEPTS ET MÉTHODES DE LA MESURE DE LA TAILLE FONCTIONNELLE

Dans ce chapitre, les concepts et méthodes de la mesure de la taille fonctionnelle seront abordés, jusqu'à la définition des *FSM patterns*. Plus particulièrement, en peignant sommairement l'histoire de la mesure de la taille fonctionnelle, il sera question des origines de celle-ci et du cadre normatif qui résulte de son évolution. Ce cadre normatif a permis de reconnaître certaines méthodes implémentant la mesure de taille fonctionnelle telles que *IFPUG*, *MkII*, *NESMA*, *FiSMA* et *COSMIC*. Chacune de ces méthodes sera présentée brièvement. Ensuite, le processus de mesure de la méthode *COSMIC* sera détaillé puisque ce projet de synthèse s'intéresse tout particulièrement à celle-ci. La raison est que c'est en utilisant *COSMIC* que le concept de *Pattern* de la mesure de la taille fonctionnelle a été énoncé la première fois ; ce concept de *Pattern* est le sujet principal de ce projet de synthèse. De plus, il sera question de l'article fondateur décrivant les *FSM patterns* en guise d'introduction aux catégories des *FSM patterns* qui seront, par la suite, décrites sommairement. Enfin, le chapitre se conclura sur les objectifs poursuivis par les *FSM patterns*.

### 1.1. Origine

Il faut remonter à la fin des années 70 pour trouver l'origine de la mesure de la taille fonctionnelle. Durant cette période, Allan Albrecht, travaillant pour IBM, propose une nouvelle méthode permettant de mesurer la taille d'une application afin de calculer la productivité de projets (Albrecht, 1979). À cette époque, une telle mesure s'effectuait en comptabilisant/estimant les composantes physiques résultant de la programmation telles que les lignes de code. L'approche novatrice d'Albrecht fut de s'intéresser aux fonctionnalités à implémenter plutôt qu'aux lignes de code. En procédant ainsi, il devenait possible de comparer des projets utilisant des langages et technologies différentes sous une base commune.

Cette première méthode de mesure de la taille fonctionnelle s'intitule *Function Point Analysis* (FPA). Elle consistait à compter individuellement le nombre de *user inputs*, *inquiries*, *outputs* et *master files* à implémenter pour donner vie au projet. Suite à cela, chacun de ces facteurs se voyait accorder un poids (multiplicateur). Albrecht (1979) mentionne qu'il a utilisé les poids suivants pour chacun des critères : *Inputs* x 4, *Outputs* x 5, *Inquiries* x 4 et *Master files* x 10. À noter que ceux-ci ont été établis suite à différents essais et débats. Enfin, la somme résultant de l'addition de tous les facteurs multipliés par leur poids correspond aux *function points*. Ainsi, les bases étaient jetées pour inspirer diverses autres méthodes de la mesure de la taille fonctionnelle. Pour ce faire, il a été nécessaire de spécifier un cadre normatif servant à la définition des normes à respecter pour qu'une méthode soit considérée sérieuse.

## 1.2. Cadre normatif

Vers la fin des années 90, l'International Organization for Standardization (ISO) a défini les principes régissant la mesure de la taille fonctionnelle. Celles-ci ont été publiées dans ISO/IEC 14143 Standard. ISO a également reconnu des méthodes définissant l'implémentation de la mesure de la taille fonctionnelle telles que *IFPUG*, *MkII*, *NESMA*, *FiSMA* et *COSMIC*.

## 1.3. Différentes méthodes reconnues

### 1.3.1. IFPUG

L'*International Function Point Users Group* (*IFPUG*) a été fondé en 1986 sur les bases de la taille fonctionnelle telle qu'énoncée initialement par Albrecht. *IFPUG* a entre autres permis de préciser les règles de calcul de la méthode. Cette méthode correspond à la première génération de la mesure de la taille fonctionnelle. Son application est limitée à des systèmes d'information. Plusieurs autres méthodes découlent de *IFPUG*, celles-ci sont donc, par extension, également de première génération. *IFPUG* a été certifiée ISO en 2003 et la révision la plus récente date de décembre 2009. Elle est identifiée en tant qu'« ISO/IEC 20926 :2009 »<sup>1</sup>.

Site web : <http://www.ifpug.org>

---

1. <https://www.iso.org/standard/51717.html>

### 1.3.2. MkII

Les travaux de Charles Symons sont à l'origine de la méthode *Mark II (MkII)* qui est une méthode de première génération. Tout comme *IFPUG*, cette méthode est basée sur les travaux de Albrecht. Le maintien de la norme est désormais sous la responsabilité de l'*UK Software Metrics Association*. La norme ISO de *MkII* FPA (1.3.1) est ISO/IEC 20968 :2002.<sup>2</sup>

Site web : <http://www.uksma.co.uk>

### 1.3.3. NESMA

Originellement appelée NEFPUG (*Vergeniging Nederlandse Functie Punt Gebruikers*), la *NESMA (Netherlands Software Metrics Users Association)* a été créée en mai 1989 par la *Nederlandse Software Metrieken Gebruikers Associatie*. Celle-ci a initialement été fondée sur les principes de *IFPUG* FPA suite à des discordances sur la méthode de calcul de celle-ci, jugée trop technique. Maintenant que *IFPUG* a évolué, elles sont considérées comme équivalentes. *NESMA* est connue sous la norme ISO/IEC 24570 :2018.<sup>3</sup>

Site web : <https://nesma.org>

### 1.3.4. FiSMA

La *Fisma Functional Size Measurement Method 1.1* a été développée par la *Finnish Software Measurement Association (FiSMA)*. À sa création en 1992 et jusqu'à 1998, *FiSMA* s'appelait *LATURI user group*. La norme ISO de cette méthode est ISO/IEC 29881 :2010.<sup>4</sup> Contrairement aux méthodes mentionnées précédemment, la notion de services offerte par l'application joue un rôle important dans le calcul de la taille fonctionnelle.

Site web : <http://www.fisma.fi>

---

2. <https://www.iso.org/standard/35603.html>

3. <https://www.iso.org/standard/72505.html>

4. <https://www.iso.org/standard/56418.html>

### 1.3.5. COSMIC

L'organisation *COSMIC* (*Common Software Measurement International Consortium*) date de 1998. La méthode *COSMIC* correspond à la norme ISO/IEC 19761. Elle a été fondée par plusieurs experts internationaux (australiens, européens et nord-américains). Elle est la première méthode dite de *seconde génération*. Cette méthode permet de mesurer aussi bien des logiciels de gestion de l'information que des logiciels *temps réel* et *embarqués*.

Site web : <https://cosmic-sizing.org>

## 1.4. Processus de mesure sous COSMIC

Le processus de mesure de la taille fonctionnelle selon la méthode *COSMIC* s'effectue en suivant les trois phases ci-dessous telles que documentées dans le *Manuel de mesurage COSMIC* (Abran *et al.*, 2015).

### 1.4.1. Phase 1 - Stratégie du mesurage

La première phase du processus se nomme *Stratégie du mesurage*. Celle-ci permet d'établir l'utilité recherchée par la mesure ainsi que les limites de cette mesure.

Pour ce faire, cette phase est divisée selon les quatre activités suivantes :

1. Déterminer la raison d'être du mesurage
2. Déterminer le périmètre et le niveau de décomposition du logiciel à mesurer
3. Identifier les utilisateurs fonctionnels
4. Déterminer le niveau de granularité du logiciel (portion de logiciel) à mesurer

Une fois ces informations recueillies, celles-ci doivent être consignées, car *l'omission de définir et d'enregistrer ces paramètres de façon cohérente conduira à des résultats de mesurage qui ne peuvent pas être interprétés de manière fiable et pouvant être comparés, ou être utilisés avec fiabilité comme entrée pour des processus, telle l'estimation de projet* (Abran *et al.*, 2015, p. 19).

### 1.4.2. Phase 2 - Arrimage

La deuxième phase du processus est intitulée *Arrimage*. C'est dans celle-ci que les *Functional user requirements* (FUR) du logiciel (ou de la portion de logiciel) à mesurer sont traités afin de les arrimer au modèle générique *COSMIC* et ainsi permettre la mesure de ces FURs.

Cette phase se décompose en quatre étapes :

1. Identifier les processus fonctionnels (*FP*)
2. Identifier les groupes de données (*DG*)
3. Identifier les attributs des données (étape optionnelle)
4. Identifier les mouvements de données

À la fin de cette phase, tous les mouvements de données d'entrée (E), de sortie (S), de lecture (L) et d'écriture (C) sont répertoriés pour chacun des groupes de données de tous les processus fonctionnels correspondant aux FURs du logiciel à mesurer.

### 1.4.3. Phase 3 - Mesurage

La dernière phase du processus de mesure de la taille fonctionnelle selon la mesure *COSMIC* porte le nom de *Mesurage*. C'est à cette étape que la mesure a lieu en utilisant les données récoltées lors de la phase 2. Pour ce faire, chaque mouvement de données correspond à 1 *COSMIC Function Point* (CFP).

Il suffit donc de faire la somme de tous les mouvements de données correspondant à tous les groupes de données rattachés aux processus fonctionnels afin d'obtenir la taille totale du logiciel (ou portion de logiciel) en CFP.

## 1.5. Difficultés de la mesure

Le processus de mesure de la taille fonctionnelle selon *COSMIC* peut sembler simple à implémenter. Cependant, plusieurs articles documentent les difficultés éprouvées par les mesureurs, particulièrement par ceux les moins expérimentés (Trudel et Abran, 2009). Il faut cependant noter que la plupart des erreurs de mesurage sont dues à des ambiguïtés dans les exigences fonctionnelles. Les erreurs commises ont comme conséquence de fausser la mesure de la taille fonctionnelle ce qui a un impact direct, entre autres,

sur les divers leviers d'un projet informatique (coût, durée) ou sur les améliorations de processus utilisant la taille fonctionnelle comme mesure de progrès.

Une autre difficulté rencontrée par la mesure est celle de son adoption. La mesure d'un logiciel, même par un mesureur expérimenté, nécessite un certain effort. Malgré les avantages que l'on peut en tirer, cet effort supplémentaire est souvent jugé superflu puisqu'il ne fait pas progresser directement le projet. Dans ces circonstances, aucun budget n'est dégagé au niveau du projet afin de réaliser cette activité.

Afin de minimiser les erreurs ainsi qu'accélérer la mesure de la taille fonctionnelle, plusieurs se sont tournés vers des tentatives d'automatisation de celle-ci. La plupart des différentes méthodes d'automatisation proposées sont basées sur les différents artefacts (diagrammes UML, code source, exigences, etc.) créés lors d'un projet de développement logiciel. Au lieu de se baser sur une automatisation de la mesure utilisant ces artefacts, l'article *Functional Size Measurement Patterns : A Proposed Approach* de Trudel, Desharnais et Cloutier (2016), élabore un nouveau concept basé sur les observations effectuées par les chercheurs Trudel et Desharnais lors de leurs nombreuses expériences professionnelles en tant que mesureurs *COSMIC* certifiés.

## **1.6. Identification des *FSM patterns* de la taille fonctionnelle via la rédaction d'un article**

L'étape initiatrice à ce projet de synthèse correspond à l'identification des *FSM patterns* de la taille fonctionnelle. Même si celle-ci est moins documentée dans le présent document, cette phase s'avère importante et essentielle. En effet, ceci s'inscrit intégralement dans le présent projet de synthèse, car il correspond à la portion *Identification de patterns de la mesure de la taille fonctionnelle*.

En effet, c'est lors de cette phase que le concept des *FSM patterns* s'est précisé, particulièrement grâce aux discussions entre les professeurs-chercheurs universitaires Sylvie Trudel (*UQAM*) et Jean-Marc Desharnais (*ÉTS*) ainsi que moi-même. Les idées résultantes de ces échanges ont permis de rédiger un article (Trudel *et al.*, 2016) sur le sujet afin de définir ce concept (voir l'article intégral à l'annexe A).

Plusieurs activités ont été nécessaires afin d'aider à la rédaction de cet article scientifique sur une telle nouveauté et couvrant plusieurs sujets. En plus d'avoir rédigé et révisé certaines sections de l'article, ce qui suit documente sommairement les autres activités résultantes de ma contribution directe à la réalisation de celui-ci.

Tout d'abord, des revues de littérature sommaires ont été faites sur divers sujets tels que le concept générique des *patterns*, la mesure de la taille fonctionnelle et la méthode *COSMIC*, les difficultés rencontrées des mesureurs *COSMIC* inexpérimentés et l'automatisation de la mesure selon *COSMIC*.

Ceci a permis d'asseoir sur des bases solides le contenu de l'article à rédiger. Par exemple, les articles trouvés concernant le concept générique de *patterns* ont permis d'identifier une structure adéquate pour définir les *FSM patterns*. En effet, suite à ma suggestion, il a été décidé d'utiliser une forme *Alexandrian* réduite telle que décrite dans Riehle et Züllighoven (1996). Ces recherches ont également permis de s'assurer que le concept de *FSM pattern* était bel et bien nouveau.

Après avoir moi-même soumis diverses structures pour que l'article prenne forme, l'article a été rédigé selon le cadre *Design Science Research Methodology* (DSRM) tel que décrit dans Peffers *et al.* (2007) afin que le tout soit présenté de façon cohérente. En effet, le DSRM est destiné à fournir un cadre permettant de créer un nouvel artefact ou de nouvelles connaissances, ce qui correspond précisément à l'identification des *FSM patterns* et à la définition de ce nouveau concept.

Concernant les retombées, l'article Trudel *et al.* (2016) a été publié et a fait l'objet d'une présentation par Mme Trudel lors de la conférence IWSM-MENSURA de 2016 à Berlin. Lors de cette conférence, l'article a remporté le prix du *Best Paper* tel que démontré par l'imprimé d'écran suivant récupéré de l'historique des conférences IWSM-MENSURA<sup>5</sup>.

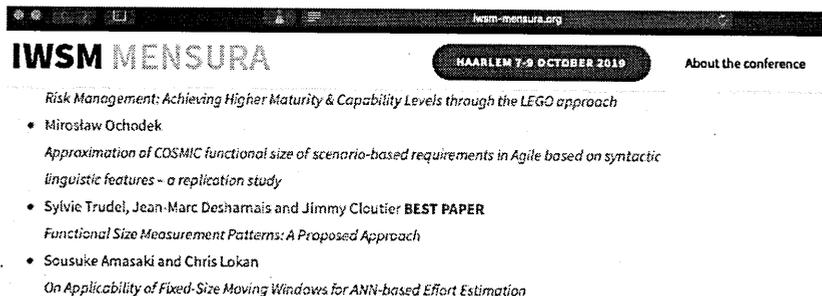


FIGURE 1.1. – *Best Paper* IWSM-MENSURA

5. <http://www.iws-smensura.org/2016-conference/program/>

## 1.7. FSM patterns

Le terme *FSM pattern* est défini ainsi (traduction libre) : « [...] correspond à un modèle générique prédéfini d'un logiciel permettant de répondre à une situation de mesure récurrente dans un contexte spécifique » (Trudel et al., 2016). Plus concrètement dans le cadre de la méthode *COSMIC*, un *FSM pattern* correspond aux mouvements de données d'un ou plusieurs groupes de données relativement à une portion, à un seul ou à plusieurs *FUR* correspondant à une situation survenant régulièrement. Le bien connu « *CRUD-L* » (*Create, Retrieve, Update, Delete et List*) nécessaire à l'implémentation des systèmes sert d'exemple dans l'article pour illustrer les différentes catégories de *FSM patterns* tels que le *micro FSM pattern*, le *basic FSM pattern*, le *composite FSM pattern* et, enfin, le *multi-composite FSM pattern*. Ces différentes catégories représentent la structure hiérarchique de composition des *FSM patterns* en allant du « *micro* » (le plus petit) jusqu'au « *multi-composite* » (le plus grand).

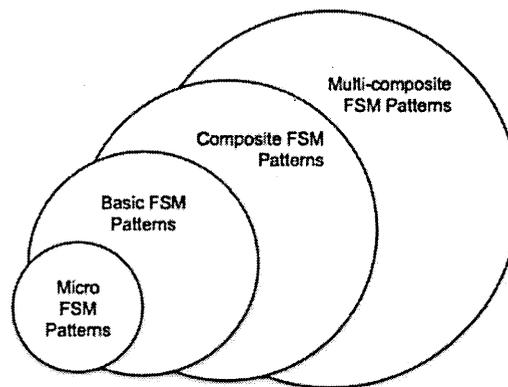


FIGURE 1.2. – Hiérarchie des *FSM patterns* (tiré de Trudel et al., 2016)

Ce qui suit définit les différents *patterns* selon les définitions tirées de l'article de Trudel et al. (2016).

### 1.7.1. Micro FSM pattern

Le *micro FSM pattern* représente un *FSM pattern* dans son niveau le plus petit. Celui-ci ne correspond qu'à une portion d'un processus fonctionnel. Ainsi, un seul ou plusieurs groupes de données reliés au processus fonctionnel peuvent former ce type de *FSM pattern*. Par exemple, ce peut être la portion des mouvements de données requis dans un ensemble de fonctionnalités pour traiter les droits d'accès de l'utilisateur à certaines

données (volet sécurité). Cependant, il est important que les groupes de données du processus fonctionnel en question ne soient pas tous définis en tant que *micro FSM pattern*. Dans une telle circonstance, ceci fait référence à la prochaine catégorie de *FSM pattern*.

### 1.7.2. *Basic FSM pattern*

Le *basic FSM pattern* correspond au second niveau de granularité. Contrairement au *micro* qui ne correspond qu'à une portion d'un processus fonctionnel, le *basic FSM pattern* représente un seul et même processus fonctionnel complet avec l'ensemble de ses groupes de données. Par exemple, les fonctionnalités d'affichage d'une liste correspondent souvent à un *basic FSM pattern*, c'est-à-dire qu'il est fréquent que les fonctionnalités d'affichage d'une liste aient sensiblement la même taille fonctionnelle puisqu'elles auront les mêmes mouvements de données.

### 1.7.3. *Composite FSM pattern*

Le *composite FSM pattern* tire son nom du fait qu'il représente un regroupement de plusieurs *basic FSM patterns* ayant une correspondance fonctionnelle entre chacun d'eux. Il correspond donc au troisième niveau de granularité. L'exemple classique est le regroupement de fonctionnalités « CRUD-L », où chacune en soi pourrait être un *basic FSM pattern*, mais comme on les retrouve souvent ensemble, alors ce *pattern* devient un regroupement de *basic FSM patterns*.

### 1.7.4. *Multi-composite FSM pattern*

Le quatrième et dernier niveau de granularité des *FSM patterns* se nomme *multi-composite*. Celui-ci est formé de plusieurs *composite FSM patterns* et de *basic FSM patterns*. Ce dernier niveau est davantage utile pour représenter des modules complets, comme un module de gestion des comptes à recevoir, de gestion des employés, de gestion de l'inventaire et des produits, etc.

De nouveaux *FSM patterns* peuvent également être définis par l'utilisateur afin de représenter son contexte d'affaires. Ils ne sont donc pas limités à ceux de la section III de Trudel *et al.* (2016).

Par exemple, un utilisateur pourrait spécifier un *multi-composite FSM pattern* représentant un module typique de son application. Lors de la création d'un nouveau module, il pourrait sélectionner ce *multi-composite FSM pattern* puis définir les particularités du module tel que le(s) nom(s) des processus fonctionnels et les divers groupes de données.

Pour ce faire, un outil est requis pour définir les *FSM patterns* de l'utilisateur à un niveau générique pour, ensuite, être en mesure de les utiliser dans un contexte spécifique.

### 1.8. Objectifs recherchés par les *FSM patterns*

Dans Trudel *et al.* (2016), quatre objectifs recherchés par les *FSM patterns* sont énoncés, soit (traduction libre) :

1. Aider les mesureurs inexpérimentés ;
2. Réduire l'effort de mesure ;
3. Obtenir une taille dans la phase initiale d'un projet ;
4. Assurer l'exactitude de la mesure.

Afin de réaliser les objectifs de « réduire l'effort de mesure » et d'« obtenir une taille dans la phase initiale d'un projet », un logiciel implémentant les *FSM patterns* est préférable. Ceci a donc motivé les objectifs de la problématique présentés dans le prochain chapitre.

## CHAPITRE 2.

### PROBLÉMATIQUE DE L'ABSENCE D'OUTILS INFORMATISÉS SUPPORTANT LES *FSM PATTERNS*

Tel qu'énoncé au chapitre 1, afin que les *FSM patterns* aient pleinement du sens, particulièrement au niveau de l'atteinte des objectifs de « réduire l'effort de mesure » et « obtenir une taille dans la phase initiale d'un projet », un outil implémentant le concept de *FSM pattern* est souhaitable.

#### 2.1. Objectif 1 - Réduire l'effort de la mesure

Obtenir la taille fonctionnelle selon *COSMIC* nécessite plusieurs opérations. En effet, tel qu'énoncé au chapitre 1, il incombe d'identifier :

1. les processus fonctionnels ;
2. les groupes de données ;
3. les attributs des données (étape optionnelle) ;
4. les mouvements de données (Entrée, Sortie, Lecture et Écriture).

Par la suite, les mouvements de données sont comptabilisés afin d'obtenir la mesure de la taille fonctionnelle en CFP.

La mesure de la taille fonctionnelle en *COSMIC* est donc possible en consignnant toutes ces informations, habituellement sous format électronique à la manière d'un tableur. L'avantage d'un tableur est de faciliter le décompte des mouvements de données. Cependant, le reste de l'information, tels que les processus fonctionnels, les groupes de données et les mouvements de données, doit être saisi manuellement à chaque fois, ce qui nécessite un certain effort.

En utilisant un outil informatisé implémentant les *FSM patterns*, au lieu de saisir manuellement à répétition l'information des groupes de données et des mouvements de

données, cette information n'a qu'à être saisie qu'une seule fois, puis être enregistrée en tant que *pattern*. En effet, une fois le *FSM pattern* défini, l'information des groupes de données y est déjà rattachée. Pour ce qui est de renseigner l'information relative aux groupes de données, il a été prévu que l'outil devrait implémenter une fonctionnalité permettant de saisir les groupes de données de façon générique au niveau de la définition du *FSM pattern*, puis de les remplacer par le nom spécifique du groupe de données lors de l'utilisation du *FSM pattern*.

Enfin, ceci peut être aisément illustré en utilisant la solution énoncée pour le *FSM pattern* intitulée *composite CRUDL-3DG* dans Trudel *et al.* (2016).

Functional Process	Data Group	Data Mvnts	Funct. Size (in CFP)	Remark
Create <First DG>	<First DG>	ERW	3	Create new occurrence
	<Second DG>	RX	2	Read and display list
	<Third DG>	RX	2	Read and display list
	Error message	X	1	<b>Subtotal: 8 CFP</b>
Retrieve <First DG>	<First DG>	ERX	3	Select, read and display existing occurrence
	<Second DG>	RX	2	Must read its ID to display its name
	<Third DG>	RX	2	Same as above
	Error message	X	1	<b>Subtotal: 8 CFP</b>
Update <First DG>	<First DG>	ERW	3	Update existing occurrence
	<Second DG>	RX	2	Read and display list
	<Third DG>	RX	2	Read and display list
	Error message	X	1	<b>Subtotal: 8 CFP</b>
Delete a <First DG>	<First DG>	ERW	3	Delete an occurrence, reads it 1st, no other DG required
	Message	X	1	<b>Subtotal: 4 CFP</b>
List <First DG>	<First DG>	RX	2	Read and display list
	Filter	<b>E</b>	1	Search filter applicable to all DGs
	<Second DG>	RX	2	Read/display list (filter)
	<Third DG>	RX	2	Same as above
	Error message	X	1	<b>Subtotal: 8 CFP</b>
		<b>Total:</b>	<b>36</b>	For this FSM pattern

<sup>a</sup> Legend: E=Entry; X=Exit; R=Read; W=Write; The "E" in bold represents the triggering Entry of its functional process.

FIGURE 2.1. – Solution du *FSM pattern Composite CRUDL-3DG* tiré de Trudel *et al.* (2016)

L'analyse de l'information contenue dans le tableau permet de réaliser que, lors de la création de la définition de ce *FSM pattern*, les groupes de données sont nommés de façon générique représentée entre <...> (*First DG*, *Second DG* et *Third DG*). En utilisant un outil informatisé, ces groupes de données génériques seront remplacés par

le terme adéquat selon le contexte en une seule et même étape. Une telle opération n'est pas possible sans un tel outil. En effet, une consignation manuelle de la mesure nécessite que la saisie des groupes de données s'effectue à chaque fois.

En réduisant la saisie nécessaire à la mesure en utilisant un outil implémentant les *FSM patterns* ainsi que la fonctionnalité de remplacer en une seule étape les groupes de données génériques par les termes spécifiques représentatifs, on peut en déduire que l'effort nécessaire à la mesure devrait être réduit.

## 2.2. Objectif 2 - Obtenir une taille dans la phase initiale d'un projet

Dans le même ordre d'idée qu'énoncé précédemment, puisque la mesure nécessite l'élicitation de diverses informations, il est difficile d'avoir une mesure juste et fiable lors du lancement d'un projet. Pour ce faire, il existe diverses techniques d'estimation utilisant la méthode *COSMIC*. Celles-ci sont énoncées dans Vogelezang (2015).

L'utilisation des *FSM patterns* pourrait également servir à produire une telle approximation. De plus, en consignat le tout dans un outil supportant les *FSM patterns*, il serait aisé, par la suite, d'apporter les ajustements à la mesure au fur et à mesure que des précisions sont obtenues pendant l'évolution du projet.

En effet, en utilisant les *FSM patterns* (ceux documentés dans l'article Trudel *et al.* (2016) ou ceux qui seraient définis par l'utilisateur) intégrés dans un outil informatisé, il est aisé d'obtenir rapidement une mesure en ne connaissant que certaines informations. En reprenant l'exemple de la solution du *composite CRUDL-3DG* représenté à la figure 2.1, il est rapide de déterminer que l'ajout d'un module dans un SI formant les processus fonctionnels de « Création », de « Lecture », de « Mise à jour », de « Suppression » et de « Liste » requérant trois groupes de données correspond à une taille de 36 CFP avec peu d'efforts. Par la suite, lors des phases subséquentes du projet, les noms des groupes de données peuvent être modifiés ainsi que les termes utilisés pour nommer le processus fonctionnel afin de refléter le contexte du projet. De plus, lors d'une phase subséquente, si le besoin d'un rapport sur ce nouveau module se fait sentir, il est aisé de mesurer la taille de celui-ci en utilisant le *FSM pattern* intitulé *report-3DG* qui correspond à une taille de 7 CFP (la taille de ce *FSM pattern* est documenté dans le tableau représentant la solution du *pattern* intitulé *multi-composite module-3DG* dans l'article Trudel *et al.* (2016).

À l'aide d'une telle mesure obtenue tôt dans la phase de projet en utilisant des *FSM patterns*, il semble donc possible de fournir une estimation de l'effort nécessaire à sa réalisation. Par le fait même, il semble possible d'estimer l'impact sur le budget de tout ajout tôt dans son évolution (par exemple, l'ajout d'un rapport initialement non prévu lors de la phase de lancement du projet).

### **2.3. Approche proposée**

Dans le but de satisfaire les deux objectifs évoqués ci-haut, nous nous sommes demandé s'il n'existait pas déjà des outils supportant la prise de mesure de la taille fonctionnelle selon *COSMIC* dans lesquels il nous serait possible d'implémenter la fonctionnalité des *FSM patterns*. Pour ce faire, nous avons effectué une revue systématique de la littérature qui est présentée dans le prochain chapitre.

## CHAPITRE 3.

### REVUE SYSTÉMATIQUE DE LA LITTÉRATURE SUR LES OUTILS INFORMATISÉS SUPPORTANT LA PRISE DE MESURE DE LA TAILLE FONCTIONNELLE SELON LA MÉTHODE *COSMIC*

Tel qu'expliqué dans le chapitre 1, la mesure de la taille fonctionnelle avec *COSMIC* nécessite de répertorier les différents mouvements de groupes de données qui sont rattachés aux processus fonctionnels. Ceci implique donc une certaine organisation de cette information nécessaire au calcul de la mesure de la taille fonctionnelle. De ce fait, l'objectif poursuivi dans ce travail est d'identifier les outils existants pour supporter la mesure de la taille fonctionnelle des logiciels, plus spécifiquement, ceux en soutien à la méthode *COSMIC*. Pour ce faire, nous avons effectué une revue systématique de la littérature ciblant les publications scientifiques. Celle-ci a été bonifiée par des recherches sur internet afin d'identifier ce qui est proposé au niveau de l'industrie.

Cette recherche d'outil se voulant systématique, ce chapitre vise à documenter la méthodologie (Kitchenham et Charters, 2007) adoptée pour recenser les différents outils de la mesure de la taille fonctionnelle en *COSMIC* pour ensuite présenter les résultats obtenus. Pour ce faire, le chapitre sera structuré de façon à décrire, en premier lieu, les différents mots clés utilisés pour la recherche ainsi que les critères de sélection des articles. Suivront certaines mesures provenant des résultats des requêtes effectuées sur les sources pertinentes avec les divers mots clés. En plus des mesures, cette recherche permettra d'identifier certains articles ou outils et ainsi de répondre au but premier de cette revue systématique de la littérature, soit d'identifier les outils supportant la mesure de la taille fonctionnelle sous *COSMIC*. Enfin, le chapitre se terminera sur les réponses à nos questions qui sont énoncées dans le prochain paragraphe.

### 3.1. Questions de recherche d'articles

Lors d'une revue systématique de la littérature (RSL), la première étape consiste à énoncer une ou des question(s) afin d'orienter celle-ci. Dans le cas présent, nous l'avons formulée ainsi :

- Qu'existe-t-il comme outil informatisé supportant la prise de mesure de la taille fonctionnelle selon la méthode *COSMIC*?

En effectuant des recherches d'articles en lien avec cette question, il nous a été également possible de répondre aux deux autres sous-questions suivantes motivant cette RSL dans le cadre de ce projet de maîtrise.

La première sous-question est :

- Est-ce possible de faire évoluer ces outils afin d'y implémenter le concept de *functional size measurement patterns*?

La seconde est :

- Existe-t-il des outils implémentant déjà ce concept de *FSM patterns*?

Pour ce faire, un choix de mots clefs pertinents était nécessaire afin de répondre à ces questions, sujet traité dans la section suivante.

### 3.2. Critères de sélection des articles

Une seconde étape nécessaire à une revue systématique de la littérature consiste à sélectionner les critères qui seront utilisés pour former les requêtes de recherche (Kitchenham et Charters, 2007). Afin d'obtenir des résultats les plus exhaustifs possible, nous avons employé une stratégie nécessitant peu de mots clefs et constituant des termes généraux. Ce choix s'explique par le fait que les outils supportant la mesure de la taille fonctionnelle peuvent s'appliquer à plusieurs contextes tels que l'automatisation de la mesure, la prise de mesure standard, l'estimation, etc. Il s'agissait donc de recenser tous les articles proposant des outils indépendamment du contexte de leur utilisation.

Ainsi, pour ne pas se limiter à un contexte et pour répondre de la manière la plus exhaustive possible à notre question de recherche, il a été convenu d'utiliser les mots clefs suivants : *COSMIC*, *Functional Size Measurement (FSM)* et *tool*. Les mots clefs sont définis dans les sections suivantes.

### 3.2.1. *COSMIC*

Même s'il existe plusieurs méthodes permettant de mesurer la taille fonctionnelle, puisque les *patterns* de la mesure ont été initialement définis en *COSMIC*, cette RSL a ciblé spécifiquement la méthode *COSMIC*.

### 3.2.2. *Tool*

Ce mot clef générique permet de rechercher tous les types d'outils employés avec la méthode *COSMIC*. Lors des tests initiaux servant à valider les termes à utiliser, nous avons expérimenté des recherches en utilisant les mots clefs « *application* » ou « *software* ». Cependant, c'est le mot clef « *tool* » qui a permis de recenser le maximum d'articles pertinents.

### 3.2.3. *Functional Size Measurement (FSM)*

Lors des premières recherches sur un échantillon restreint afin de tester les mots clefs sélectionnés, nous avons remarqué que plusieurs articles trouvés correspondaient à des domaines autres que l'informatique. La raison est que le terme « *COSMIC* » fait aussi référence aux cosmos. Ainsi, afin de maximiser les chances de ne recenser que les articles en lien avec le sujet de recherche, il a été nécessaire de circonscrire la recherche en définissant le contexte d'utilisation du terme *COSMIC*. Pour ce faire, le mot clef *Functional Size Measurement* a été utilisé. De plus, lors des recherches préliminaires pour tester les mots clefs, il a été remarqué que l'utilisation de l'acronyme de *Functional Size Measurement*, soit FSM, permettait de recenser des articles supplémentaires. Cependant, dans le cas de FSM, il s'est avéré important de le combiner avec les autres mots clefs, car cet acronyme est également utilisé pour signifier *Finite State Machine*. Donc, pour maximiser la recherche d'articles pertinents, les deux termes ont été utilisés pour former les requêtes de recherche à exécuter sur les différentes bases de données sélectionnées. Les bases de données sélectionnées sont présentées dans la section suivante.

## 3.3. Sélection des bases de données

En plus des critères de recherche, il a été nécessaire de définir les sources de données sur lesquelles lancer nos requêtes. Pour ce faire, la liste des bases de données auxquelles

la bibliothèque de l'UQAM donne accès a été passée en revue afin d'identifier celles susceptibles de permettre de recenser des articles pertinents. L'expérience de Mme Sylvie Trudel, une chercheuse du domaine, a été mise à profit afin de les sélectionner. Les bases de données suivantes ont été retenues :

- *ACM Digital Library*,
- *Digital Scholarly Editions*,
- *ACM - Guide to computing Literature*,
- *Scopus*,
- *ScienceDirect*,
- *Whiley Outline Library - Journals*,
- *IEEE XPLORE* et
- *Springer Link*.

### 3.4. Requête de recherche pour chaque base de données

Pour chacune des bases de données sélectionnées, une double recherche a été effectuée. La première requête était formée des mots clefs : *COSMIC*, *Functional size measurement* et *Tool*. La seconde requête utilisait l'acronyme de *Functional size measurement*, soit les mots clefs : *COSMIC*, FSM et *Tool*. Rappelons que lors des tests préliminaires des mots clefs, nous avons constaté que FSM et *Functional size measurement* ne donnaient pas nécessairement accès aux mêmes articles. Ainsi, pour maximiser les résultats, nous avons décidé de lancer ces deux requêtes. Par ailleurs, puisque les différents outils de recherche diffèrent selon la base de données et que les combinaisons d'opérateurs logiques « Et » et « Ou » ne fonctionnent pas toujours de la même façon, nous avons contourné le problème en exécutant séparément les deux requêtes au lieu de combiner *Functional size measurement* et FSM. Cependant, ceci a eu l'effet de renvoyer plusieurs fois les mêmes articles. Un traitement spécifique expliqué un peu plus loin a été nécessaire afin d'éliminer ces doublons.

De plus, un filtre a été ajouté aux requêtes limitant la recherche aux travaux publiés à partir de 2003 inclusivement. La raison est que la méthode *COSMIC* a subi une refonte importante durant l'année 2002 et que les outils existants avant janvier 2003 ne seraient plus adaptés et donc non pertinents.

Par ailleurs, les mots clefs utilisés pour former les requêtes de recherche ont été formulés en anglais. La raison est que les bases de données interrogées contiennent surtout des

textes dans cette langue. De plus, les textes rédigés dans une autre langue que l'anglais ou le français ne peuvent être retenus, car nous ne les maîtrisons pas assez bien. Nous savons, par expérience, que la méthode *COSMIC* est utilisée en Amérique du Sud et en Chine. Nous pouvons donc émettre l'hypothèse que divers articles aient été écrits en espagnol ou en chinois. Cependant, puisque nous ne pourrions comprendre le contenu rédigé dans ces langues, il est inutile d'effectuer des recherches en utilisant des mots clefs traduits dans celles-ci.

### **3.5. Requête de recherche dans d'autres sources d'information**

La recherche systématique a également été bonifiée par des requêtes dans le moteur de recherche de Google en utilisant les mêmes mots clefs. Le but de cette manœuvre était d'identifier des outils commerciaux existants pouvant supporter la mesure de la taille fonctionnelle en *COSMIC*. En effet, la littérature scientifique fait rarement référence à de tels logiciels. Cependant, cette information provenant de l'industrie était pertinente pour répondre à notre question de recherche.

De plus, nous avons également analysé certaines des références des articles qui nous semblaient prometteuses. Ceci a permis d'élargir la recherche et d'accéder à des sources d'informations supplémentaires à celles récupérées par les diverses requêtes employées sur les bases de données sélectionnées. Les deux prochaines sections présentent les résultats de la sélection.

### **3.6. Résultats de la sélection - Littérature scientifique**

#### **3.6.1. Échantillonnage**

L'exécution des requêtes (faite en décembre 2017) sur les diverses bases de données a permis d'identifier plusieurs articles pertinents pour répondre à notre question de recherche initiatrice de cette RSL. Le tableau B.1 se trouvant à l'annexe B résume le nombre d'articles résultant des requêtes formées des différents mots clefs sur les diverses bases de données sélectionnées. Soulignons que les recherches ont été effectuées avec les possibilités de recherche avancée des bases de données retenues.

L'information du tableau B.1 provient de la consignation de toute l'information trouvée et annotée dans 16 onglets d'un tableur Excel. Cette opération de consignation de

l'information de façon structurée a exigé un investissement en temps considérable. Un aperçu du chiffrier Excel produit peut être apprécié à l'annexe C

De plus, les mots clefs FSM et *Functional size measurement* étant souvent utilisés ensemble dans les articles, les requêtes ont régulièrement donné accès aux mêmes articles. Afin d'éliminer les doublons et éviter de comptabiliser plusieurs fois un même article, les requêtes sur les mêmes bases de données retournant un grand nombre d'articles ont été fusionnées et les doublons ont été retirés en utilisant la fonction prévue à cette fin offerte dans Excel. Plus précisément, ce fut le cas pour les résultats obtenus provenant de Scopus, IEEE XPLORE et Springer link. Le tableau suivant présente le résultat de cette fusion.

Tableau 3.1. – Résultat des requêtes fusionnées

ID	Base de données	NB
SCO-Fusion	Scopus	168
IEEE-Fusion	IEEE XPLORE	255
SPL-Fusion	Springer link	120

En tout, près de 575 articles ont été recensés à l'aide des requêtes lancées sur les différentes bases de données sélectionnées. Ensuite, une sélection manuelle a été effectuée afin de ne garder que les articles pertinents en lien avec notre question de recherche. Celle-ci s'est effectuée en balayant les différentes sections des articles afin d'évaluer la pertinence de l'article relativement aux questions à l'origine de cette RSL. Cette opération a permis d'éliminer près de 80% des articles recensés via l'interrogation des bases de données. Le 20% restant, totalisant 121 articles, a été analysé plus en profondeur dans le but d'en confirmer la pertinence. Cette deuxième étape de sélection a permis de retenir 29 des 121 articles résultant de la première sélection manuelle. Cette deuxième sélection s'est effectuée en lisant les divers articles afin de confirmer qu'ils étaient en lien avec la question de recherche.

Durant cette deuxième vague de sélection d'articles, une carte conceptuelle (*mind map*) a été créée afin de dégager les divers regroupements possibles à effectuer sur les outils permettant la mesure de la taille fonctionnelle sous *COSMIC*. Un aperçu de cette carte conceptuelle est reproduit à la figure D.1 de la page 102 de l'annexe D. Ces regroupements ont par la suite été repris et consignés dans le tableau Excel afin d'aider à la comptabilisation des résultats présentés ci-après. De plus, cette étape a permis de

valider les regroupements et d'ajuster le tout au besoin, par exemple, en fusionnant certains regroupements. Un aperçu de ce tableau contenant tous les articles sélectionnés se trouve en annexe E. Les résultats et les décomptes énoncés ci-après proviennent de ce chiffrier Excel.

### 3.6.2. Interprétation des résultats de la RSL

Le premier constat est que les différents outils décrits dans les articles sélectionnés poursuivent deux objectifs. Le premier objectif correspond à l'automatisation (ou semi-automatisation) de la mesure de la taille fonctionnelle. C'est l'objectif majoritaire recherché par 28 des 29 articles. Le second objectif correspond à aider la saisie directe et structurée via un programme informatique de la taille fonctionnelle en *COSMIC*. En effet, il n'y a que Lavazza et Del Bianco *et al.* (2012) qui traite de ce sujet. Puisqu'il n'y a qu'un seul article qui traite de ce deuxième objectif et puisque ce sujet s'avère important dans le cadre de notre projet, nous avons élargi notre recherche d'outils adressant ceci à ce que l'industrie propose. Ceci est présenté dans la section 2.7 de ce rapport.

L'automatisation de la mesure de la taille fonctionnelle sous *COSMIC* s'avère légitime, car celle-ci cherche à pallier diverses problématiques rencontrées dans la pratique de cet exercice tel que : réduire l'effort nécessaire à la mesure et éviter les erreurs (Trudel *et al.*, 2016).

Parmi les articles retenus concernant l'automatisation, 14 des 29 articles stipulaient explicitement que l'aspect recherché était de produire une estimation de la mesure de la taille fonctionnelle et non une mesure exacte. C'est le cas par exemple des articles de Heričko *et al.* (2006) et de Ungan et Demirörs (2015) pour ne nommer que ceux-ci. Afin d'en consulter la liste complète, se référer aux articles ayant un X à la colonne « Estimation » de l'annexe E.

Diverses approches ont été expérimentées afin de tenter d'automatiser la mesure de la taille fonctionnelle. La modélisation UML compte 12 articles sur les 29, par exemple Bagriyanik et Karahoca (2016) ou Sellami *et al.* (2013) pour ne nommer que ceux-ci. Pour une liste exhaustive, voir les articles ayant un X à la colonne « Modélisation UML » de l'annexe E. En plus de la modélisation UML, d'autres langages de modélisation ont été utilisés. En effet, 2 des 29 articles sont basés sur une modélisation à l'aide de Simulink (Oriou *et al.*, 2014; Soubra *et al.*, 2011). Toujours parmi l'ensemble des articles sélectionnés, 1 article utilise AUTOSAR afin d'automatiser la mesure de la taille fonctionnelle sous *COSMIC* (Soubra *et al.*, 2015). La modélisation utilisant BPM a

également été utilisée dans 3 articles (B. Aysolmaz et Demirors, 2014 ; Banu Aysolmaz et Demirors, 2014 ; Kaya et Demirors, 2011). Enfin, 1 des 29 articles traitait de l'utilisation de la modélisation PRiM (Grau, 2008). Force est de constater que l'automatisation de la mesure de la taille fonctionnelle en *COSMIC*, via une modélisation, constitue 65% des expérimentations et solutions proposées. Selon ces résultats, nous pouvons donc en conclure que c'est la voie la plus populaire depuis 2003.

Ces tentatives d'automatisation, énoncées plus haut, de la mesure de la taille fonctionnelle en *COSMIC* via une modélisation peuvent se résumer comme suit. La modélisation des exigences fonctionnelles est faite sous forme de diagramme en respectant la notation du langage de modélisation utilisé. Ensuite, l'outil implémenté permet l'automatisation en extrayant l'information pertinente via l'analyse du ou des fichier(s) correspondant au(x) diagramme(s). Enfin, cette information est traitée afin d'en extraire les processus fonctionnels, les groupes de données ainsi que les mouvements appliqués à ceux-ci (Entrée, Sortie, Lecture et Écriture). Ceci permet donc de mesurer la taille fonctionnelle en points de fonction *COSMIC*. Bref, les articles fondés sur la modélisation dérivent les règles permettant d'extraire, à partir de diagramme(s), toute l'information nécessaire à la mesure *COSMIC*. Cette méthode d'automatisation de la mesure *COSMIC* nécessite donc la création préalable de ces diagrammes. De plus, la précision de la mesure obtenue sera influencée par le niveau de détail des diagrammes utilisés. C'est sans doute ce qui explique que 11 des 18 articles basés sur la méthode de modélisation stipulent clairement que la méthode décrite dans l'article génère une estimation de la taille fonctionnelle et non une mesure précise.

Au lieu de se baser sur la modélisation, 6 des articles provenant de notre sélection utilisaient des artefacts de l'implémentation du logiciel. Ces articles sont identifiés par un X à la colonne « Artefact Impl. » de l'annexe E.

Dans 4 articles (Akca et Tarhan, 2012, 2013 ; Gonultas et Tarhan, 2015 ; Sag et Tarhan, 2014), la méthode proposée correspondait à ajouter, manuellement ou automatiquement, une portion de code afin que, lors de l'utilisation du logiciel, cet ajout permette d'identifier les processus fonctionnels et les mouvements de données nécessaires au calcul selon *COSMIC*. Cette méthode correspond à une mesure semi-automatisée. En effet, dans un premier temps, la méthode décrite dans les articles stipule qu'il faut ajouter, aux endroits stratégiques dans les sources de l'application, du code spécifique permettant l'identification des prérequis nécessaires au calcul selon *COSMIC*. Cet ajout s'effectue manuellement ou en exécutant un outil conçu spécialement pour cette tâche. Ensuite, un usager de l'application doit exécuter des scénarios d'utilisation correspondant à des

processus fonctionnels en utilisant le logiciel. Enfin, suite à cette exécution du logiciel, la mesure de la taille fonctionnelle est produite. Cette méthode semi-automatisée nécessite donc un logiciel existant, ainsi qu'un accès à son code source. Il est également important que la portion de code à incorporer, permettant la mesure de la taille fonctionnelle en *COSMIC*, soit implémentée aux endroits appropriés. Puisque ceci engendre une altération des sources, le code à ajouter doit être rédigé dans le même langage de programmation que le logiciel. Dans les cas décrits des articles, le langage de programmation utilisé était *Java*. De plus, lors de l'exécution, il est important de s'assurer que tous les scénarios exécutés permettent d'identifier et de mesurer tous les processus fonctionnels de l'application.

Un autre article (Özkan, 2011) décrit une méthode d'automatisation provenant d'artefacts consistant à analyser divers fichiers (de configurations et du code source) d'un système ayant une architecture trois tiers utilisant un *Object Relational Mapping* (ORM). Celle-ci consiste à analyser les fichiers de configuration de l'ORM ainsi que les fichiers représentant le code source (c'est-à-dire une analyse statique du code). Cette opération permet d'identifier les objets du domaine d'affaires, les contrôles graphiques permettant l'interaction avec ces objets et les divers traitements s'effectuant entre ces objets. Toute cette information permet de récupérer tous les prérequis nécessaires à la mesure de la taille fonctionnelle en *COSMIC*. La modification du logiciel est donc requis afin d'en mesurer la taille en CFP. De plus, puisqu'elle est basée sur l'analyse du code afin de repérer les informations permettant de compter les entrées, sorties, lectures et écritures, l'uniformisation de l'implémentation est importante. Il y a donc un fort couplage à la technologie employée pour le développement du logiciel à mesurer. Dans le cas précis de l'article, l'implémentation combinait le C# (logique d'affaires), l'ASP.Net (interfaces utilisateurs) et NHibernate (ORM).

Un dernier article (Li *et al.*, 2003) sur les six articles basés sur les artefacts de l'implémentation logicielle utilisait seulement les fichiers d'interfaces afin de produire la mesure automatiquement. Plus précisément, cette méthode est basée sur l'analyse de fichier *XForms* qui offre une représentation XML de l'interface du logiciel à concevoir. Ce fichier *XForms* est généré à partir d'un diagramme *Visio*. Cette conversion de *Visio* à *XForms* s'effectue à l'aide d'un outil de conversion conçu par les auteurs. Celui-ci produit un fichier *XForms* avec diverses informations complémentaires provenant du fichier *Visio* qui permettront à un autre outil de mesurer la taille en CFP du fichier *XForms*. Dans sa forme native, un fichier de type *XForms* ne contient pas tout le nécessaire pour produire une mesure de la taille fonctionnelle. C'est pour cette raison que la conversion

de *Visio* vers *XForms* bonifie celui-ci avec l'information nécessaire à la mesure. Même s'il est altéré par l'ajout de balises XML, le format *XForms* demeure valide et continue d'accomplir correctement sa tâche principale.

Selon les articles retenus, l'automatisation de la mesure de la taille en *COSMIC*, via des artefacts provenant de l'implémentation logicielle, représente 21% des expérimentations et solutions proposées. Elle est donc la deuxième méthode la plus populaire depuis 2003.

En plus d'une automatisation via la modélisation ou via les artefacts provenant de l'implémentation du logiciel, quatre articles sur les 29 retenus proposent une dernière méthode pour automatiser la mesure de la taille fonctionnelle en *COSMIC*. Ces articles s'intéressent aux textes décrivant les exigences fonctionnelles. En utilisant diverses techniques issues du *Natural Language Processing* (NLP) tels que *supervised test mining* et *unsupervised text clustering technique*, cette méthode propose d'extraire d'un document *Software Requirements Specifications* (SRS) toute l'information nécessaire à la mesure approximative de la taille fonctionnelle sous *COSMIC*. Ces articles sont identifiés par un X à la colonne « Requis textuels » de l'annexe E.

Puisque la mesure est basée sur le texte, celle-ci est donc sujette aux aléas de la prose et de la forme utilisée pour la rédaction. Dans ce cas précis, les articles ne faisaient état que d'expérimentations utilisant des SRS en anglais.

L'automatisation de la mesure de la taille en *COSMIC* basée sur l'analyse du texte correspond à 14% des articles sélectionnés.

### 3.7. Résultats de la sélection - Industrie

En plus d'interroger les diverses bases de données scientifiques sélectionnées, nous avons élargi nos recherches afin d'identifier les outils commerciaux en relation avec la question et les sous-questions énoncées dans la section « Questions de recherche d'articles ».

Cette recherche a été motivée par le fait qu'un seul article (Lavazza et Bianco *et al.*, 2012) décrivait un outil permettant l'entrée de mesures dans le but de structurer celle-ci. De plus, parmi les 575 articles trouvés en littérature scientifique, certains énonçaient l'existence d'outils commerciaux. Cependant, lorsque nous tentions de rechercher ces outils afin de récupérer plus d'informations, ceux-ci s'avéraient ne plus exister. Vu cet échec à obtenir une information pertinente sur les outils permettant une entrée directe de

la mesure en *COSMIC*, nous avons décidé d'approfondir nos recherches en interrogeant le moteur de recherche de Google<sup>1</sup> afin de débusquer ces outils commerciaux.

Ainsi, similairement à la démarche effectuée pour la recherche d'articles pertinents dans les bases de données scientifiques sélectionnées, nous avons utilisé les mots clefs décrits dans la section « Critères de sélection des articles » avec Google. Les sites qui nous semblaient les plus prometteurs contenus dans les cinq premières pages retournées par le moteur de recherche ont été consultés. Cette opération a permis d'identifier d'autres sites web commerciaux pertinents. C'est le cas par exemple du site de *International Software Benchmarking Standards Group* (ISBSG). En effet, le contenu d'une des pages de leur site fournit divers produits conçus pour aider l'estimation de projets à l'aide de la mesure de la taille fonctionnelle.

Cette recherche au niveau des produits existant dans l'industrie nous a permis de constater que les outils de mesure de la taille fonctionnelle les plus aboutis implémentaient *IFPUG* et non *COSMIC*. Par exemple, c'est le cas de l'outil « SCOPE » proposé par *Total Metrics*<sup>2</sup> ou *Metrics Quest*.<sup>3</sup> Dans le cas de ce dernier, celui-ci utilise par défaut la méthode *NESMA 2.2*, mais supporte également *IFPUG*.

De plus, certains de ces outils, même si basés sur *IFPUG*, permettent d'effectuer des mesures selon d'autres méthodes, entre autres *COSMIC*. C'est le cas de l'outil « Seer » offert par la compagnie *Galorath*.<sup>4</sup>

Ceci a également permis d'identifier deux anciens projets *open sources* tels que *Functional Point Analysis GUI Tool to help count FPs for your system*<sup>5</sup> ou *Open function point calculator*.<sup>6</sup> Ces découvertes nous ont donné l'idée d'effectuer une recherche sur le site *GitHub* afin d'identifier des projets du genre plus récents et peut-être même toujours actifs. En utilisant les termes *COSMIC* et *Functional Size Measurement*, la recherche sur *GitHub* fut infructueuse. L'utilisation du nom de la méthode, soit *COSMIC*, n'a pas non plus permis de récupérer des projets pertinents, contrairement à ce qui survient avec les autres noms des méthodes. Par exemple, en utilisant le mot *IFPUG*, il a été possible d'identifier quatre dépôts *Git* :

- 
1. <https://www.google.com>
  2. <https://bit.ly/2HPfXGD>
  3. <https://www.metricsquest.com/agile-estimating/>
  4. <http://galorath.com>
  5. <https://bit.ly/2Mp9Tsv>
  6. <https://sourceforge.net/projects/ofpc/reviews?source=navbar>

1. leftpudding/Function-Point-Spreadsheet<sup>7</sup>, dernière mise à jour 28 juillet 2015
2. krisaore/ifpug<sup>8</sup>, dernière mise à jour 7 mai 2017
3. chenclouing/IFPUG<sup>9</sup>, dernière mise à jour 8 mars 2015
4. deliaqi/myIFPUG<sup>10</sup>, dernière mise à jour 6 juin 2017

Notre recherche nous a également permis de découvrir des outils simplistes et gratuits utilisables directement à partir d'une page web. C'est le cas de *Function Point Calculator*<sup>11</sup> et de *FP Fast Calculator*.<sup>12</sup> Dans le cas du second outil, *4sumpartners* offre également un outil professionnel nommé « Experience » supportant partiellement la mesure de la taille fonctionnelle selon la méthode *COSMIC*. Plus particulièrement, selon l'information pouvant être récoltée du site de la compagnie,<sup>13</sup> le logiciel supporte complètement *FiSMA* 1.1, largement *NESMA* et *IFPUG* et partiellement *COSMIC* et *MkII*.

Enfin, en lien direct avec la méthode *COSMIC* choisie pour ce projet, cet exercice nous a tout de même permis d'identifier un outil autre que ceux déjà présentés permettant le support de la mesure de la taille fonctionnelle. Cet outil se nomme « Visual FSM » et est offert par la compagnie *Pentad-SE* au Royaume-Uni. Fait intéressant, en plus d'être un des rares outils permettant la prise de la mesure selon *COSMIC*, celui-ci est le seul que nous ayons trouvé implémentant les *patterns* de la mesure de la taille fonctionnelle, et qui fut ajouté après que notre directrice de projet ait partagé au président de cette société le concept de *pattern* lors d'une discussion privée. Nous avons contacté, en vain, le président de cette compagnie avec l'espoir que celui-ci voudrait collaborer à notre projet.

### 3.8. Conclusion de la RSL

En prenant en considération l'information récoltée dans les sections précédentes documentant le résultat de la RSL ainsi que les résultats provenant de la recherche des outils commerciaux, il est possible de répondre comme suit aux questions ayant servi de prémisse aux recherches.

---

7. <https://github.com/leftpudding/Function-Point-Spreadsheet>

8. <https://github.com/krisaore/ifpug>

9. <https://github.com/chenclouing/IFPUG>

10. <https://github.com/deliaqi/myIFPUG>

11. <http://groups.umd.umich.edu/cis/course.des/cis375/projects/fp99/table.html>

12. <http://www.4sumpartners.com/fpfastcalculator/>

13. <http://www.4sumpartners.com/experience@/experience@-service/>

### 3.8.1. Qu'existe-t-il comme outil informatisé supportant la prise de mesure de la taille fonctionnelle selon la méthode *COSMIC* ?

La RSL a permis d'identifier que les outils développés par la communauté scientifique se concentraient principalement sur l'automatisation de la mesure de la taille fonctionnelle selon *COSMIC*. Cette automatisation s'effectue soit via la modélisation, les artefacts de l'implémentation logicielle ou les textes décrivant les exigences fonctionnelles. Sur les 29 articles pertinents répertoriés, seul Lavazza et Bianco *et al.* (2012) traitait d'une application permettant la saisie directe de la mesure afin de structurer celle-ci. Par ailleurs, ces outils ne semblaient pas être accessibles.

Au niveau de la recherche des outils commerciaux, les outils les plus matures permettant la saisie de l'information nécessaire au calcul de la taille fonctionnelle implémentent la méthode *IFPUG* au détriment de *COSMIC*. Dans certains cas, le support de *COSMIC* n'est que partiel. Nous avons également trouvé des sites web minimalistes permettant de fournir un calcul de la taille fonctionnelle à partir des entrées de données. Cependant, ceux-ci ne peuvent servir que pour fin de calcul sommaire, car ils n'offrent pas de persistance de l'information. Nous avons élargi notre recherche à *GitHub* dans l'espoir de trouver des projets *open sources*, mais le constat demeure le même. Le seul outil commercial se démarquant se nomme « Visual FSM » de la compagnie *Pentad-SE*.

### 3.8.2. Est-ce possible de faire évoluer ces outils afin d'y implémenter le concept de *FSM pattern* ?

Concernant les outils décrits dans les articles sélectionnés provenant de la RSL, il serait plutôt inutile d'implémenter le concept de *FSM pattern* dans ceux en lien avec l'automatisation. Par ailleurs, l'outil décrit dans Lavazza et Del Bianco *et al.* (2012) serait un candidat potentiel. En effet, celui-ci permet la saisie des données nécessaires à la prise de mesure en *COSMIC*. Il serait donc possible d'implémenter le concept de *FSM pattern* afin que le système propose d'entrer l'information nécessaire à la mesure des CFP via les *patterns* et affiche le tout prérempli à l'écran à l'utilisateur. Cet outil permettrait également aux utilisateurs de créer leurs propres *FSM patterns* puis de les réutiliser par la suite. Cependant, n'ayant pas accès au code source de l'outil, cet ajout devrait s'effectuer par les auteurs de l'article.

Du côté des outils commerciaux, il pourrait être avantageux pour les sociétés propriétaires d'implémenter le concept de *FSM pattern*. Cependant, pour maximiser l'utilisation

de ce concept, celui-ci devrait préalablement être adapté à la méthode *IFPUG* puisque c'est cette méthode que supportent majoritairement les outils commerciaux.

### **3.8.3. Existe-t-il des outils implémentant déjà ce concept de *FSM pattern* ?**

Tel qu'énoncé précédemment, le seul outil implémentant le concept de *FSM pattern* est l'outil commercial « Visual FSM » de la compagnie *Pentad-SE*.

Cet outil étant une application *Windows*, nous avons tenté de rejoindre le créateur de celui-ci afin de lui offrir de convertir son outil en application web. Nous n'avons pas eu de retour. Nous pouvons donc supposer que ce dernier n'est pas intéressé à moderniser son application.

Notre requête auprès du président de *Pentad-SE* demeurant lettre morte et suite à la prise de conscience résultant de nos recherches qu'il existe peu d'outils permettant la saisie de la taille fonctionnelle selon *COSMIC*, nous avons décidé de démarrer un projet *open source* offrant cette fonctionnalité. En plus de permettre une saisie structurée de la mesure de la taille fonctionnelle, cet outil implémentera également le concept des *FSM patterns* ce qui nous permettra de valider nos idées de comment ce principe devrait être implémenté dans une application web.

### **3.8.4. Constat**

La RSL nous a permis de constater qu'il n'existait pas d'outil dans lequel il nous serait possible d'implémenter la fonctionnalité des *FSM patterns*. À la lumière des résultats obtenus, nous avons décidé de démarrer le développement d'un outil supportant les *FSM patterns* puisque nous n'avons rien trouvé qui permettait d'incorporer ceux-ci dans un outil déjà existant et libre de droits. De plus, la création d'un tel outil était nécessaire pour satisfaire deux des objectifs poursuivis par les *FSM patterns*, soit « réduire l'effort de la mesure » et « obtenir la taille en CFP dans les premières étapes d'un projet » repris en tant qu'objectifs de notre problématique.

## CHAPITRE 4.

### MÉTHODOLOGIE DE DÉVELOPPEMENT D'UN OUTIL SUPPORTANT LES *FSM PATTERNS*

La réalisation de ce projet de synthèse repose sur la méthodologie décrite dans le présent chapitre. Celle-ci peut être résumée par la figure suivante.

	« Brainstorm » sur les besoins fonctionnels nécessaires à la mesure en COSMIC	Analyse des besoins	Carnet de produit sous forme de « User stories »	
			Maquettes d'interfaces	
Démarrage du développement d'un outil supportant les « FSM patterns »	Technologies disponibles	Choix de la technologie	Technologies retenues	Informations et choix préalables pour débuter le développement
	Architectures disponibles	Choix de l'architecture	Architecture retenue	
	Hébergements disponibles	Choix de l'hébergement	Hébergement retenu	
	Structure de bases de données disponibles	Choix du modèle de données	Modèle de données retenu	
	SGDB disponibles	Choix du SGDB	SGDB retenu	
	Carnet de produit	Planifier le développement	Tableau Kanban	
Développement itératif d'un prototype	Carnet de produit Maquettes d'interface Choix technologique, architecture, modèle de données et SGDB	Développer l'outil	Incrément de l'outil	Prototype outil de mesure implémentant les « FSM patterns »

FIGURE 4.1. – Aperçu de la méthodologie

Ce qui suit décrit les phases ainsi que les activités contenues dans celle-ci.

#### 4.1. Démarrage du développement d'un outil supportant les *FSM patterns*

Suite au constat qu'il n'y avait pas d'outil préexistant auquel nous pourrions ajouter le concept de *FSM pattern*, nous avons décidé de développer notre propre outil implémentant ce nouveau concept. Ce qui suit décrit les activités qui ont été nécessaires au développement de cet outil.

##### 4.1.1. Analyse des besoins

L'une des premières étapes du développement de cet outil a été de clarifier les besoins et fonctionnalités à implémenter. Ceci a eu lieu lors d'une réunion avec la *Product Owner* (PO), Mme Trudel, considérée comme étant l'itération 0 du lancement du projet. Lors de cette rencontre, il a été décidé que l'outil devait être accessible via le web sans

nécessiter d'installation. De plus, l'outil devait être utilisable, peu importe le système d'exploitation de l'utilisateur ; une application web était donc le choix qui s'imposait.

De plus, étant dans un cadre académique et voulant assurer une certaine pérennité de l'outil, l'idée a été lancée de créer celui-ci en tant que projet libre de droits (*open source*). Ainsi, le développement futur pourrait bénéficier de la communauté *open source* et même être poursuivi par des étudiants lors de travaux pratiques à effectuer.

Également lors de cette rencontre, les fonctionnalités principales ont été énoncées, expliquées et priorisées. Un premier *backlog* sous forme de *post-it* a été utilisé pour consigner les idées. C'est aussi lors de cette phase qu'un nom a été donné à l'outil : *COSMIC-Sizing-Tool*.

Une fois les idées clarifiées, le premier *backlog* sous forme de *post-it* a été transcrit dans le système informatisé rendu disponible par la plateforme *GitHub*.

En plus d'avoir permis l'élicitation des fonctionnalités souhaitées de l'application, ce *backlog* a également servi d'exercice de mesure de la taille fonctionnelle selon *COSMIC* dans le cours MGL7240 Mesures et développement de logiciel. En effet, durant ce cours, la méthode *COSMIC* a été enseignée et les étudiants à la maîtrise en génie logiciel ont eu comme exercice de mesurer l'outil à implémenter en se basant sur les *user stories* décrites.

Une copie de la dernière version de ce *backlog* peut être récupérée à l'annexe F. Pour les contributeurs du projet ayant accès au dépôt *GitHub* du projet, le *backlog* peut être consulté en ligne à l'adresse : <https://waffle.io/cosmic-sizing-tool/cosmic-sizing-tool>.

#### **4.1.2. Choix de la technologie**

Le développement logiciel implique obligatoirement divers choix technologiques permettant la mise au monde d'un nouvel outil. Ces choix sont importants, car ils ont un impact tout au long de la durée de vie du logiciel ainsi créé. Cette section présente les différents choix technologiques qui ont été effectués pour développer l'outil *COSMIC-Sizing-Tool*.

##### **4.1.2.1. Langages de programmation**

Un des choix technologiques ayant un impact majeur sur la réalisation d'un projet de développement concerne le/les langages(s) de programmation utilisé(s) pour implémenter

ter la solution. Dans le cas présent, plusieurs contraintes devaient être prises en compte afin de respecter certains besoins mentionnés lors de la phase d'analyse.

Premièrement, les langages à utiliser devaient être gratuits. En effet, nous n'avions aucun budget pour l'achat de licence. De plus, le projet se voulant *open source* et l'évolution de l'outil pouvant se poursuivre dans un cadre académique, le fait d'utiliser un langage de programmation ne nécessitant pas de licence permet de s'assurer que personne ne sera rebuté par le fait de devoir acheter une licence pour participer au projet.

Deuxièmement, lors de l'analyse des besoins, il a été clairement identifié que le développement du *COSMIC-Sizing-Tool* se devait d'être offert en tant qu'application web. Les langages sélectionnés devaient donc être adaptés au développement d'applications web. Ceci est particulièrement le cas pour tout ce qui correspond à la programmation des interfaces utilisateurs.

Tenant compte de ces contraintes, une première tentative (voir « planifier le développement » ci-après pour les détails) de développement a été effectuée en utilisant le langage *Java*<sup>1</sup> et le *framework Play*<sup>2</sup>. Selon le site web, ce *framework* d'application web utilisant le langage *Java* et *Scala* permet de bâtir des applications web modernes. Il est basé sur une architecture « sans état » (*stateless*) nécessitant un minimum de ressources permettant l'extensibilité (*scalability*) des applications.

L'hypothèse émise était que les étudiants connaîtraient déjà le langage *Java*. Ainsi, une prise en main rapide de cette technologie serait envisagée et permettrait ainsi de se concentrer sur l'implémentation des fonctionnalités. De plus, ce choix a été conseillé par Alexandre Viau de la Maison du Logiciel libre. Par ailleurs, cette personne a également joué le rôle de conseiller-expert et d'intégrateur lors de cette première tentative ayant lieu dans le cadre d'un projet de développement avec des étudiants du baccalauréat en informatique dans du cours INF6150 Conduite de projets informatiques.

Avec cette première tentative aux résultats mitigés, il a été décidé de réduire la portée et de débiter seulement par le développement d'un prototype permettant la saisie de la mesure et supportant le concept de *FSM pattern*. Lors de cette deuxième tentative, les choix technologiques ont été révisés, toujours en considérant les contraintes énoncées en début de section. En effet, puisqu'il s'agissait d'un prototype, nous voulions que le développement soit le plus indépendant possible afin de pouvoir changer sans trop d'impact l'interface graphique ou la logique d'affaires. Notre choix final s'est donc arrêté

---

1. <https://www.java.com/>

2. <https://www.playframework.com>

sur les langages de programmation et *frameworks* suivants : *Python*, *React*, *Redux* et autres, décrits ci-après.

**4.1.2.1.1. Python** *Python*<sup>3</sup> est un langage de programmation interprété multiplateforme ayant pour objectif d'offrir une meilleure productivité grâce à sa syntaxe simple proche du langage naturel. Fondée sur cette promesse d'un langage de programmation productif, la création du prototype devrait donc se faire plus rapidement que si le tout était programmé en *Java*.

De plus, concernant la maintenabilité et l'évolution du *COSMIC-Sizing-Tool*, le choix du langage *Python* s'avère plus judicieux. Selon *Google trends*,<sup>4</sup> qui permet d'identifier les tendances des recherches effectuées sur un sujet donné,<sup>5</sup> le langage *Java* est en perte de popularité depuis les 10 dernières années au Canada. De plus, dernièrement, le langage *Python* est plus populaire au niveau des recherches que le *Java*. Ceci est illustré par la figure suivante tirée d'une recherche sur *Google trends* :

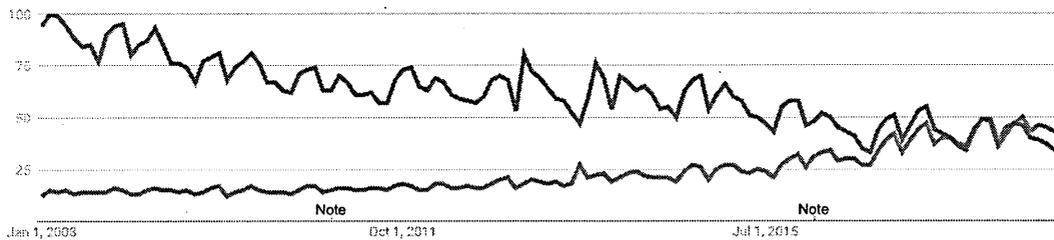


FIGURE 4.2. – Tendence des recherches *Python* (en bleu) versus *Java* (en rouge)

Selon les informations données sur le site *Google trends*, les valeurs vont de 100 à 0 : 100 indiquant un terme de recherche très populaire, 50 un terme de recherche moyennement populaire et 0 indique qu'il n'y a pas assez de données pour ce terme (traduction libre).

**4.1.2.1.2. React** *React*<sup>6</sup> est une bibliothèque *JavaScript* spécialisée pour le développement d'interfaces utilisateurs. La particularité de celle-ci est qu'elle est orientée *component* au lieu de *template*. Ainsi, le développement s'effectue en créant une hiérarchie de composants s'encapsulant les uns dans les autres. La programmation s'effectue idéalement à l'aide du *JSX*, un langage à la syntaxe XML générant ensuite le *JavaScript*

3. <https://www.python.org>

4. <https://trends.google.com>

5. [https://support.google.com/trends/answer/6248105?hl=en&ref\\_topic=6248052](https://support.google.com/trends/answer/6248105?hl=en&ref_topic=6248052)

6. <https://reactjs.org>

qui est interprété par le navigateur permettant le rendu visuel et l'exécution des actions implémentées.

Similairement au choix du langage *Python*, il a été décidé d'utiliser le *framework JavaScript React* dans un contexte de pérennité de l'application. En effet, la figure suivante tirée de *Google trends* démontre bien l'intérêt grandissant au Canada pour ce *framework*. Pour fin de comparaison, la figure affiche également la tendance du populaire *framework Angular* qui est un compétiteur de *React*.

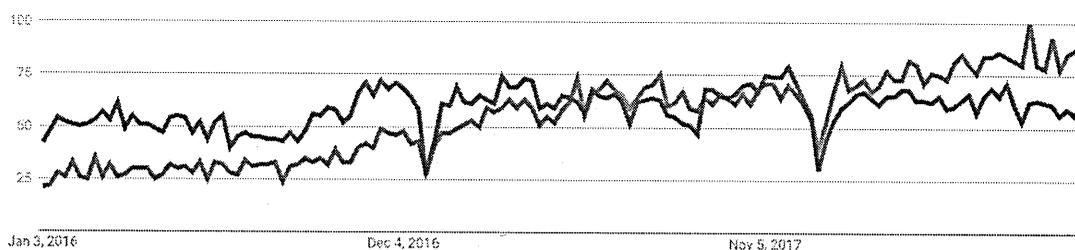


FIGURE 4.3. – Tendance des recherches *React* (en bleu) versus *Angular* (en rouge)

**4.1.2.1.3. Redux** *Redux*<sup>7</sup> est une bibliothèque *JavaScript* couramment utilisée pour les applications complexes développées avec *React* afin de mieux générer l'état des composants. En effet, celui-ci sert de *container* aux composants *React* afin « de tenter de rendre prédictibles les changements d'état en imposant certaines restrictions sur comment et quand une mise à jour peut avoir lieu » (traduction libre, tirée de <https://redux.js.org/introduction/motivation>).

Par ailleurs, celle-ci n'est pas limitée à être utilisée qu'avec *React*. Elle peut également être utilisée avec d'autres *frameworks* tels qu'*Angular* afin d'effectuer une meilleure gestion de l'état.

**4.1.2.1.4. Autres** Les trois technologies décrites ci-haut représentent celles majeures en lien avec la programmation. Cependant, le développement d'application web nécessite également certains incontournables tels que le *HTML*, *CSS* et *JavaScript*.

7. <https://redux.js.org>

#### 4.1.2.2. Composant graphique

La première tentative du *COSMIC-Sizing-Tool*, effectuée par des élèves du baccalauréat en informatique, nous a permis d'identifier certaines complexités. En effet, l'interface de saisie représentait un défi important vu le niveau de complexité à implémenter afin de respecter les exigences émises par la *PO* concernant la saisie de la mesure.

Dans le but d'éviter de refaire ce qui a déjà été fait, une recherche de contrôles de grilles déjà existants a eu lieu. Ceci a permis de découvrir plusieurs contrôles de grilles :

- jsGrid : <http://js-grid.com> ;
- SlickGrid : <https://github.com/mleibman/SlickGrid> ;
- FancyGrid : <https://www.fancygrid.com> ;
- Guriddo jqGrid : <http://www.guriddo.net> ;
- jqxGrid : <https://www.jqwidgets.com/react/react-grid/index.htm> ;
- w2ui Grid : <http://w2ui.com/web/demo/grid> ;
- dhtmlxGrid : <https://dhtmlx.com/docs/products/dhtmlxGrid/> ;
- appendGrid : <http://appendgrid.apphb.com> ;
- Material Design Data tables : <https://material.io/design/components/data-tables.html> ;
- UI Grid : <http://ui-grid.info> ;
- ag Grid : <https://www.ag-grid.com>.

Afin de sélectionner le contrôle de grille le plus adéquat, les diverses fonctionnalités exprimées par la *PO* ont été converties en critères de sélection que voici :

1. Supporter le multi-langues (*internationalization*) ;
2. Permettre les sous-grilles ;
3. Offrir la possibilité du *drag and drop* ;
4. Permettre l'édition du contenu des lignes formant la grille ;
5. Avoir l'autocomplétion ;
6. Pouvoir naviguer dans la grille à l'aide du clavier ;
7. Ajouter des raccourcis clavier personnalisés ;
8. Rendre possible l'ajout d'autres contrôles dans la grille (ex. liste de choix) ;
9. Pouvoir ajuster le rendu visuel ;
10. Permettre l'ajout de nouvelles lignes dans n'importe quelle position de la grille ;
11. Être compatible avec *React*.

Les différents contrôles de grilles précédemment listés ont ensuite été évalués en tenant compte des critères de sélection émanant des exigences fonctionnelles de la *PO*. De cette évaluation, deux contrôles de grilles semblaient prometteurs, soit *dHtmlxGrid* et *jqxGrid*. Afin de départager celui à utiliser, quelques prototypes simples ont été réalisés afin de tester les fonctionnalités des deux grilles. Il en est ressorti que le contrôle de grille répondant le mieux aux besoins était *jqxGrid* de *jqWidgets*. De plus, *jqWidgets* offre également une panoplie d'autres contrôles graphiques web compatibles avec *React*<sup>8</sup> qui pourront être utilisés pour simplifier la création de l'interface utilisateur.

#### 4.1.2.3. Modèle de données et SGBD

Comme la plupart des SI, le *COSMIC-Sizing-Tool* nécessite la persistance des données saisies. Pour ce faire, une modélisation de la structure des données est préalable à la conception de la base de données qui permettra de contenir les données saisies. Pour ce faire, il existe plusieurs types de modèles de base de données, notamment :

- Modèle relationnel ;
- Modèle hiérarchique ;
- Modèle orienté objet ;
- Modèle entité attribut-valeur.

Un choix s'imposait donc pour modéliser la base de données. Puisque la structure sous forme de table, caractérisée par le modèle relationnel ainsi que la normalisation de l'information, s'avérait importante, le choix s'est arrêté sur une modélisation relationnelle de la base de données. De plus, c'est dans ce type de modélisation que nous avons le plus d'expérience. Même si les modèles de type *NoSQL* deviennent de plus en plus présents avec l'arrivée du *Cloud*, l'intérêt des bases de données relationnelles demeure et reste un choix éclairé pour le développement logiciel. Ceci peut être représenté par la figure 4.4 (page 38) tirée de *Google trends*.

Ce choix a également restreint la sélection des systèmes de gestion de base de données à ceux relationnels. Le site *db-engines.com* liste les SGBD relationnels les plus populaires. Ceux-ci sont représentés dans la figure 4.5 (page 38).

La méthode utilisée pour calculer le rang de chacun peut être trouvée à partir du site [https://db-engines.com/en/ranking\\_definition](https://db-engines.com/en/ranking_definition).

---

8. <https://www.jqwidgets.com/react/>

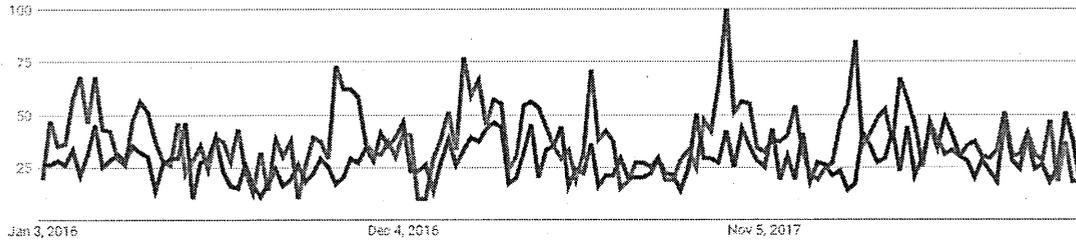


FIGURE 4.4. – Tendence des recherches modèle relationnel (en bleu) versus *NoSQL* (en rouge)

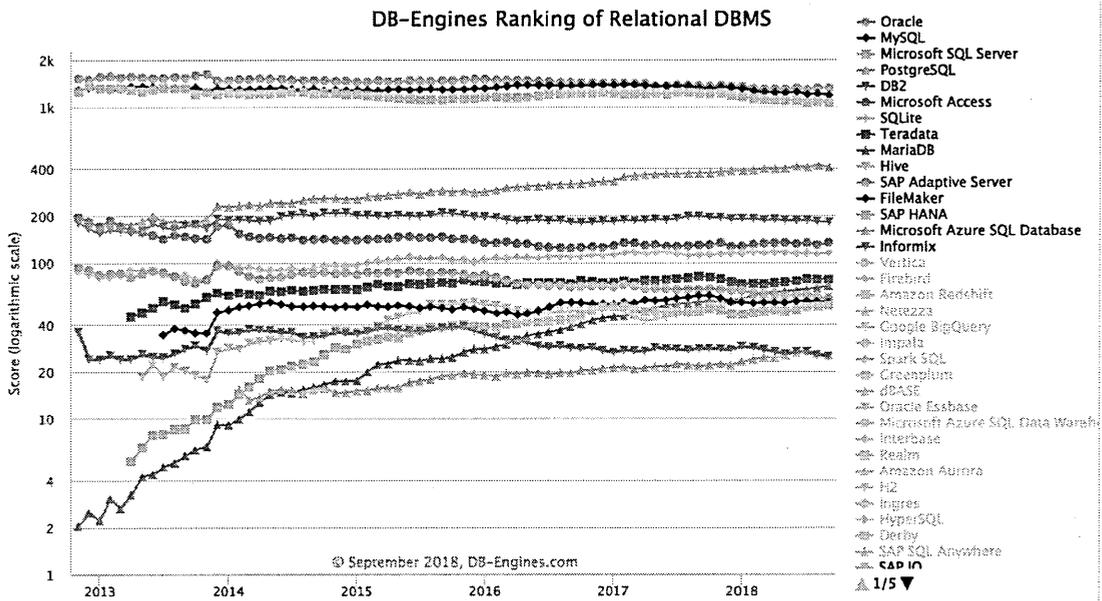


FIGURE 4.5. – Tendence des SGBD relationnels

Comme on peut le constater en consultant la figure précédente, le « top 5 » est formé de : *Oracle*, *MySQL*, *Microsoft SQL Server*, *PostgreSQL* et *DB2*. Puisque l'application doit être *open source*, les SGBD ayant des licences gratuites mais limitatives ont été exclus. Le choix s'est donc fait entre *MySQL* et *PostgreSQL*. En final, c'est *PostgreSQL* qui a été sélectionné, car, suite à diverses recherches, celui-ci semblait le plus recommandé.

Par ailleurs, afin d'accélérer le développement, même s'il a été choisi d'utiliser *PostgreSQL* comme SGBD pour développer *COSMIC-Sizing-Tool*, le prototype a été réalisé avec *SQLite*<sup>9</sup> et avec l'ORM *SQLAlchemy*.<sup>10</sup> En effet, *SQLite* est inclus avec *Python*. Il n'est donc pas nécessaire d'effectuer une phase d'installation et de configuration avant de pouvoir l'utiliser, ce qui n'est pas le cas pour *PostgreSQL*. De plus, l'avantage d'utiliser un ORM tel que *SQLAlchemy* est que celui-ci est compatible avec *PostgreSQL* et *MySQL*. Ainsi, lorsque le temps sera venu de transférer le tout en production sur une base de données *PostgreSQL*, il suffira de changer la configuration du connecteur à la base de données. Vu sa compatibilité avec *MySQL*, il serait donc possible dans le futur de changer de *PostgreSQL* vers *MySQL* sans trop d'impact. Par exemple, ceci pourrait être nécessaire dans le cas où l'outil devrait être installé chez un hébergeur n'offrant pas d'instance de *PostgreSQL*, mais seulement de *MySQL*.

#### 4.1.3. Choix de l'architecture

Concernant la première tentative effectuée par les élèves du baccalauréat à l'UQAM, l'architecture *Model-View-Controller* (MVC) s'est imposée. En effet, c'est sur cette architecture qu'est fondé le *framework Play*. En voulant utiliser la puissance offerte par ce *framework*, nous étions contraints de construire l'application en nous appuyant sur l'architecture prescrite par le *framework*.

Cependant, pour la deuxième tentative, la création du prototype laissait plus de latitude sur l'architecture du système. Même s'il s'agissait d'un prototype, nous voulions être en mesure d'offrir une base qu'il serait possible de faire évoluer afin d'en faire un produit final, en tout ou en partie. De plus, l'un des souhaits énoncés lors de la phase de prise de besoins stipulait qu'il serait utile de rendre les fonctions de gestion des *FSM patterns* accessibles à tous afin que d'autres puissent les utiliser dans leur propre outil. Il a donc été décidé que le développement devait se faire de façon la plus modulaire possible nécessitant un faible couplage entre la couche de présentation et la couche d'affaires

---

9. <https://www.sqlite.org>

10. <https://www.sqlalchemy.org>

afin de rendre interchangeable les deux couches. Ceci permettra à d'autres développeurs d'utiliser le principe de *FSM pattern* dans leurs outils.

Ainsi, le développement de l'interface utilisateur se devait d'être complètement indépendant de la logique d'affaires. Il a donc été décidé de construire le prototype en deux sous-projets distincts. Le premier sous-projet, correspondant à l'interface utilisateur, s'appuierait sur les technologies *React* et *Redux* employant des contrôles graphiques offerts par *jQuery Widgets*, alors que le second sous-projet contiendrait toute la logique d'affaires accessible via un *REST API*. En rendant publiques les fonctions de l'*API*, il serait ainsi possible à d'autres projets logiciels de consommer l'*API* afin d'utiliser, à leur manière, le concept de *FSM pattern*.

Pour concevoir un tel *REST API*, *Python* offre le micro web *framework Flask*<sup>11</sup>. Celui-ci contient une fonctionnalité permettant de facilement répartir les différentes requêtes *RESTful* en plus de fournir un serveur de développement et un *debugger* permettant de rapidement tester le tout sans l'installation d'un serveur dédié. Le déploiement demeure donc minimal et permet de se concentrer rapidement sur l'effort d'implémentation. Cependant, ce serveur ne doit être utilisé que pour le développement, car il s'avère non performant pour une utilisation en production, si on se fie à ce qui est mentionné dans la documentation.

#### 4.1.4. Choix de l'hébergement

L'hébergement s'avère également important lors du développement d'un logiciel, particulièrement pour une application web. En effet, pour que celle-ci soit accessible en ligne, le logiciel doit être hébergé. N'ayant pas la logistique nécessaire ni le budget pour héberger *COSMIC-Sizing-Tool* sur nos propres serveurs, la décision a été prise d'héberger ce dernier sur un serveur externe. Les plus populaires tels que *Google Cloud Platform*, *Amazon Web Services* et *Heroku* ont été envisagés. Chacune de ces plateformes offre des périodes d'essais gratuites ou bien un hébergement avec des ressources limitées satisfaisantes pour le moment. Lorsque viendra le temps de mettre le tout en production, il serait possible d'étendre le déploiement afin d'utiliser les ressources nécessaires à l'exécution adéquate de l'outil. En tentant d'évaluer le meilleur hébergeur, nous nous sommes vite rendu compte qu'il nous était difficile de départager toute l'information disponible par ceux-ci. Le choix s'est donc fait sur la base d'une recommandation provenant de Yvan Ross M.Ing enseignant en génie logiciel à l'ÉTS. À titre de consultant, M. Ross a

---

11. <http://flask.pocoo.org>

fait divers projets de développement hébergés sur *Heroku* et il connaît bien la plateforme. En cas de difficulté, celui-ci pourra donc servir de personne-ressource.

Concernant la deuxième tentative, le choix de l'hébergement ne s'applique pas, car, puisqu'il s'agit d'un prototype, celui-ci ne sera fonctionnel que sur l'environnement de développement du développeur.

En plus d'héberger l'outil, nous devons également trouver un emplacement pour loger les sources. En effet, afin d'en faire un projet *open source*, nous devons rendre accessible le code source du projet tout en nous dotant d'un mécanisme de gestion des versions de celui-ci. Pour ce faire, nous avons choisi la populaire plateforme *GitHub*.<sup>12</sup> En effet, nous avons déjà de l'expérience de gestion de source avec celle-ci. De plus, cette plateforme s'avère être plus qu'un gestionnaire de source, mais plutôt une plateforme supportant le développement de logiciels. En effet, *GitHub* contient tout le nécessaire pour la revue de code, la gestion de projet, l'intégration, la gestion d'équipe et la documentation.

#### 4.1.5. Planifier le développement

En génie logiciel, il existe plusieurs types de modèles pour représenter le cycle de développement, par exemple en cascade, en V, itératif, etc. Le choix du modèle a une influence sur la planification. Dans le cas présent, basé sur les expériences résultant de succès des personnes participant à la planification du développement, le cycle itératif a été choisi. Pour ce faire, une approche agile (Beck *et al.*, 2001) a été utilisée, plus précisément la méthode *Kanban* (Anderson, 2010).

Pour ce faire, le dépôt *GitHub* du *COSMIC-Sizing-Tool* a été lié à *waffle.io*.<sup>13</sup> Ceci a permis de convertir les *issues* rattachées au dépôt dans *GitHub*, correspondant au *backlog*, en tableau *Kanban*. Ainsi, il s'avère très simple de visualiser l'état de chaque *user story* et leur assignation. Les différents états possibles sont paramétrés en tant que colonnes du tableau *Kanban* :

- *Backlog* pour les éléments non commencés et sans être nécessairement priorisés ;
- *Ready* pour des éléments priorisés et prioritaires dont l'analyse détaillée a été complétée ;
- *In Progress* pour les éléments en cours de réalisation ;

---

12. <https://github.com>

13. <https://waffle.io>

- *In Review* pour les éléments dont le développement est terminé, mais qui doivent être approuvés par la PO ;
- *Done* pour les éléments terminés et approuvés par la PO.

## 4.2. Développement itératif

Une fois la phase « Démarrage du développement d'un outil supportant les *FSM patterns* » complétée, ce qui a permis de statuer sur les technologies, l'architecture et l'hébergement, il ne restait plus qu'à s'attaquer à la prochaine phase, soit le développement itératif de l'outil.

Tel que déjà énoncé sommairement dans les sections précédentes, il y a eu deux tentatives au développement. La première a consisté à faire réaliser certaines fonctionnalités par un groupe d'étudiants lors d'une seule journée intensive afin de simuler le développement logiciel en mode agile. Ainsi, chaque équipe, formée préalablement, s'est assignée une *user story*. Chaque équipe avait comme tâche de réaliser la *user story* avant la fin de la journée. Afin de donner des compléments d'information sur les exigences, la PO, Mme Trudel, était présente, supportée par une équipe de bénévoles formée de mesureurs *COSMIC* certifiés et d'étudiants à la maîtrise en génie logiciel.

Parmi ces bénévoles figuraient Jean-Marc Desharnais, un expert du domaine, et Jimmy Cloutier (moi-même). Diverses questions provenant des étudiants ont permis aux auteurs d'échanger et de mieux définir le concept de *FSM pattern* car, à ce moment, il existait une compréhension différente, mais complémentaire, entre les chercheurs. Les discussions générées ont eu un impact positif sur l'écriture de l'article (Trudel *et al.*, 2016) qui était en début de rédaction à ce moment.

De plus, ayant assisté à des cours sur la méthode *COSMIC* prodigués par M. Desharnais, dont certains avaient comme sujet une vision embryonnaire des *FSM patterns*, et ayant également travaillé en parallèle avec Mme Trudel sur sa vision de ce concept et des fonctionnalités de l'outil, j'ai pu jouer le rôle de bras droit de la PO lors de cette première tentative. Plus spécifiquement, j'ai pu encadrer diverses équipes afin de m'assurer que ce qu'ils implémentaient correspondait aux exigences énoncées par Mme Trudel. En plus de l'aspect fonctionnel, j'ai vérifié sommairement le code source produit par les étudiants afin d'être en mesure de poursuivre leurs travaux : le but à ce moment était de concevoir l'outil en tant que projet pour l'obtention de la maîtrise en génie logiciel.

Pour ce qui est de l'aide technique, celle-ci était octroyée par Alexandre Viau de la Maison du Logiciel Libre. Une fois qu'une équipe avait complété sa *user story*, celle-ci était validée par la *PO*. Lorsque le tout satisfaisait aux attentes, un *pull request* était envoyé à M. Viau afin que celui-ci intègre le tout dans la branche principale du dépôt *Git*. Avant le début de l'implémentation des fonctionnalités avec le *framework Play* utilisant le langage *Java*, chaque équipe a récupéré le dépôt *Git* du projet. Celui-ci contenait une structure de base pour le projet. Afin d'avoir accès, chaque étudiant a été préalablement ajouté au dépôt du *COSMIC-Sizing-Tool* sur *GitHub*. Il a résulté de cette journée quelques interfaces graphiques plus ou moins complètes. Cependant, ceci nous a permis de réaliser que le comportement de l'interface pour la saisie de la mesure posait problème. Nous avons donc décidé d'effectuer un prototype se concentrant particulièrement sur ce point afin de bien définir cette fonctionnalité névralgique et pouvoir expérimenter les objectifs énoncés au chapitre traitant de la problématique.

Parallèlement à cela, un modèle de base de données complet a été conçu par un DBA d'expérience, M. Luc-André Jolivet, lui aussi étudiant à la maîtrise en génie logiciel.

La deuxième tentative correspond à la création d'un prototype. Le but visé était de créer une interface utilisateur permettant la saisie de la mesure de la taille fonctionnelle selon *COSMIC* en plus d'implémenter toute la logique d'affaires, sous forme de *REST API*, permettant le calcul de la mesure et l'utilisation des *FSM patterns*. Plusieurs itérations ont eu lieu afin de clarifier l'ergonomie de l'interface de saisie de la mesure. Ces itérations ont été faites avec la *PO* et avec le post-doctorant Erdir Ungan, un spécialiste de la méthode *COSMIC*. Les premières itérations ont consisté à mieux définir le besoin. Afin de clarifier la compréhension de tous, une phase de prototypage papier a eu lieu. Suite à celle-ci, le développement de l'interface utilisateur en utilisant *React/Redux* et l'implémentation des règles d'affaires sous forme de *REST API* à l'aide de *Python/Flask* ont eu lieu. De ces activités, un prototype simpliste, mais fonctionnel, est né.

Les artefacts produits lors des différentes tentatives sont détaillés dans le chapitre suivant.



## CHAPITRE 5.

### MISES EN OEUVRE DE L'OUTIL DE MESURE *COSMIC-SIZING-TOOL*

Le chapitre précédent a entre autres permis d'apprendre qu'il y a eu deux tentatives pour l'implémentation du *COSMIC-Sizing-Tool*. Ce qui suit présente les divers artefacts résultants de ces deux tentatives. Celles-ci correspondent à la dernière phase de la méthodologie présentée dans le chapitre précédent, soit la phase consacrée au « Développement itératif ».

#### 5.1. Première tentative - Implémentation de fonctionnalités par des étudiants du baccalauréat

Ce qui suit correspond à la première tentative de développement. C'est lors de cette phase que le *COSMIC-Sizing-Tool* a pris une tournure plus tangible en générant divers artefacts.

##### 5.1.1. *Web Application Site Map*

Dans le but d'identifier la hiérarchie d'appels des modules constituant *COSMIC-Sizing-Tool*, ainsi que pour avoir une vue d'ensemble de ceux-ci, un *site map* de l'application a été créé par Mme Trudel et moi-même.

La figure 5.1 (page 46) représente la version la plus à jour de ce *site map*.

En un coup d'oeil, il est possible d'identifier les différents modules ainsi que leurs fonctionnalités les plus importantes. Les encadrés plus foncés correspondent à des fonctionnalités pouvant être utilisées par les utilisateurs alors que ceux plus pâles correspondent à des fonctionnalités systèmes ou d'administration du système. Pour ce qui est de l'image de chronomètre, elle représente la prise de mesure afin de calculer l'effort qu'il a été

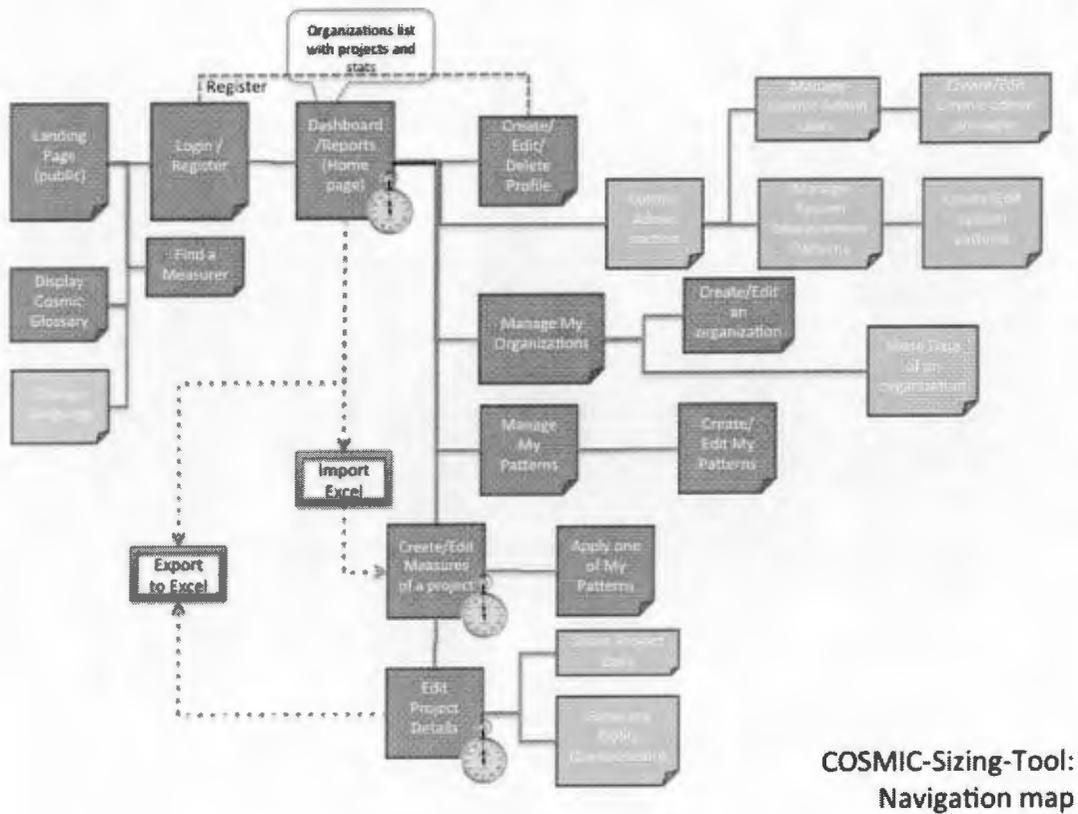


FIGURE 5.1. – Site Map du COSMIC-Sizing-Tool

nécessaire de déployer pour mesurer un projet à l'aide de l'outil, le but étant de fournir une mesure de l'efficacité du mesureur en minutes par CFP.

### 5.1.2. Hébergement

Durant un certain temps, le *COSMIC-Sizing-Tool* a été accessible via un hébergement sur *Heroku*. Celui-ci bénéficiait d'une intégration continue faisant en sorte que lorsque le dépôt *Git* principal du *COSMIC-Sizing-Tool* était mis à jour, l'application sur le site de *Heroku* l'était également. Cependant, même si nous étions censés bénéficier d'un compte gratuit, *Heroku* facturait de petits montants chaque mois (environ 10\$US/mois). Nous avons donc été dans l'obligation de désactiver l'intégration continue du site et fermer l'hébergement, raison pour laquelle il n'est plus possible d'accéder en ligne au travail effectué lors de cette première tentative.

### 5.1.3. Modèle de base de données

Un autre apport important de la première tentative fut la réalisation du modèle de la base de données. La figure 5.2 (page 48) donne un aperçu du modèle créé par Luc-André Jolivet, DBA expert, étudiant à la maîtrise en génie logiciel.

Ce modèle s'avère très complet et permet de sauvegarder l'information efficacement pour chacune des fonctions. En effet, une attention particulière a été portée au respect des meilleures pratiques (formes normales) de gestion de base de données.

### 5.1.4. *GitHub*

Dans le but d'effectuer la gestion centralisée du code source, c'est lors de la première tentative qu'un dépôt *GitHub* a été créé pour le développement du *COSMIC-Sizing-Tool*. L'adresse de ce dépôt est <https://github.com/cosmic-sizing-tool/cosmic-sizing-tool>. Cependant, celui-ci est présentement privé, il faut donc être un collaborateur confirmé par la *PO* pour y avoir accès.

Par ailleurs, *GitHub* étant un gestionnaire de source, mais également une plateforme de développement, c'est à cet endroit que le suivi du projet s'est effectué. En effet, les *user stories* élémentaires rédigées sur *post-it* ont été recopiées dans la section *Issues* afin de créer un *backlog* plus robuste. Enfin, celui-ci a été lié à *waffle.io*, ce qui a permis d'afficher les *issues* dans un tableau *Kanban* comme on peut le voir dans la figure 5.3.

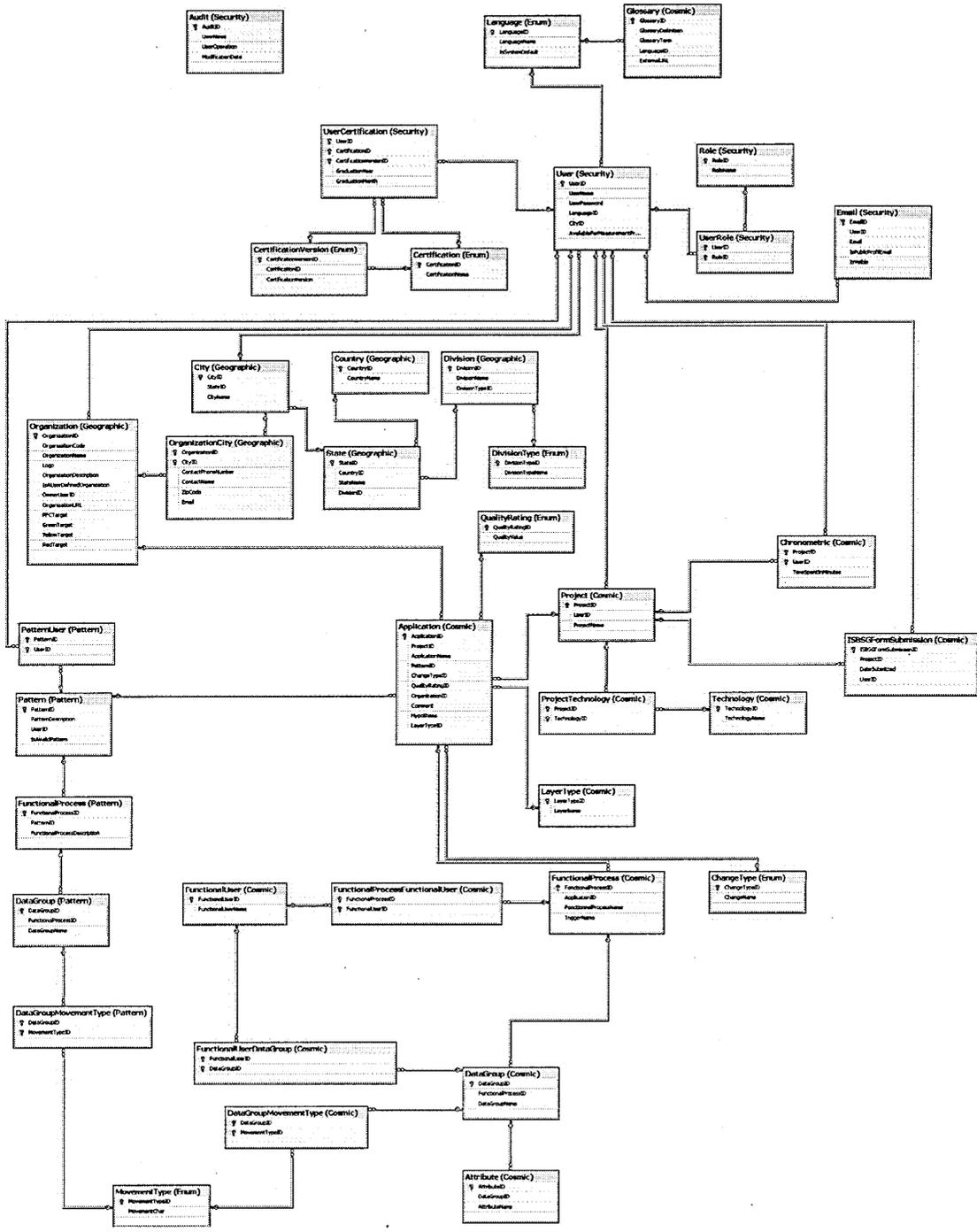


FIGURE 5.2. – Modèle relationnel du *COSMIC-Sizing-Tool*

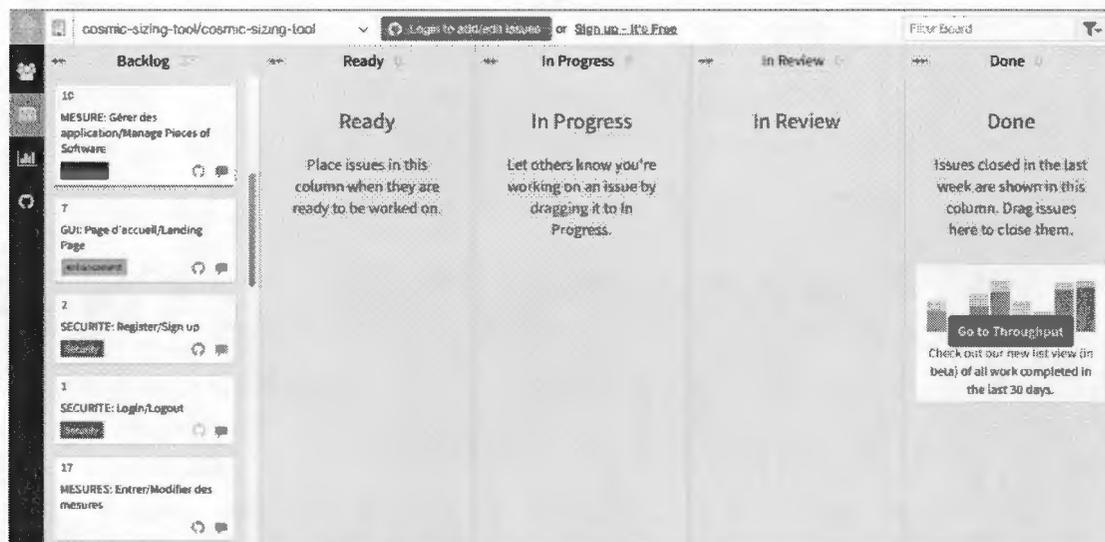


FIGURE 5.3. – Kanban *COSMIC-Sizing-Tool* dans waffle

### 5.1.5. Aperçu des fonctionnalités (partiellement) implémentées

Lors de la journée intensive simulant le développement d'un projet en mode agile, qui correspond à la première tentative de faire naître le *COSMIC-Sizing-Tool*, la majorité des fonctionnalités ayant un encadré foncé de la figure 5.1 avait été assignée à diverses équipes. Cependant, aucune équipe n'a réussi à implémenter complètement la *user story* qui leur avait été assignée. Par exemple, l'information saisie n'est pas enregistrée dans la base de données. Néanmoins, certaines fonctionnalités se sont vues dotées d'une interface utilisateur avec une portion de la logique de validation.

Rappelons que lors de cette journée, Mme Trudel a fait office de *PO* aidée par une équipe bénévole de mesureurs *COSMIC* certifiés, tels que M. Jean-Marc Desharnais, ainsi que d'étudiants à la maîtrise en génie logiciel dont moi-même. En plus d'aider les équipes de développement avec le volet fonctionnel en faisant office de bras droit au *PO*, je me suis attardé également au volet technique afin d'assimiler cette portion et d'être en mesure de poursuivre les travaux une fois l'encadrement des étudiants terminé.

### 5.1.5.1. Page d'accueil

La page d'accueil est présentée dans la figure 5.4.



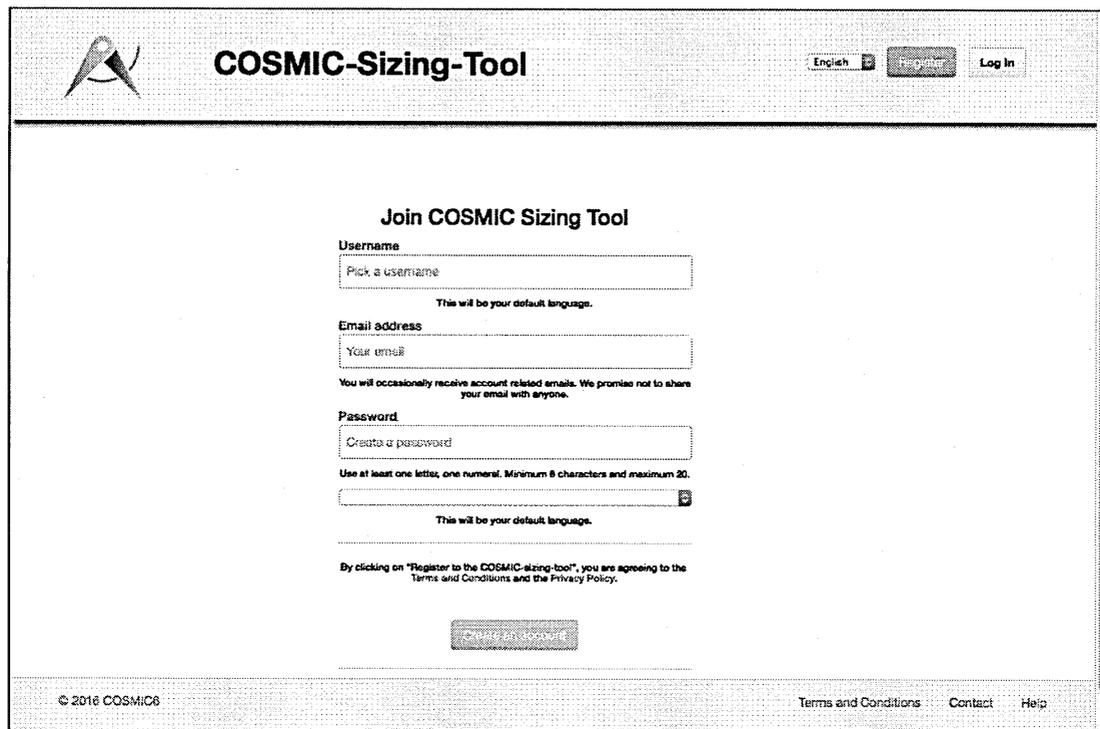
FIGURE 5.4. – *COSMIC-Sizing-Tool* : Page d'accueil

C'est la première page qui s'affiche lorsqu'un utilisateur navigue vers *COSMIC-Sizing-Tool*. À partir de cette section publique, il est prévu de pouvoir s'inscrire gratuitement afin d'accéder aux fonctionnalités « privées » du logiciel.

### 5.1.5.2. Inscription

Les fonctionnalités du *COSMIC-Sizing-Tool* ne sont accessibles que pour les utilisateurs inscrits. Tel que le démontre la figure 5.4, cette inscription est possible en remplissant les boîtes d'édition se trouvant au centre droit de la page d'accueil. En plus de pouvoir s'inscrire via celle-ci, une interface spécifique a été développée. Celle-ci est accessible par le biais de l'utilisation du bouton *Register* de la page principale.

La figure 5.5 donne un aperçu de cette page spécifique servant à l'inscription.



The screenshot shows the registration page for the COSMIC-Sizing-Tool. At the top left is the logo, a stylized 'A' with a pencil. The title 'COSMIC-Sizing-Tool' is centered at the top. On the top right, there are links for 'English', 'Register', and 'Log In'. The main content area is titled 'Join COSMIC Sizing Tool'. It contains three input fields: 'Username' with the placeholder 'Pick a username', 'Email address' with 'Your email', and 'Password' with 'Create a password'. Below the password field is a note: 'Use at least one letter, one numeral. Minimum 8 characters and maximum 20.' There is also a language selection dropdown set to 'English' with the note 'This will be your default language.' Below the form is a checkbox for 'By clicking on "Register to the COSMIC-sizing-tool", you are agreeing to the Terms and Conditions and the Privacy Policy.' and a 'Register account' button. The footer contains '© 2016 COSMIC6' on the left and 'Terms and Conditions', 'Contact', and 'Help' on the right.

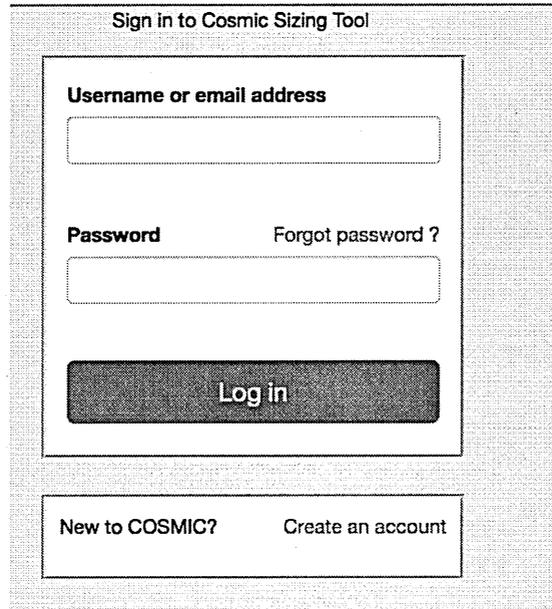
FIGURE 5.5. – *COSMIC-Sizing-Tool* : Inscription

Même si l'interface semble complète, ce n'est pas le cas de l'implémentation de l'inscription (création de l'utilisateur). En effet, l'enregistrement dans la base de données de l'utilisateur n'a pas été implémenté.

Une fois l'utilisateur inscrit, celui-ci doit s'authentifier dans le système afin d'avoir accès aux fonctionnalités de l'outil.

### 5.1.5.3. Authentication

La figure 5.6 correspond à l'interface permettant à un utilisateur préalablement inscrit de s'authentifier au système.



The image shows a web form titled "Sign in to Cosmic Sizing Tool". It contains two input fields: "Username or email address" and "Password". To the right of the password field is a link "Forgot password?". Below the input fields is a "Log in" button. At the bottom of the form, there are two links: "New to COSMIC?" and "Create an account".

FIGURE 5.6. – *COSMIC-Sizing-Tool* : Authentication

Cette fenêtre est accessible via le bouton *Log In* de la page d'accueil. Une fois authentifié, l'utilisateur est redirigé vers un « tableau de bord » correspondant aux mesures de ses projets.

Puisque la première tentative a laissé la fonctionnalité d'inscription non fonctionnelle, l'authentification ne fonctionne pas. En effet, un utilisateur doit initialement être inscrit afin de pouvoir par la suite s'authentifier.

### 5.1.5.4. Projets

La première tentative a également produit comme artefact une interface pour la création de projets de mesure. La figure 5.7 (page 53) en donne un aperçu.

Cette interface contient diverses sections et peut paraître rebutante pour la création d'un projet. Cependant, seulement quelques informations sont obligatoires. Les autres sections sont utiles afin d'envoyer les mesures à ISBSG. Comme pour les autres fonctions

1. Contact information for the questionnaire submitter.

Contact person:

Organisation:

Country:

E-mail:

2. Your identifying name or ID for this submitted project.

Project ID:

3. What was the role in this submitted project of the person who completed this questionnaire?

Analyst/Programmer     User     Customer/End     Development manager

Independent Reviewer     IT/MS Manager     Metrics Manager/Consultant

Project Manager/Leader     Project Office/Tech.Support    Other (specify):

Section B

Section C

Section E

Section F

Section G

Submit

FIGURE 5.7. – *COSMIC-Sizing-Tool* : Création d'un projet

implémentées lors de la première tentative, l'information saisie n'est pas enregistrée et les validations ne sont pas implémentées. Il n'y a que l'implémentation du module de *Patterns* qui interagit sommairement avec la base de données.

#### 5.1.5.5. *Patterns*

L'implémentation la plus aboutie provenant de la première tentative se trouve au niveau du module des *Patterns*. Une première interface permet de lister les *patterns* enregistrés dans la base de données telle que représentée dans la figure 5.8 (page 54).

Lorsqu'un utilisateur sélectionne un *pattern* dans la liste ou appuie sur *New Pattern*, l'interface bascule vers une page d'édition de *patterns* telle que présentée dans la figure 5.9 (page 54).

Puisque la fonctionnalité n'est qu'à moitié implémentée, l'interface n'affiche pas le *pattern* sélectionné. En fait, que ce soit à la suite d'une sélection ou après avoir appuyé sur *New Pattern*, l'interface affiche en édition toujours le premier *pattern* de la liste.

Néanmoins, diverses fonctionnalités ont été implémentées. En effet, les boutons *Add functional process* et *Add data group* permettent d'ajouter respectivement un nouveau

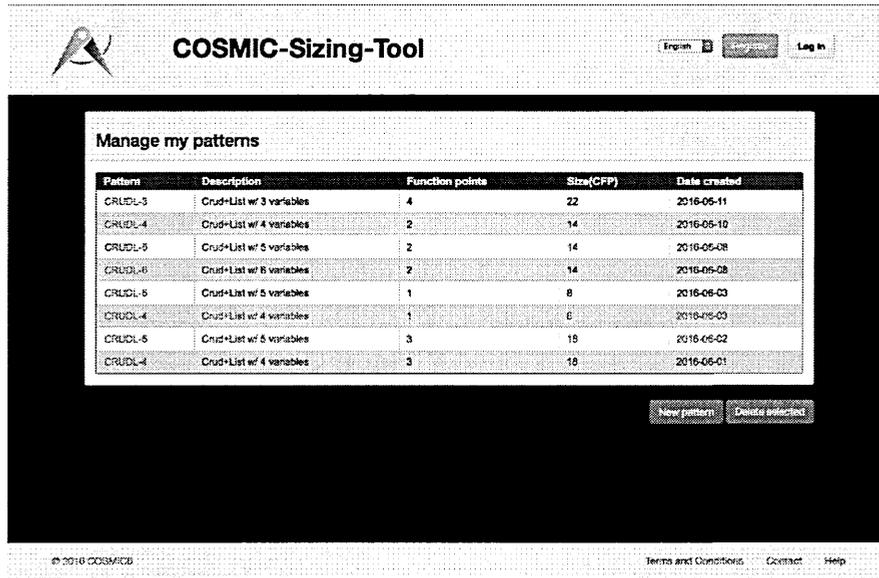


FIGURE 5.8. – COSMIC-Sizing-Tool : Liste des patterns

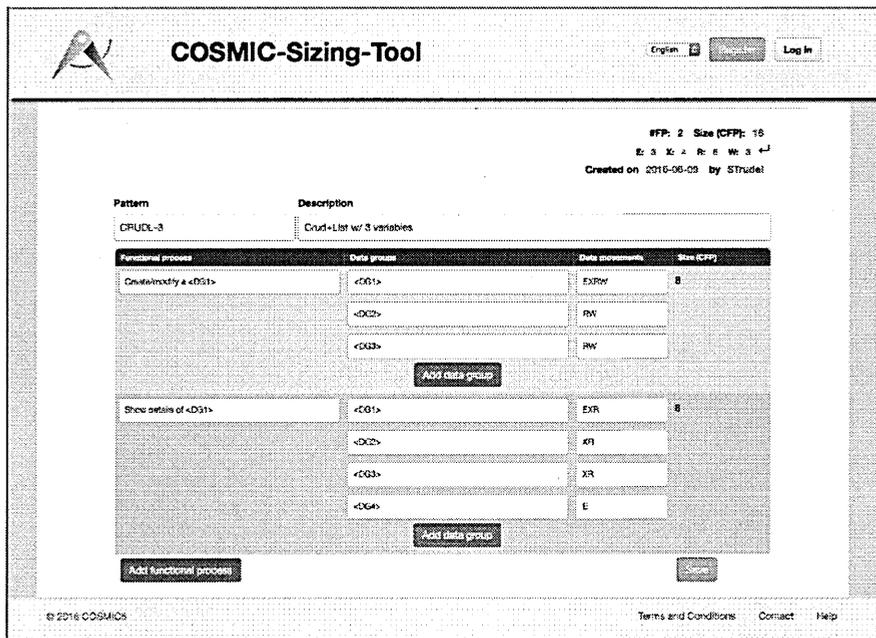


FIGURE 5.9. – COSMIC-Sizing-Tool : Édition d'un pattern

processus fonctionnel et des groupes de données. Enfin, lorsque l'information des *data movements* est entrée correctement, puis que l'utilisateur appuie sur *Save*, la taille en CFP du nouveau processus fonctionnel est calculée et le tableau sommaire sur la mesure se trouvant dans le coin en haut à droite de l'interface est mis à jour. Ceci est présenté dans la figure 5.10.

```
#FP: 3 Size (CFP): 19
E: 3 X: 5 R: 6 W: 5 ←
Created on 2016-06-09 by STrudel
```

FIGURE 5.10. – *COSMIC-Sizing-Tool* : Sommaire de la mesure

Cependant, l'information saisie n'est pas enregistrée dans la base de données. Résultat, les modifications effectuées sont présentement perdues.

Même si aucune des fonctionnalités n'est complète et utilisable, ceci nous a permis de clarifier certains points et de nous rendre compte que le tout devait se centraliser sur la prise de mesure en utilisant les *FSM patterns*. Nous avons donc recalibré l'implémentation afin de créer, dans un second temps, un prototype répondant à cette priorité. Le développement du *COSMIC-Sizing-Tool* pourrait reprendre une fois ceci clarifié.

## 5.2. Deuxième tentative - Prototype de saisie de la mesure avec les *FSM patterns*

Suite à la première tentative d'implémentation du *COSMIC-Sizing-Tool* dont résultent les artefacts de la section précédente, la priorité d'implémentation a été mise sur la création d'un prototype permettant la prise de mesure selon la méthode *COSMIC* en utilisant les *FSM patterns*. Ce qui suit présente les divers artefacts résultant de l'implémentation de ce prototype.

### 5.2.1. Maquettes d'interfaces

Les premières rencontres ont servi, entre autres, à *brainstormer* sur l'expérience utilisateur, plus précisément concernant l'interface de saisie des informations nécessaires à la mesure de la taille fonctionnelle, telles que les processus fonctionnels, les groupes de données et leurs mouvements.

Toutes les rencontres ont eu lieu avec Erdir Ugan, post-doctorant à l'UQAM, et certaines d'entre elles avec Mme Sylvie Trudel, *Product Owner* de l'outil. La technique de *Paper-Prototyping* a été utilisée lors de quelques séances afin de valider les concepts d'interface exposés lors des rencontres. Les artefacts produits lors des phases de *Paper-Prototyping* n'ont pas été consignés. Cependant, à partir de ces conceptualisations temporaires, des *wireframes* ont été réalisés afin d'en donner une meilleure représentation.

L'un des premiers concepts réalisés était basé sur l'utilisation de divers contrôles d'interfaces, tous indépendants les uns des autres et affichés selon une arborescence.

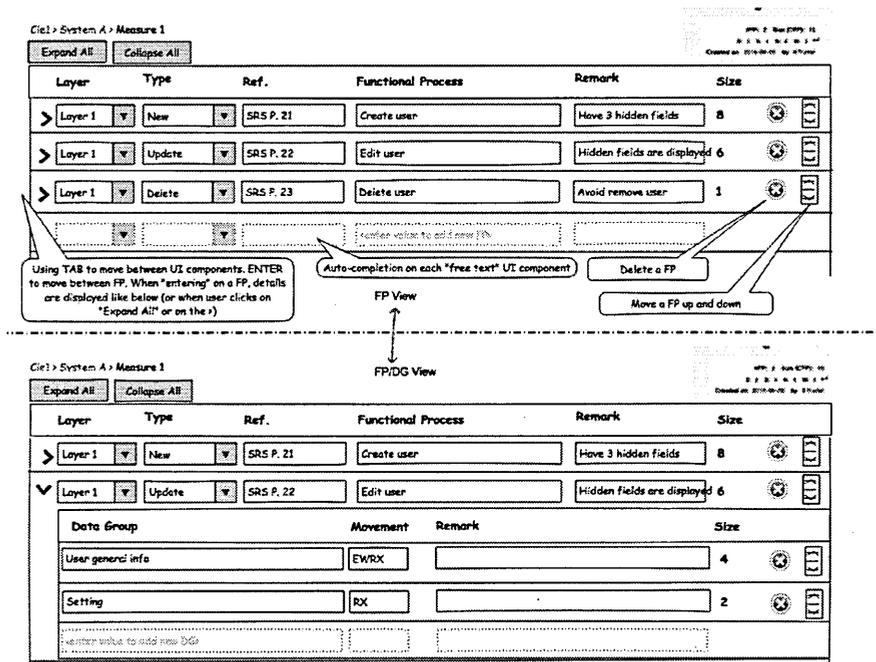


FIGURE 5.11. – Prototype : Arborescence composée de contrôles multiples

Pour ce qui est du deuxième concept, au lieu d'utiliser des contrôles séparés permettant de récolter l'information comme dans le prototype précédent, la saisie de données s'effectue dans une grille (voir figure 5.12, page 57). Ceci rappelle l'utilisation d'Excel, l'outil souvent utilisé par les mesureurs, et un affichage en arbre.

Puisque plusieurs mesureurs utilisent Excel pour consigner leurs mesures, un *wireframe* reprenant l'entrée dans une grille comme dans Excel a été créé (voir figure 5.13, page 57).

Au final, il a été décidé d'utiliser une conception en arbre du deuxième *wireframe*. Le premier niveau de l'arborescence concerne l'information des processus fonctionnels alors

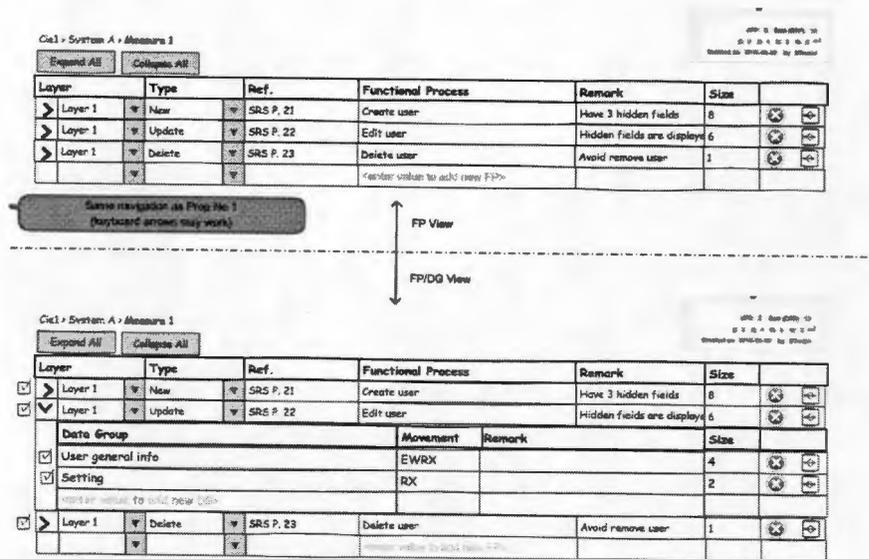


FIGURE 5.12. – Prototype : Arborescence avec grille

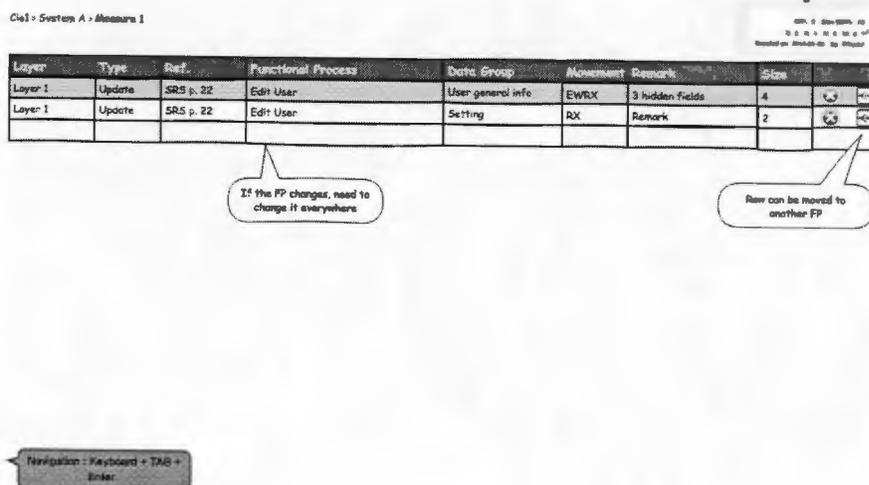


FIGURE 5.13. – Prototype : Similaire à Excel

que le second niveau se rapporte aux groupes de données et à leurs mouvements.

De plus, ce concept a été préféré au premier, car la difficulté technique semblait moins importante. En effet, en utilisant une grille dans un arbre, il sera possible d'utiliser un contrôle implémentant les différentes fonctionnalités voulues sans devoir les développer.

### 5.2.2. Modélisation

En plus de définir les maquettes d'interfaces, les premières rencontres ont également eu trait aux clarifications sur la modélisation du prototype. Lors de la première tentative, un modèle relationnel complet a été conçu. Dans le cas présent, l'implémentation d'un tel modèle n'étant pas nécessaire, nous nous sommes concentrés sur l'implémentation du nécessaire pour la prise de mesure et son rattachement à un projet lié à une organisation. Pour ce qui est des *patterns* de la mesure fonctionnelle, la structure serait la même que pour une mesure standard, mais elle serait reliée à un *pattern* au lieu d'un projet. Cette modélisation a émergé au cours de la création itérative et incrémentale du prototype en s'inspirant du modèle relationnel complet. Selon l'un des principes agiles provenant du manifeste : « Les meilleures architectures, spécifications et conceptions émergent d'équipes autoorganisées »<sup>1</sup>.

### 5.2.3. Web API du prototype

Suite aux rencontres préliminaires ayant servi à définir des maquettes d'interfaces et à circonscrire la modélisation, un effort important a été fourni afin de créer un *Web API* en *REST* afin de supporter les différentes logiques d'affaires nécessaires à l'implémentation du prototype. Ainsi, à l'aide du langage *Python* et de *Flask*, un service a été créé de façon à découpler au maximum les différentes fonctionnalités. L'objectif de ce découplage est de faciliter la réutilisation des fonctions implémentées dans le prototype en vue de la création éventuelle de l'outil. De plus, ceci correspond aux bonnes pratiques afin de facilement ségréger le tout en microservices si les performances venaient à se dégrader. Également, afin de faciliter l'évolution des fonctions accessibles via le service web, une notion de version est utilisée pour l'appel des fonctions *REST*.

Ce qui suit documente succinctement les différentes fonctionnalités du *Web API*.

---

1. <https://agilemanifesto.org/iso/fr/principles.html>

Le service web permet d'effectuer diverses opérations relativement aux organisations. Dans le cas du prototype, une organisation correspond à la donnée maître permettant de rattacher des projets pour ensuite entrer toute l'information relative à la saisie de la mesure.

Ainsi, l'appel du service avec *GET* en utilisant « /v1.0/organizations » permet de retourner la liste complète de toutes les organisations de la base de données.

Afin de retourner une organisation précise, l'appel s'effectue avec *GET* à l'adresse « /v1.0/organizations/<organization\_id> ». La dernière portion de l'adresse <organization\_id> correspond à un paramètre envoyé à l'adresse. Comme son nom l'indique, ce paramètre s'attend à recevoir l'identifiant unique correspondant à l'organisation dans la base de données.

La création d'une nouvelle organisation s'effectue avec *POST* avec l'appel de l'adresse « /v1.0/organizations ». L'appel se fait également en incluant un *BODY JSON* contenant l'information nécessaire à la création de l'organisation, dont son nom (obligatoire) et une adresse web (optionnelle). Pour le moment, ce ne sont que les deux informations pouvant être enregistrées au niveau de l'organisation. Cependant, avec l'évolution du prototype, d'autres informations seront ajoutées, au besoin, et le modèle devra évoluer en conséquence.

Pour ce qui est de la mise à jour, celle-ci est possible avec *PUT* en appelant l'adresse « /v1.0/organizations/<organization\_id> ». Comme dans le cas de la fonction permettant de retourner une organisation spécifique, le paramètre <organization\_id> se trouvant à la fin de l'adresse doit représenter l'identification unique d'organisation valide. Afin de mettre à jour l'information, la fonction reçoit également en *BODY* un *JSON* représentant l'information de l'organisation à mettre à jour.

Enfin, il est possible de supprimer une organisation avec *DELETE* en utilisant l'adresse « /v1.0/organizations/<organization\_id> ». Le paramètre <organization\_id> correspond à l'identifiant unique de l'organisation à supprimer.

Une gestion sommaire d'erreurs est implémentée dans l'outil. Ainsi, lors de l'appel d'une fonction, un code *HTTP* est retourné afin de préciser si le traitement a réussi ou échoué, suivi d'une description afin de fournir un complément d'information sur le traitement de la demande.

Les codes *HTTP* utilisés sont calqués sur ceux établis par HTTP/1.1<sup>2</sup>. Voici un aperçu des codes utilisés par les fonctions :

Tableau 5.1. – Codes *HTTP*

Code	Description
201	Appel fructueux et création de la ressource
202	Appel fructueux et mise à jour de la ressource en cours
400	Mauvaise syntaxe du <i>JSON</i> reçu
404	Lien ou information non trouvé(e)

Cette logique est répliquée pour les autres fonctionnalités : *Projet*, *Pattern*, *Processus* fonctionnel et *Mouvement de données*.

Seuls les liens d'appels, le nombre de paramètres et la structure du *JSON* sont différents. Par exemple, afin de créer un mouvement de données, l'appel se fait ainsi : `/v1.0/organizations/<organization_id>/projects/<project_id>/funcprocesses/<fp_id>/datamoves`. Comme on peut le constater, l'appel nécessite trois paramètres représentant les niveaux supérieurs où se trouve le mouvement de données (« *datamoves* ») à créer.

#### 5.2.4. Interface utilisateur du prototype

Simultanément à la création du *Web API*, un prototype d'interface a été développé afin de valider un concept de prise de mesure en utilisant les *patterns*. Ainsi, en utilisant diverses technologies web spécifiques pour la création d'interfaces utilisateurs, entre autres *React*, des interfaces ont été développées. Celles-ci sont dans leur plus simple expression afin de valider le concept de création de mesures. La forme et la présentation graphique ont donc été mises de côté.

Les prochaines figures de cette section donnent un aperçu des interfaces développées.

2. <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

Lors de la consultation du prototype, seule la liste des organisations présentes dans la base de données s'affiche.

Organization	Project
Organization 1	
Organization 2	
Organization 3	
Organization 4	

FIGURE 5.14. – Prototype : Liste des organisations

Suite à la sélection d'une organisation, les projets associés à celle-ci s'affichent. La sélection du premier projet de la liste s'effectue et l'information des processus fonctionnels de celui-ci s'affiche.

Organization	Project	Functional Processes and Datamovements	
Organization 1	Project 1	Org: 1 / Proj: 1	
Organization 2	Project 2		
Organization 3			
Organization 4			

Basic Create		Select Pattern
Add Row   Add 10 Rows   Delete Row		Apply Pattern
Fnc Name	Size	
▶ Create	0	
▶ Retrieve	0	
▶ Update	0	
▶ List	0	

Refresh Data | Clear

FIGURE 5.15. – Prototype : Liste des projets et processus fonctionnels du premier projet

Afin d'afficher les groupes de données appartenant à un processus fonctionnel, il suffit de cliquer sur le triangle en début de ligne. Ceci aura pour effet d'ouvrir l'arborescence (voir figure 5.16, page 62).

Le prototype contient divers boutons d'action permettant d'ajouter ou de supprimer des enregistrements (voir figure 5.17, page 62). Les boutons *Add Row*, *Add 10 Rows*, *Delete Row* se trouvant au début du tableau permettent d'ajouter et de supprimer des processus fonctionnels.

## Functional Processes and Datamovements

Org: 1 / Proj: 1

Select Pattern

Basic Create Apply Pattern

Add Row | Add 10 Rows | Delete Row

Fnc Name	Size															
▼ Create	0															
<table border="1"> <thead> <tr> <th>DG Name</th> <th>DG Move</th> <th>Size</th> </tr> </thead> <tbody> <tr> <td>First DG</td> <td>ERW</td> <td>3</td> </tr> <tr> <td>Second DG</td> <td>RX</td> <td>2</td> </tr> <tr> <td>Third DG</td> <td>RX</td> <td>2</td> </tr> <tr> <td>Error message</td> <td>X</td> <td>1</td> </tr> </tbody> </table>		DG Name	DG Move	Size	First DG	ERW	3	Second DG	RX	2	Third DG	RX	2	Error message	X	1
DG Name	DG Move	Size														
First DG	ERW	3														
Second DG	RX	2														
Third DG	RX	2														
Error message	X	1														
Add row   Add 10 rows   Delete row																
Retrieve	0															

FIGURE 5.16. – Prototype : Liste des groupes de données du processus fonctionnel sélectionné

## Functional Processes and Datamovements

Org: 1 / Proj: 1

Select Pattern

Basic Create Apply Pattern

Add Row | Add 10 Rows | Delete Row 

Fnc Name	Size
▼ Create	

FIGURE 5.17. – Prototype : Processus fonctionnels - boutons d'action

Les actions relatives aux groupes de données se font à la fin du tableau réservé à cette information.

Add Row Add 10 Rows Delete Row		
Fnc Name	Size	
▼ Create		
DG Name	DG Move	Size
First DG	ERW	
Second DG	RX	
Third DG	RX	
Error message	X	
Add row Add 10 rows Delete row ←		

FIGURE 5.18. – Prototype : Groupes de données - boutons d'action

Ces boutons sont utiles pour ajouter ou supprimer des enregistrements. Pour ce qui est de l'édition, ceci est possible directement dans la grille.

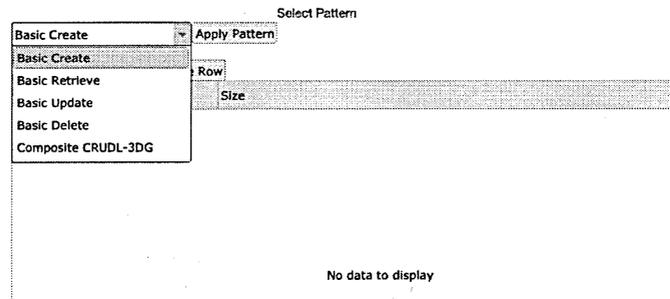
Add Row Add 10 Rows Delete Row		
Fnc Name	Size	
▼ Create		
DG Name	DG Move	Size
First DG - modif	ERW	
Second DG	RX	
Third DG	RX	
Error message	X	
Add row Add 10 rows Delete row		

FIGURE 5.19. – Prototype : Édition au niveau de la liste

Enfin, le prototype contient également une fonctionnalité importante, soit la possibilité d'utiliser les *patterns* de la mesure afin de rapidement créer une mesure de la taille fonctionnelle. Ceci se fait en sélectionnant le *pattern* à appliquer puis en appuyant sur *Apply Pattern* (voir figure 5.20, page 64).

## Functional Processes and Datamovements

Org: 1 / Proj: 2

FIGURE 5.20. – Prototype : Liste des *patterns*

Ensuite, en appuyant sur *Apply Pattern*, une nouvelle fenêtre s'affiche permettant de spécifier les noms des groupes de données utilisés par le *pattern* sélectionné.

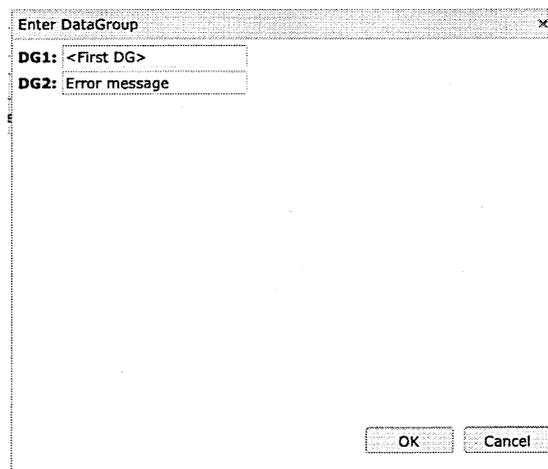


FIGURE 5.21. – Prototype : Spécification des groupes de données

Finalement, en appuyant sur le bouton *OK*, le *pattern* est appliqué en utilisant les noms spécifiés précédemment. La mesure de la taille fonctionnelle se fait donc rapidement sans devoir entrer toute l'information nécessaire, par exemple, en omettant les groupes de données (voir figure 5.22, page 65).

## Functional Processes and Datamovements

Org: 1 / Proj: 2

Select Pattern

Basic Create    Apply Pattern

Add Row | Add 10 Rows | Delete Row

Func Name	Size
Create Customer	
DG Name	DG Move
Customer	ERW
Error message	X
	3
	1

Add row | Add 10 rows | Delete row

Refresh Data | Clear

FIGURE 5.22. – Prototype : Mesure de la taille fonctionnelle en appliquant un *pattern*

### 5.3. Évaluation des objectifs énoncés dans la problématique

Le chapitre traitant de la problématique énonce deux objectifs : 1) réduire l'effort de la mesure et 2) obtenir une taille dès la phase initiale d'un projet. Avec l'état actuel du *COSMIC-Sizing-Tool*, il est possible d'évaluer jusqu'à quel point ces objectifs ont été atteints. Ce qui suit traite de cette évaluation et des résultats obtenus.

#### 5.3.1. Objectif 1 - Réduire l'effort de la mesure

Le premier objectif est lié à l'effort nécessaire à la prise de mesure afin de réduire celui-ci. Avec l'état actuel du *COSMIC-Sizing-Tool*, il est possible de calculer l'effort théorique pour mesurer un processus fonctionnel avec la méthode *Keystroke-level model* (KLM)<sup>3</sup> en moins de 5,69 secondes en utilisant les *FSM patterns* déjà présents dans le système (*Basic Create, Basic Retrieve, Basic Update, Basic Delete* et *Composite CRUDL-3DG*) et en gardant inchangés les termes génériques proposés pour représenter les *data groups* (*First DG, Second DG, etc.*). Le nombre de secondes a été calculé en utilisant le scénario suivant :

3. [https://en.wikipedia.org/wiki/Keystroke-level\\_model](https://en.wikipedia.org/wiki/Keystroke-level_model)

Tableau 5.2. – Temps requis - *COSMIC-Sizing-Tool*

Action	Durée (sec.)
Initier l'action	1,35
Utiliser la liste déroulante pour le choix du <i>pattern</i>	3,04
Déplacer la souris vers le bouton OK	1,10
Appuyer sur le bouton OK pour fermer la fenêtre	0,10
Relâcher le bouton OK	0,10
Total	5,69

Par ailleurs, lors de simulations en utilisant l'outil, nous réalisons l'opération en moins de 3 secondes. Néanmoins, pour une personne moins habituée, un effort de près de 6 secondes demeure plausible.

De plus, cette durée n'est pas influencée par le nombre de groupe de données contenu dans le processus fonctionnel à mesurer<sup>4</sup>. Ainsi, une mesure utilisant le *FSM pattern Basic Create* requiert le même temps qu'un *Composite CRUDL-3DG* en utilisant l'outil.

Comparativement, un mesureur entrant l'information contenue dans un *Basic Create* dans un tableur tel qu'Excel peut nécessiter près de 14,43 secondes selon la méthode KLM, telle qu'illustrée ci-dessous avec l'hypothèse d'une personne de niveau « *average non-secretary typist* ».

Tableau 5.3. – Temps requis - Excel - *Basic Create*

Action	Durée (sec.)
Initier l'action	1,35
Déplacer la souris vers la colonne	1,10
Déplacer la main de la souris vers le clavier	0,40
Écrire le processus fonctionnel « <i>Create</i> » (6 entrées clavier * 0,28)	1,68
TAB pour changer de case (1 entrée clavier * 0,28)	0,28

4. Ceci est vrai pourvu que le processus fonctionnel demeure raisonnable. Par exemple, obtenir une telle mesure en utilisant un *FSM pattern* personnalisé par l'utilisateur contenant des dizaines de groupes de données pourrait influencer le temps nécessaire à réaliser la mesure causée par un ralentissement du système.

Action	Durée (sec.)
Écrire le groupe de données « <i>data group</i> » (10 entrées clavier * 0,28)	2,80
TAB pour changer de case (1 entrée clavier * 0,28)	0,28
Écrire le mouvement de données « ERW » (3 entrées clavier * 0,28)	0,84
Déplacer la main du clavier vers la souris	0,40
Déplacer la souris vers l'autre ligne colonne	1,10
Écrire le groupe de données « <i>Error message</i> » (13 entrées clavier * 0,28)	3,64
TAB pour changer de case (1 entrée clavier * 0,28)	0,28
Écrire le mouvement de données « X » (1 entrée clavier * 0,28)	0,28
Total	14,43

Pour ce qui est d'un *Composite CRUDL-3DG*, ceci peut prendre jusqu'à 2 minutes selon une estimation avec la méthode KLM. Selon ces données, l'utilisation du *COSMIC-Sizing-Tool* en utilisant les *FSM patterns* peut donc être jusqu'à 15 fois plus rapide qu'une saisie manuelle dans un tableur.

Dans l'éventualité où l'utilisateur modifierait les termes proposés par défaut par le système au niveau des groupes de données, l'économie de temps se manifesterait principalement par ce qui est nécessaire au mesureur pour saisir les noms des groupes de données contenus dans un *Basic FSM pattern*. Cependant, le gain est plus important dans le cas d'un *FSM pattern* plus complexe, par exemple, pour un *Composite FSM pattern* ou un *Multi-composite FSM pattern*. En effet, le système permet de ne saisir qu'une seule fois le nom du groupe de données au lieu de nécessiter la saisie partielle comme c'est le cas dans un tableur.

Cependant, ces résultats doivent être interprétés avec réserve, car ils ont été obtenus sur un ordinateur exécutant simultanément les portions *front end* et *back end* en ne traitant qu'une seule connexion. Les résultats pourraient donc différer dans un autre environnement. En effet, dans un environnement de production, le produit final pourrait obtenir des résultats différents provenant, par exemple, de la latence du réseau entre

les serveurs hébergeant le *front end*, le *back end* et la base de données. Bref, le résultat obtenu est directement relié, entre autres, à l'architecture de déploiement et à la charge d'utilisation de l'outil. Dans l'éventualité d'une poursuite du projet, cet objectif devrait être converti en un indicateur de suivi puis mesuré tout au long du développement afin de s'assurer que l'outil permette à un mesureur d'être plus rapide que s'il effectuait la consignation de la mesure dans Excel.

### **5.3.2. Objectif 2 - Obtenir une taille dès la phase initiale d'un projet**

Similairement à l'objectif 1, le fait que le système permette d'omettre d'entrer certaines informations nécessaires à la mesure selon *COSMIC* (processus fonctionnels, groupes de données et mouvements de données) puisqu'elles sont fournies par l'utilisation des *FSM patterns*, permet d'avoir celle-ci très tôt et rapidement. En effet, il suffit d'utiliser les *FSM patterns* correspondants au projet de développement logiciel à mesurer pour obtenir une estimation de la taille en CFP. Par la suite, si le développement du projet est confirmé, cette estimation réalisée à l'aide de *FSM patterns* pourra être ajustée afin, par exemple, d'utiliser les bons termes au niveau des groupes de données ou d'ajouter/retirer des mouvements de données après l'analyse des spécifications.

## CHAPITRE 6.

### DISCUSSION SUR LES OBSTACLES ET DÉFIS RENCONTRÉS LORS DE LA RÉALISATION DU PROJET DE MAÎTRISE

Les chapitres précédents ont décrit les différentes activités effectuées dans le cadre de notre projet de maîtrise. Celles-ci ont été variées telles que, entre autres, la rédaction d'un article, la réalisation d'une revue systématique de la littérature, l'encadrement d'étudiants dans la création d'une première mouture du *COSMIC-Sizing-Tool* et la programmation de prototypes dans une autre technologie. Tout ceci relié par le nouveau concept des *FSM patterns*. Pour sa part, le présent chapitre amorce une discussion sur l'exécution de ces différentes activités.

Plus précisément, il sera question d'une rétrospective énonçant les obstacles et les défis rencontrés. Ceci permettra d'effectuer un retour d'expérience afin d'identifier ce qui pourrait être fait autrement afin d'obtenir un meilleur résultat, être plus efficace ou efficient. Enfin, l'état du projet *COSMIC-Sizing-Tool* sera traité afin d'identifier les améliorations qui pourraient être apportées, le but étant de faire passer l'outil d'un stade de prototype à un produit fonctionnel. Ainsi, ceci permettra de statuer sur les travaux futurs dans l'éventualité où le développement du *COSMIC-Sizing-Tool* se poursuivrait par un tiers.

#### **6.1. Rédaction de l'article *Functional Size Measurement Patterns : A Proposed Approach* (Trudel et al., 2016)**

Même accompagné par deux chercheurs chevronnés tels que Mme Trudel et M. Desharnais, participer à la rédaction d'un premier article scientifique s'est avéré être un défi en soi. Particulièrement lorsque le sujet principal, la méthode *COSMIC*, est en cours d'apprentissage. À cet apprentissage académique devaient s'ajouter certains autres concepts nécessaires à la rédaction de l'article afin d'asseoir les affirmations inscrites dans l'article

sur des bases solides. Ainsi, plusieurs recherches et lectures d'articles ont été nécessaires afin d'en apprendre davantage sur les travaux effectués sur le concept de *pattern* particulièrement ceux faits par Alexander *et al.* (1977) puis adaptés dans le domaine du développement logiciel par Gamma *et al.* (1993). De plus, l'article a été rédigé selon le *framework Design Science Research Methodology* tel que décrit par Peffers *et al.* (2007). Ce *framework* était tout désigné afin d'aider à produire l'artefact voulu des *FSM patterns*. Il a donc été nécessaire d'apprendre celui-ci.

Ainsi, il est important d'avoir de bonnes compétences informationnelles afin de trouver les différents articles sur lesquels baser la rédaction. En effet, il faut garder en tête, lors de celle-ci, que tout ce qui est avancé dans l'article doit être appuyé par une source crédible vérifiée par les pairs. Ceci peut contraster avec mes autres habitudes d'écriture non académique dans lesquelles certains aspects peuvent être avancés sans preuve ou basés sur des expériences personnelles. Par la suite, il est également important de développer des compétences pour identifier rapidement si l'article trouvé s'avère être une source pertinente sans devoir lire avec minutie tout le document.

## 6.2. Réalisation d'une revue systématique de la littérature

La réalisation d'une revue systématique de la littérature ayant comme sujet les outils supportant la mesure de la taille fonctionnelle des logiciels selon la méthode *COSMIC* a exigé un effort considérable dans le cadre de ce projet. Le chapitre 3 y est entièrement consacré. Au premier abord, rechercher de l'information dans diverses sources est habituel pour un étudiant à la fin de la maîtrise en génie logiciel. En effet, cette habileté est nécessaire pour presque tous les travaux académiques à remettre. Cependant, dans le cadre d'une revue systématique de la littérature, cette recherche d'information se fait de façon beaucoup plus rigoureuse. Celle-ci s'effectue en interrogeant, à l'aide de mots clés précis, des bases de données d'articles sélectionnés avec soin qui utilisent des moteurs de recherches ayant une syntaxe et des fonctionnalités différentes. Selon les mots clés utilisés et les possibilités de recherche offertes par le moteur de recherche, le nombre d'articles retournés peut être impressionnant. De plus, un même article peut se retrouver dans différents moteurs de recherche et être ainsi comptabilisé plusieurs fois. Lorsque le nombre d'articles trouvés explose de la sorte, ceci peut être décourageant.

Afin d'éviter de se noyer sous la charge de travail nécessitant la vérification d'un nombre faramineux d'articles plus ou moins pertinents avec le sujet de la RSL, le choix des mots clés s'avère être une étape cruciale. Pour ce faire, il est important de tester les mots

clefs afin de vérifier le nombre d'articles retournés puis vérifier un échantillon afin de déterminer si les mots clefs choisis doivent être raffinés. Lorsque possible, faire appel à un professionnel de la recherche informationnelle sur ces différentes bases de données scientifiques (ex. un des bibliothécaires) serait un avantage considérable.

En plus de tester les mots clefs afin de les raffiner, il est également important de vérifier ceux-ci sur les différentes bases de données. En effet, plusieurs s'avèrent être des méta-bases de données. Ainsi, chercher dans celles-ci, en plus des bases de données sources, fait en sorte de récupérer plusieurs fois le même article, ce qui fait gonfler le nombre total d'articles à traiter. Après vérification, la décision pourra être prise d'exclure les recherches dans l'une des bases de données (par exemple, la base de données source où l'article a été publié) afin d'éviter de traiter inutilement un même article plusieurs fois.

Autre aspect important à considérer permettant de limiter les articles à traiter est la sélection de l'intervalle de date. En effet, la plupart des moteurs de recherche des bases de données offrent une option permettant de filtrer selon la date de parution des articles. Avec l'évolution rapide des différents domaines du génie logiciel, récupérer des articles datant de plusieurs décennies ne fait parfois aucun sens. Il faut donc se questionner sur ce critère de recherche. Dans notre cas, nous pouvions limiter nos recherches d'articles dans le temps afin d'exclure ceux implémentant une ancienne version de la méthode *COSMIC*. Cependant, ce n'est pas toujours possible. En effet, effectuer une telle limitation priverait le chercheur de récupérer des articles correspondant à des travaux fondateurs servant de base à d'autres articles.

Enfin, le but ultime recherché est de récupérer directement le maximum d'articles pertinents en lien avec le/les objectifs de la RSL en évitant le triage manuel. Dans le cas de la RSL dont a fait l'objet ce projet, le fait de passer de 575 articles récupérés des bases de données à une sélection finale de 29 articles montre clairement des lacunes qui auraient pu être minimisées en mettant en pratique le retour d'expérience ci-haut.

Faire tout le nécessaire afin de limiter les articles superflus ne serait pas d'une grande utilité sans une organisation de ceux-ci. Ayant en tête le principe bien connu *Don't repeat yourself (DRY)*, il est fortement suggéré de se doter, dès le départ, d'une structure adéquate pour éviter de relire plusieurs fois le même article ou de devoir le chercher de nouveau dans les bases de données. Dans le cas présent, cette organisation a été effectuée en utilisant les fonctions d'exportation des différentes bases de données interrogées. En effet, la plupart d'entre elles offrent la possibilité d'exporter le résultat de recherche

en format *CSV (Comma Separated Value)* et même de spécifier certaines informations complémentaires à intégrer au fichier tel que le résumé de l'article, le(s) auteur(s), l'année de publication, le lien direct vers l'article, etc. En plus de permettre de garder des traces des résultats obtenus, le fait d'avoir une liste pouvant s'ouvrir sous Excel nous a permis d'utiliser diverses fonctions de ce logiciel. La première, qui fut d'une grande aide, fut la fonction de détection et de suppression des doublons. Ceci était essentiel, car nos diverses requêtes ont souvent retourné les mêmes articles. Cette fonction nous a permis d'éliminer les doublons sans peine. Un autre avantage de l'utilisation d'un tableur a été la facilité avec laquelle il est possible d'ajouter de l'information pour la consigner avec l'article. Pour ce faire, il suffit d'ajouter une nouvelle colonne. Ceci nous a permis d'identifier les articles intéressants, de garder en note les noms d'outils et de consigner d'autres notes importantes en lien avec les questions justifiant la RSL ou autres nécessaires à la rédaction. Par la suite, grâce à la fonctionnalité de filtre au niveau des colonnes d'Excel, il s'avérait aisé d'extraire l'information voulue et pouvoir manipuler les données.

Un dernier retour d'expérience sur la RSL se situe au niveau de l'extraction de regroupements possibles à effectuer avec les articles pertinents et ainsi nourrir la RSL avec quelques statistiques intéressantes. Dépendamment des sujets, ceci peut s'avérer un défi en soi, ce qui fut notre cas. Afin d'aider à dégager les sujets traités par les différents articles, nous avons créé une carte conceptuelle. Cet exercice de réflexion nous a ainsi permis d'effectuer les constats décrits au chapitre consacré à la RSL.

### **6.3. Première tentative de développement du prototype *COSMIC-Sizing-Tool***

La première tentative de mise en oeuvre du *COSMIC-Sizing-Tool* par les étudiants de baccalauréat a connu son lot de défis et d'obstacles. Notre optimisme nous a fait croire que les étudiants seraient prêts et que l'effort de la journée intensive produirait des livrables sur lesquels nous pourrions continuer la réalisation de l'outil pour en faire une application utilisable. Cependant, nous avons vite réalisé que ce n'était pas le cas, malgré les rencontres d'information préalables sur la technologie et sur le déroulement de la journée afin que tous soient prêts lors de celle-ci. En effet, la majorité n'avait pas expérimenté le *framework Play* et certains ne connaissaient *Java* que de nom. Une portion de la journée a donc été consacrée, pour certaines équipes, à l'installation du *framework* et

à expérimenter celui-ci. Un grand défi d'encadrement des étudiants s'annonçait parsemé de plusieurs obstacles.

Si cela était à refaire, au lieu de demander à tous les étudiants de se familiariser avec les technologies proposées, il aurait été préférable d'identifier les aptitudes des étudiants et de leur attribuer des objectifs à atteindre selon celles-ci. Ainsi, ceux ayant déjà de fortes compétences en *Java*, avec le *framework Play* ou avec le développement d'applications web auraient été en mesure d'avancer certaines fonctionnalités de l'outil au lieu d'être mobilisés à aider les personnes en difficulté. D'autres auraient pu aider à préciser certaines exigences fonctionnelles ou à réaliser des design d'écrans utilisateurs conviviaux et attrayants.

De plus, en procédant ainsi, le résultat attendu aurait été mieux défini au lieu de ne focaliser que sur la production d'un incrément logiciel tel que l'avaient sous-entendu les équipes. Puisque le contexte était la création d'une application web et que les *user stories* décrivaient les fonctionnalités, les étudiants ont interprété qu'ils devaient absolument programmer alors que ce n'était pas obligatoire. En effet, il aurait été tout à fait acceptable que certaines équipes effectuent des exercices de *Paper prototyping* afin de faire accepter un design d'interface par la *PO*. Ainsi, les compétences de chacun auraient été mieux utilisées.

Après réflexion, si le but ultime avait été de produire un incrément logiciel, il aurait été utile, entres autres, de fournir aux équipes une architecture existante implémentant déjà certaines fonctionnalités, ainsi que des *user stories* plus précises incluant des imprimés d'écrans correspondant aux fonctionnalités à produire. Les étudiants auraient alors disposé d'une base à améliorer au lieu de devoir tout définir eux-mêmes. Pour la plupart, ceux-ci ne possédaient pas l'expérience requise pour réaliser un développement logiciel sans une base préalable. Ainsi, la tâche, pouvant être perçue comme un défi très motivant pour une personne d'expérience, s'est avérée être déstabilisante.

Enfin, notre vision optimiste nous a aussi voilé le fait que les *user stories* étaient dans un niveau de détail sommaire afin de susciter la conversation. Des séances de *grooming* sur les *user stories* de deux heures par équipes ont eu lieu avec la *PO*. Cependant, la majorité des équipes n'ont envoyé que deux ou trois personnes sur les cinq ou six membres de leur équipe. De toute évidence, l'information détaillée des *user stories* ne s'est pas rendue à tous par la suite. Il aurait été nécessaire que chacun des membres puisse bien comprendre le besoin exprimé afin de proposer une solution adéquate. Ainsi,

si c'était à refaire, une séance de *grooming* avec tous les membres de chaque équipe aurait été avantageuse avant de s'attaquer à leur *user story*.

Malgré tout, en combinant le manque de préparation des étudiants, la courbe d'apprentissage technologique, la mauvaise interprétation des buts à atteindre et le manque de détails des *user stories*, certaines équipes ont été en mesure de produire des artefacts tel que présentés au chapitre 5. De plus, cette première tentative a été bénéfique à la définition des *FSM patterns* grâce aux discussions suscitées suivant les questions des étudiants et les réponses données par les participants. Ainsi, malgré les résultats mitigés, ceux-ci ont tout de même été source de motivation pour poursuivre les travaux en effectuant une deuxième tentative plus ciblée.

#### **6.4. Deuxième tentative de développement du prototype *COSMIC-Sizing-Tool***

La première tentative n'ayant pas permis de produire un incrément logiciel stable pouvant servir de base à la création d'un outil supportant les *FSM patterns*, nous avons décidé de continuer notre effort de réalisation en effectuant une deuxième tentative de création d'un prototype pour *COSMIC-Sizing-Tool*. Lors de la première tentative, mon défi se concentrait sur l'encadrement des étudiants en jouant le rôle de bras droit de la *PO* avec un accent sur le volet technique afin de poursuivre le développement une fois la journée complétée. Cependant, lors de la deuxième tentative, j'ai vécu les défis et obstacles rencontrés par les étudiants lors de leur journée intensive constituant la première tentative de création du *COSMIC-Sizing-Tool*.

Les défis et obstacles rencontrés lors de cette deuxième tentative peuvent se résumer ainsi : courbe d'apprentissage. Malgré mes 15 années sur le marché du travail dans le domaine du développement logiciel, mes compétences pratiques de programmation sont désuètes. En effet, celles-ci correspondent à mon début de carrière en tant que programmeur *Delphi* d'applications Client/Serveur fonctionnant sous Windows, carrière qui a rapidement bifurqué vers l'élicitation des exigences et la gestion. Participer à la création d'une application web telle que le *COSMIC-Sizing-Tool* s'avérait donc être une opportunité de mettre à jour mes compétences en programmation.

La réalisation de la portion *back end*, l'*API REST* programmé en *Python* à l'aide de *Flask*, n'a présenté aucune difficulté particulière. Bien entendu, un apprentissage a été

nécessaire au niveau du langage, de la structure et de la bibliothèque, mais rien de comparable à ce qui a été nécessaire pour le *front end*.

En effet, le choc est venu lors de la création de l'interface utilisateur (*front end*). Fort de mon expérience avec le langage *Delphi* où la création d'une interface Windows s'effectue rapidement en positionnant des contrôles graphiques existants dans l'IDE anciennement offert par Borland, la quantité de code nécessaire à écrire et le nombre de technologies nécessaires à maîtriser sont, de prime abord, contre-intuitifs et démotivants. En développement web, il faut une expertise sur le *HTML*, *CSS*, *Javascript*, des *frameworks JavaScript* pour faciliter la gestion des contrôles et l'interface, etc. Ceci, sans compter les problèmes inhérents aux diverses versions des navigateurs internet. Pour un ancien programmeur *Delphi* créant des interfaces Windows à l'aide de l'IDE *Delphi* rapidement et simplement, il est étonnant de devoir passer autant de temps à la création d'interfaces sans un support visuel ; ceci s'apparente à une régression malgré l'avancement de la technologie.

De plus, la documentation disponible des contrôles tiers laisse parfois à désirer, surtout lorsqu'il est nécessaire de les intégrer à d'autres *frameworks*. Ce fut le cas avec la grille utilisée. À première vue, celle-ci était compatible avec *React* et semblait posséder la documentation adéquate, mais lors de l'implémentation, nous nous sommes rapidement rendu compte qu'il fallait adapter ce qui était mis de l'avant par la documentation. Celle-ci décrivant une utilisation native en *JavaScript*, nous avons été contraints d'user d'astuces afin d'implémenter certaines fonctions avancées à l'intérieur du *framework React*. Il faut donc rester attentif à cela, car ceci complexifie l'apprentissage, ce qui freine la vitesse de développement.

### 6.5. État du *COSMIC-Sizing-Tool*

Cette section donnera un aperçu de l'état actuel du *COSMIC-Sizing-Tool*, de même que les prochaines étapes qui seraient à envisager afin de continuer celui-ci pour livrer une première version stable utilisable par tous.

Malgré le nombre d'heures passées à l'implémentation du *COSMIC-Sizing-Tool* en combinant les deux tentatives, l'outil demeure dans un état de prototype. En effet, l'implémentation de toutes les fonctionnalités initialement énoncées, correspondant à la portée initialement établie, s'avère trop grande. Lors du cours MGL7240 - Mesure et développement logiciels, l'application complète a été mesurée à plus de 300 CFP. En prenant

un effort médian de programmation de 10 heures/CFP, plus de 3000 heures seraient nécessaires pour réaliser l'ensemble. Même si ma propre productivité s'était située autour de 3 heures/CFP, ceci aurait constitué plus de 900 heures de travail seulement pour le développement, ce qui est nettement supérieur aux exigences du programme de la maîtrise pour un projet de 15 crédits. Ceci explique les raisons pour lesquelles l'outil est toujours dans un état de prototype.

Plus précisément, la portion *back end* représente ce qui est le plus avancé. En effet, il existe plusieurs fonctionnalités permettant d'insérer, de modifier et de lire l'information dans la base de données afin de créer des projets de mesure ainsi que des fonctionnalités pour utiliser les *FSM patterns*. En fait, ce qui a été développé peut servir de squelette d'architecture afin de poursuivre le développement de l'application. Par ailleurs, l'appel aux fonctions du service bénéficierait d'une amélioration. Celles-ci pourraient être encapsulées sous forme d'objets en utilisant *Flask-RESTful*<sup>1</sup>. Cette possibilité ayant été découverte durant le développement, une approche conventionnelle de l'utilisation de *Flask* a été implémentée.

Comparativement au *back end*, le *front end* est dans un état beaucoup plus sommaire. En effet, dans son état actuel, celui-ci ne peut être utilisé que pour certaines tâches restreintes. Par exemple, il peut être utilisé pour simuler certaines idées précises telles que l'emploi d'un *FSM pattern* déjà inclus dans le système afin d'obtenir une mesure. Il sert également d'exemple à la consommation de certaines fonctionnalités de l'*API REST*. Autre utilité, cette version réduite du *front end* permet de tester le comportement du contrôle de grille sélectionné (*jqxGrid*). Enfin, dans le cas d'une poursuite de l'implémentation du *COSMIC-Sizing-Tool*, un travail important de designs d'interfaces devra être effectué afin de rendre le tout convivial.

De façon plus générale et afin de transformer ce prototype en version 1 utilisable, il serait impératif d'implémenter la gestion de la sécurité. Pour le moment, aucune gestion de sécurité n'est implémentée. La confidentialité et l'intégrité des données, ainsi que la disponibilité de l'application, peuvent donc être mises en péril sans effort, ce qui est inacceptable pour une application web.

Autre point important, lors du développement, *SQLite* a été utilisé pour la persistance des données. Il serait donc important de connecter l'application à un SGBD tel que *PostgreSQL* (le choix technologique pour *COSMIC-Sizing-Tool*). Pour ce faire, une configuration spéciale devra être programmée afin de permettre une compilation rapide pour le

---

1. <https://flask-restful.readthedocs.io/en/latest/>

développement en utilisant *SQLite*, puis une seconde en vue de la création d'une version utilisant *PostgreSQL*.

Finalement, dans l'éventualité où une tierce personne serait intéressée à poursuivre le projet, un « guide du développeur » sommaire a été rédigé. Celui-ci est inclus en annexe G.



## CONCLUSION

Ce projet de synthèse, présenté comme exigence partielle de la maîtrise en génie logiciel, a comme fil conducteur le nouveau concept des *Functional Size Measurement Patterns*. L'approche a été décrite pour la première fois dans Trudel et Abran (2009).

Partant de la prémisse que l'implémentation des *FSM patterns* dans un outil informatisé était nécessaire pour atteindre deux des quatre objectifs décrits dans l'article, soit 1) réduire l'effort de mesure et 2) obtenir une taille dans la phase initiale d'un projet, une revue systématique de la littérature a été réalisée. Celle-ci avait comme but d'identifier les outils existants qui implémentent la mesure selon *COSMIC* et dont nous pourrions potentiellement nous servir afin d'implanter les *FSM patterns*.

Puisque cette RSL n'a pas permis d'identifier d'outils potentiels, des recherches ont été amorcées au niveau des produits offerts par l'industrie. Malgré cet effort de recherche additionnel, nous n'avons pas été en mesure de trouver un outil informatisé sur lequel nous pourrions implémenter les *FSM patterns*. Nous avons alors décidé de créer un outil implantant la mesure de la taille fonctionnelle afin d'y incorporer les *FSM patterns* et ainsi tenter de répondre à la problématique qui correspond aux deux objectifs mentionnés ci-haut.

L'implémentation de cet outil nommé *COSMIC-Sizing-Tool* a requis jusqu'à maintenant deux tentatives.

La première tentative a été réalisée par des étudiants du baccalauréat en informatique lors d'une journée intensive. Bien qu'il y ait eu de l'encadrement et des rencontres d'information préalables, ceci n'a pas permis d'obtenir un résultat final pouvant servir d'architecture de base pour poursuivre l'implémentation. Malgré tout, cette journée a permis de clarifier certaines fonctionnalités et le concept de *FSM pattern*.

La deuxième tentative, réalisée à partir de technologies différentes de la première, a permis de réaliser une version sommaire d'un service web permettant de persister les données pour un projet de mesure ainsi qu'une interface simpliste. Même si le résultat obtenu demeure à l'état de prototype, il est possible de donner un état sur les objectifs énoncés dans la problématique puisque les fonctionnalités nécessaires à l'implémentation

des *FSM patterns* s'y trouvent partiellement. Ce qui suit effectue un retour sur ces deux objectifs.

## Retour sur les objectifs

**Objectif 1 - Réduire l'effort de la mesure :** Au chapitre 5, nous avons démontré que l'efficacité de l'outil de mesure avec les *patterns* pourrait être jusqu'à 15 fois plus rapide comparativement à une saisie manuelle.

**Objectif 2 - Obtenir une taille dans la phase initiale d'un projet :** Puisque le *COSMIC-Sizing-Tool* est dans un état de prototype contenant peu de *FSM patterns* au niveau de l'interface<sup>2</sup> et n'offrant pas la possibilité à l'utilisateur de créer ses propres *FSM patterns* pour ensuite les utiliser, nous n'avons pas été en mesure de confirmer cet objectif. Afin que ceci demeure une priorité si l'évolution du *COSMIC-Sizing-Tool* se poursuivait par une tierce personne, cet objectif devrait être transformé en indicateur de suivi de projet afin de s'assurer que le logiciel poursuive cet objectif.

En bref, dans l'état actuel, les deux objectifs nous semblent atteignables. Cependant, le *COSMIC-Sizing-Tool* demeurant dans un état de prototype, tout particulièrement au niveau de l'interface, il nous est impossible de confirmer avec certitude l'atteinte de ceux-ci. De plus, puisqu'ils sont intimement liés à l'environnement technologique et à l'évolution de l'outil, des indicateurs permettant de suivre ceux-ci seront nécessaires afin de s'assurer que ces objectifs soient maintenus. Enfin, une entité tierce, qui voudrait continuer le développement de *COSMIC-Sizing-Tool* afin de valider les objectifs ou dans un autre but, pourrait se référer au *Developer's Guide for the COSMIC-Sizing-Tool* dont une copie est disponible en annexe G.

---

2. La portion *back end* est beaucoup plus avancée que la portion *front end*. En effet, le *front end* ne tire profit que d'une infime partie des services offerts par le *REST API*. Exemple : Le *back end* permet la création de *FSM patterns* personnalisés. Cependant, l'interface utilisateur ne permet pas d'appeler ce service.

**ANNEXE A.**

**ARTICLE *FUNCTIONAL SIZE MEASUREMENT  
PATTERNS : A PROPOSED APPROACH***

# Functional Size Measurement Patterns: A Proposed Approach

Sylvie Trudel  
Dept. of computer science  
Université du Québec à Montréal  
Montreal, Canada  
Email: [trudel.s@uqam.ca](mailto:trudel.s@uqam.ca)

Jean-Marc Desharnais  
Dept. of Software Engineering and IT  
École de Technologie Supérieure  
Montreal, Canada  
Email: [jean-marc.desharnais@etsmtl.net](mailto:jean-marc.desharnais@etsmtl.net)

Jimmy Cloutier  
Dept. of computer science  
Université du Québec à Montréal  
Montreal, Canada  
Email: [cloutier.jimmy.2@courrier.uqam.ca](mailto:cloutier.jimmy.2@courrier.uqam.ca)

**Abstract**— This paper introduces the concepts of functional size measurement (FSM) patterns. These FSM patterns apply to the COSMIC FSM method and may help in solving several issues, such as being able to do an early sizing for estimation, helping inexperienced measurers in learning how to apply the COSMIC method by establishing the relationship between the method rules and the measurement results, avoiding measurement errors, and reducing measurement effort. The approach used to define FSM patterns is Design Science Research. This paper explores the motivation factors, defines the objectives of the solution, and describes the design and development of FSM patterns. FSM patterns can be defined for a portion of a functional process, a single functional process type, a set of functional process types having a main data group - or object of interest - as their common starting point, or a large set of functional processes handling multiple data groups. These types of FSM patterns have been defined as “micro FSM pattern”, “basic FSM pattern”, “composite FSM pattern”, and “multi-composite FSM pattern” respectively. FSM patterns examples are given to demonstrate their applicability in information systems for all four FSM pattern types. Application of FSM pattern with embedded real-time systems is discussed, along with other potential usage benefits for the industry.

**Keywords**— COSMIC, Patterns, Design Science, Functional Size Measurement, Early estimation

## I. INTRODUCTION

Trying to predict the effort and cost required for building or modifying software with evolutive technologies and changing requirements is the challenge that face software estimation. As stated in [1], “experience has shown that developing realistic estimates for software project is an error-prone process”. The reason is that “project attributes [...] are often quite different than the estimated parameters” [1]. In response to this problem, functional size measurement (FSM), introduced by A. J.

Albrecht (1979) [2], can be used early in the project phases to produce a sizing based on the requirements of the software to implement or modify. This sizing then becomes a valuable attribute in an estimation process.

Since 1979, FSM concepts were developed [3] and many methods emerged from this development such as IFPUG [4], MkII [5], NESMA [6] or FiSMA [7]. The COSMIC method [8] is also one of these methods compliant with the concepts of FSM and is the one selected for our research. COSMIC is a standardized ISO/IEC FSM method that uses function points - its measurement unit is named the COSMIC Function Point (CFP) - to measure the software size from functional user requirements (FUR). The COSMIC Measurement Manual [9] defines a three-phase method (Fig. 1).

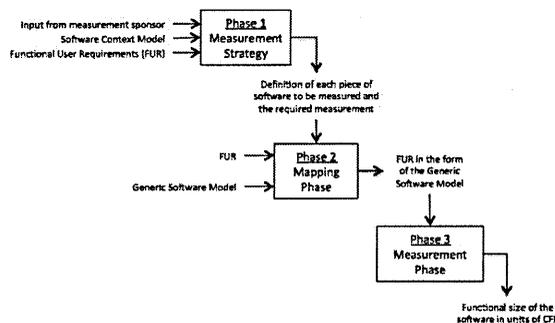


Fig. 1. The COSMIC method three-phase approach, adapted from [9].

Sizing software with COSMIC implies to apply all three phases of the method. In Phase 1, the measurement scope is established in accordance with the measurement purpose. In

phase 2, functional processes are identified from the FUR for each piece of software within the measurement scope. Then, for each identified functional process, objects of interest and their related data groups are identified. For each data group within a functional process, applicable data movements amongst Entry, eXit, Read and Write are identified (Fig. 2). In Phase 3, the number of data movements is added to produce the value of the measured software functional size. A CFP represents a single data movement.

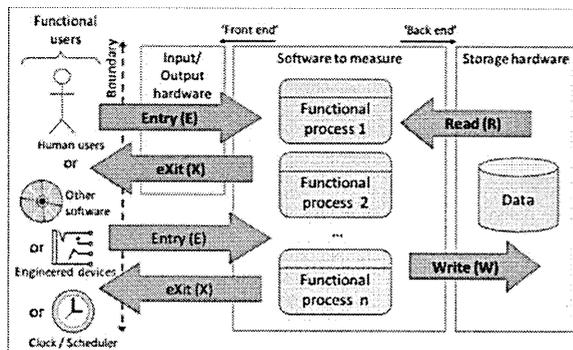


Fig. 2. Generic software data flow from a functional perspective, adapted from [10].

Even if these concepts can be easily understood by anyone having experience in software development [11], several studies such as [12] or [13] demonstrated that inexperienced measurers tend to do similar errors while learning to apply the COSMIC sizing rules. However, accuracy of sizing involves understanding of COSMIC rules but it is largely dependant on clarity and completeness of functional requirements. Wrong sizing can have repercussion not only on project effort, duration and cost estimation, but also on a diversity of practices related to software engineering such as improving software development cycle, benchmarking or performance measurement. Inexperienced measurers, because they are in a learning process, also measure functional size slower than an experienced measurer [14].

Therefore, solutions that can help inexperienced measurers to identify adequately functional processes, data groups and data movement, thus adequately applying the COSMIC method by helping to quickly mastering its rules, will have a major impact on the accuracy of measurement results. In return, this improved accuracy will reduce impacts of wrong effort, duration and cost estimation derived from incorrect sizing that can add risks to a project.

A very small portion of software projects worldwide has their functional size measured. Based on the author's experience, this can be explained from many reasons, such as low maturity of the software processes, fear that learning to measure is too expensive, or fear that the measurement cost outweigh the measurement benefits. This research paper focuses on the later issue in designing a potential solution meant to circumscribe measurement effort.

Research aimed at automating functional size measurement is currently being done but results require specific conditions, such as defining in details UML diagrams (e.g. class or sequence diagrams) [15]. There are software development projects that do not document their requirements and designs with UML. The authors of this paper have worked with many Agile teams documenting user and functional requirements using "user stories" and "epics" [16] for which measurement automation does not exist. We recognize that automating measurement might play a key role in FSM adoption in industry. However, the subject of this paper is not automation as it focuses solely on a means to accelerate FSM until research on automation be more advanced.

The remainder of this paper is organized as follows. Section II describes the research methodology where a Design Science Research Methodology framework was used. Section III demonstrates the concepts of FSM patterns through a series of examples. Section IV holds the conclusion and discusses opportunities arising from FSM patterns and future work, namely potential experiments to evaluate usage of FSM patterns.

## II. RESEARCH METHODOLOGY TO DEFINE FSM PATTERNS

The research described in this paper applies a Design Science Research Methodology (DSRM) Framework [17] with the following steps. First, we identify various aspects of the problems related to FSM, along with motivation factors. Second, we define the solution objectives to solve those problems. Our third step is the design and development of FSM patterns concepts *per se*. The fourth step consists of a demonstration of the FSM Patterns concepts through a series of examples. The fifth step of a standard DSRM framework, that consists in evaluating the designed concepts, is yet to be done and will be discussed in the future work. The last step consists of communicating the outputs and results of our research.

### A. Problem Identification and Motivation

#### 1) Helping inexperienced measurers

For a novice measurer, trying to apply a new method, such as the COSMIC method, is difficult even if he knows the rules and the software. At first, he does not know how to put all the pieces together. This is why the instructor, when teaching how to apply the rules to measure a software, normally gives a number of examples. For a specific case (e.g. modifying the name of a client in a database), the instructor shows how to apply the rules and obtain a measurement result. The instructor can present many examples and, in time, the novice gets an idea of how it works with similar cases. At that point, the novice builds his cognitive process of what the "mapping" phase is about: understanding the relationships between the COSMIC rules and the piece of software being measured. From there, it is possible for the instructor to go further and show to the novice how to measure different types of functional processes, not only modifying data, but also adding, deleting, viewing, reporting, and any other typical and non-typical functional process.

As shown in the introduction, inexperienced measurer tend to make similar measurement errors, such as omitting functional processes, data groups, or data movements,

identifying the wrong data movements, omitting the triggering Entry, and making wrong functional or data group breakdown [12][13].

Measuring a single functional process as asked by an instructor may become easier rapidly. But for an inexperienced measurer to measure by himself a whole measurement scope of one or several pieces of software imposes a larger cognitive load leading to measurement mistakes. There was also another approach used by Desharnais *et al.* [18][19] namely a knowledge base aiming to help inexperienced measurers to understand the COSMIC rules but this knowledge base was developed to show a measurer what to do, but not how. An experienced measurer also needs “how-to” knowledge.

### 2) Reducing measurement effort

The COSMIC Measurement Manual proposes a structure to record “the results of a measurement for each identified software item of an overall scope that has been mapped to the Generic Software Model” [9], that consists of a matrix representing functional processes as lines, data groups as columns, and data movements as cell content. Total of Entry, eXit, Read, and Write should appear per functional process. This type of matrix can be easily done using a spreadsheet and will work just fine when measuring a rather simple piece of software. When measuring a large system with several pieces of software, it is expected that the number of data groups be large as well. In that case, the proposed measurement recording structure will tend to be ineffective. To cope with that issue, professional measurers have developed and used a more generic spreadsheet structure, with functional processes, data groups, and data movements in their own column then using the pivot table functionality to obtain summary sizes per functional process. In both structures, the measurer has to type in all detailed measurement data. Although it may seem cumbersome, the auto-complete feature of the spreadsheet tool alleviates this effort. Nonetheless, it represents manual effort where the largest part is not the recoding activity *per se* but the reading and understanding the FUR.

In the past years, some attempts were made trying to automate software measurement. For example, Jenner proposed that software FSM be automatically measured from use cases supported by their sequence diagrams [20]. Instead of being based on UML as proposed by Jenner, Marin *et al.* [21], for their part, proposed size estimation from OO-Method approach [22]. These solutions are based on the need to create other artefacts derived from FUR that help exposing functional processes, data groups, data movements and other information required to apply the COSMIC method. Unfortunately, these artefacts are created at the design phase while sizing for estimation is needed at a much early stage. Furthermore, creating these artefacts require effort that many organisations will invest only if absolutely needed, such as in a complex messaging architecture with synchronous and asynchronous messages.

Instead of deriving measurement from other artefacts, another proposed approach was to develop tools that can support FSM in reducing measurement effort. However, available tools are still facing the problematic of using a generic spreadsheet, as stated above, but in a more structured

way and with some advanced functions. In the end, the effort for recording, reading and understanding the FUR activities are still required.

### 3) Sizing projects at an early phase

Several software practitioners show enthusiasm when they start learning a sizing method, knowing they should be able to use it to better estimate their future projects. However, they are deceived when faced with this reality: they are asked to provide an estimate before the FUR are written, at a time when detailed information about data groups and data movements is unknown. To help them “estimate” the size, rather than “measure” the size, Voegelzang *et al.* proposes a series of approaches and techniques to approximate sizing [23]. There is value in these techniques for those in need of an early estimation, but there is always a risk that the approximated value might be further than expected from the true software size, measured once the FUR are written or known, despite taking into account the scope creep that may happen in between.

One of the techniques for early estimating could consist of determining the approximate COSMIC size from an available data model. In the '90s, Desharnais and Morris, while trying to establish a validation process for the measurement process using the IFPUG method, established a link between the data entities and the transactions by analyzing 161 IFPUG measured applications. The results showed that the size portion related to data entities represented an average of 28% of the number of function points [24][25]. This means that knowing the number of data entities for a piece of software, it is possible to extrapolate the size of the software. However this technique is giving only an approximate value, knowing that the link between data and transactions could vary greatly from one software to another. For example, the size percentage related to data entities could represent only 20% of the total for a software and as much of 60% of the total for another software. Although this technique may have given promising results, it could not be extended to the COSMIC method because there is not such distinction between data and transactions when using COSMIC, but Functional Process.

When projects start and their FUR are not yet written, many organisations already have a project vision with objectives. One of the first piece of information brought forward is about what the software project is needed, that is the business objects or data entities for which a software is required for processing. These data entities may be categorized into several types [26] but we retained the following three types for the purpose of our research:

1. **Master data** “describes the essential business entities of a company, such as suppliers, customers, products, employees, and assets” [27] in the case of Information Systems (IS), and they may be equivalent to business objects or business classes in the case of embedded/real-time systems.
2. **Transactional data** represents information about an organisation’s activities, such as invoices (summary) and invoice details, customer orders with their details, purchase orders and their details, disbursements with invoices paid, or money deposits with customer

invoices paid, and so on for IS. Transactional data for embedded/real-time software is the data occurring during instances of execution of specific functional processes, such as interrupts, control signals, or specific data entered by a user or generated by the system as a result of executing an operation.

3. **Reference data** refers to parameter entities or objects containing a list of standardized values useful for referencing in several functional processes or even amongst several software systems, such as the list of countries, states or provinces, sales territories, product types, signal types, etc.

Entities in those three types should become an object of interest or a data group in the FUR to come. Process reference models such as the Functional Reference Architecture for Corporate Master Data Management (MDM) have been published in [28] to assist functional analysis activities. These models could be used at an early project phase to identify the required functional processes. Once the business domain is understood and that a clear mapping could be done between functional processes and data groups, a preliminary sizing could be done, even if the FUR are yet to be written. At the time of writing this paper, we have not found references in existing literature demonstrating these concepts for a potential early sizing instead of an approximation.

4) *Assuring accuracy of measurement*

As size measurement accuracy is influenced by the quality of functional requirements documentation [29], the analysis documentation, where all the functional requirements are detailed, needs to be precise enough to allow a measurer to identify, among other things, the functional users, functional processes, data groups, data movements and, optionally, data attributes. Afterward, the measurer will be able to size the software with precision. Missing information such as data groups or functional processes will have a negative impact on the final measurement result and, evenly, on the other metrics using the size as a source. It is the case for effort estimation. However, effort estimation is often required at an early stage of a project to provide, for example, the project cost to a customer. At this stage, the detailed analysis is not yet written. Premature measurement based on partial FUR containing missing functional users, functional processes, data groups or data movements will generate inaccurate sizing. For that reason, Desharnais *et al.* [30][31][32] have proposed a five levels scale - a through e - to qualify availability of data for measurement purposes in order to assure quality of measurement results. The five selected categories for the documentation availability, in decreasing order on an ordinal scale, are as in TABLE I.

TABLE I. DESCRIPTIONS OF QUALITY CATEGORIES FOR AVAILABILITY OF MEASUREMENT INFORMATION IN FUR [30].

Cat.	Description
a	The functional processes in the documentation are completely documented, including the data model used throughout the set of functional processes. Example: In the functional process, the data movements (i.e. Entry, Exit, Read, or Write) along with the data groups are identified.

Cat.	Description
b	The functional processes are documented, but without a precise data model, making it challenging to adequately identify all the data movements of distinct data groups. Example: The triggering event can be identified, but the grouping of the data attributes into data groups cannot be identified precisely.
c	The functional processes are identified in the requirements document, but without details that could permit identification of the precise number of their individual data movements. Example: The functional process can be identified, but not the individual data groups that would allow unambiguous identification of each corresponding data movement.
d	The number of functional processes is explicitly stated. Example: The number of functional processes is explicitly stated, but the functional processes are not individually documented. Example of a FUR in the category: 'Five reports will be provided to management.'
e	The functional process is implicit in the documentation. Example: A functional process cannot be found in the documentation, but can be inferred by an expert software engineer or given by the software engineer.

The FUR assessment of a specific software system could result as in TABLE II.

TABLE II. EXAMPLE OF AN OVERALL AVAILABILITY OF FUR DOCUMENTATION OF A GIVEN SYSTEM.

Category	Number of FUR	Percentage
a	24	48%
b	16	32%
c	4	8%
d	2	4%
e	4	8%
<b>Total</b>	<b>50</b>	<b>100%</b>

TABLE II. shows that the documentation of the software is very good (Level "a") for 48%, good for 32% and less good for 20% of the functional processes. It is also possible to analyze the FUR quality of different software systems in an organization as showed in [31]. However, FUR quality any other than of category "a" risk being wrongly sized as required information to do a precise measurement is missing. A good solution should be able to alleviate that problem.

B. *Solution Objectives Definition*

Given the identified problems, we have defined the following objectives for a suitable solution:

1. The solution must help an inexperienced measurer to apply the rules governing the COSMIC method.
2. The solution should contribute in reducing measurement effort.
3. The solution should allow for an early sizing, even without detailed written FUR.
4. The solution should provide a more accurate sizing by helping to avoid common measurement mistakes.

These four objectives constitute the large part of our motivation in finding a solution.

### C. Design and Development of FSM Patterns

#### 1) Related work on patterns

After many years of sizing software, COSMIC experts realized some recurrences, projects over projects, which can be generalized. In other words, patterns emerged from measurement experiences. In [32], Alexander defined a pattern as “a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice”. His works were initially related to architecture but resonate up to software engineering. Gamma *et al.* used Alexander’s definition and applied it to software design in [33] to “propose design patterns, a new mechanism for expressing design structures”. The suggested usages for design patterns were multiple. Among other things, design patterns:

1. “Distil and provide a means to reuse the design knowledge gained by experienced practitioners”,
2. “Help a novice perform more like an expert”, and
3. “By using design patterns early in the life cycle, one can avert refactoring at later stages of design” [33].

These design patterns utilization meet the objectives pursued by a solution responding to the problematic described in the previous section. More specifically, design patterns utilization (1) and (2) can be compared to our first and second objectives (helping the inexperienced measurers apply the COSMIC rules and become efficient in sizing). Our third objective relates to providing an accurate measure at an early stage of a project, before the detail analysis is defined, finds a similarity with the utilisation (3) to provide a prospective solution.

Therefore, inspired by the design patterns utilization enumerated above, the hypothesis is that creating artefacts defining patterns identified by COSMIC expert measurers should produce similar results than design patterns do. The name we gave to these patterns is “functional size measurement patterns” (FSM Patterns). The following subsection will define FSM Patterns in the perspective of the COSMIC method and, at the same time, explaining how it can be a solution to help inexperienced measurers to master COMIC rules in producing accurate measure and, also, helping to produce accurate FSM results without having all the details related to the FUR required by the COMIC method.

#### 2) FSM pattern definition

The mapping phase of the COSMIC method represents the biggest part of the sizing activity. Its main output is a representation of the piece of software being measured called the “COSMIC Generic Software Model”, which consists of the relationships between functional processes, data groups, and data movements, one of which should be the Entry triggering the related functional process. These relationships must be consistent with all applicable rules of the COSMIC method for any given context. Therefore, we define a FSM pattern as follows:

A FSM pattern is a predefined generic software model solving a recurring measurement problem in a specific context.

As such, the FSM pattern concept implicitly refers to the COSMIC method rules used to measure a specific set of data movements. A FSM pattern is composed of relationships of functional process(es), data group(s), and data movement(s) showing an adequate measurement solution to a recurring measurement problem. By its nature, FSM patterns should help understanding how to apply the COSMIC method to a specific measurement problem in a given context.

A FSM pattern applies on one or a series of typical sub-processes (i.e. data movements). Experimented professional COSMIC measurers may identify a large number of FSM patterns throughout the many COSMIC measurement projects he/she will do over several years. There is a possibility that every business domain be fertile to uncover a different set of FSM patterns, some of which may be more common than others as they would apply across multiple domains. For example, when a functional process is required to display error messages, regardless of the different occurrences of error messages, then the functional process has an eXit data movement on the “Error message” data group. This very simple FSM pattern can be applied for MIS software across multiple business domains, as well as for embedded/real-time software having an error display mechanism. At this level, we named this FSM pattern a micro FSM pattern, which is very small. But there are other categories of FSM patterns, and the remaining of this section describe some of the common FSM patterns categories met by measurers when sizing software.

#### 3) Categories of FSM pattern

As we pursue our digging into common solutions to recurring measurement problems in specific contexts, we realized that FSM patterns could be categorized as follows:

- Micro FSM patterns
- Basic FSM patterns
- Composite FSM patterns
- Multi-composite FSM patterns

We propose a classification of FSM patterns according to the number of functional processes and data groups involved. To demonstrate their applicability, examples are given in section III.

##### a) Micro FSM pattern

A **Micro FSM pattern** applies to a fragment of a functional process, involving one or several data groups. The example previously given about displaying an error message suits that definition.

##### b) Basic FSM pattern

A **Basic FSM pattern** applies to a complete yet single functional process.

The difference between “Basic” FSM pattern and “Micro” FSM pattern lies on functional process. The “basic” FSM

pattern applies to a complete functional process. The simplest basic FSM pattern is composed of a functional process handling a single data group. However, basic FSM patterns are not limited to only one data group but can also handle multiple data groups. The key point is that a basic FSM pattern is related to a whole functional process. For example, a common basic FSM pattern in IS would be "Create [a data group]"; in embedded/real-time, it would be "Process [a signal received]".

#### c) Composite FSM pattern

A **Composite FSM pattern** applies to a set of basic FSM patterns having a high level functional meaning together. Instead of being restricted to a single functional process (the particularity of a basic FSM pattern), a composite FSM pattern combines several functional processes. These functional processes have the characteristic of sharing the same primary data group in their data movements, along with several other data groups. For example, one common composite FSM pattern that can be defined in IS is the CRUD (Create, Retrieve, Update, Delete), or CRUDL (L is added for List) of a master data group (ex. "Customer"), that may also use other data groups (ex. "Sales rep", "Sales territory", and "Error message").

#### d) Multi-composite FSM pattern

A **Multi-composite FSM pattern** applies to a set of composite and basic patterns having functional relationships among them. A multi-composite FSM pattern combines multiple functional processes handling several data groups within the software being measured. It could apply to the handling of master, transactional, and reference data combined, all within the defined measurement scope. In IS, a multi-composite FSM pattern could represent a whole module, or even a whole software. In embedded/real-time systems, it could be the set of back-end subsystem functionalities for a family of devices.

#### 4) FSM patterns overview

FSM pattern categories can be viewed partly embedded since categories at one level can combine FSM patterns of lower levels (Fig. 3). Also, FSM patterns in every category can be defined as unique in a given context, without ever being part of another FSM pattern in another level.

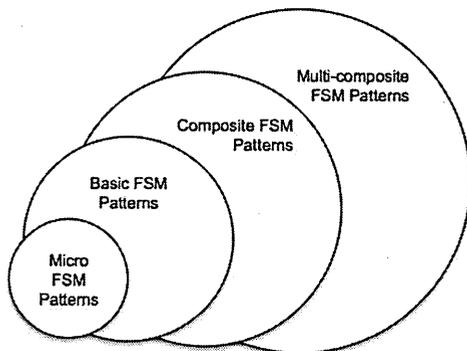


Fig. 3. Hierarchical representation of FSM patterns.

The following section provides specific examples for all four FSM pattern categories.

#### D. Demonstration of FSM Patterns

The chosen approach to demonstrate FSM patterns concepts is to present a series of practical examples of measurements. Since patterns can be presented in many forms as related in [33], the essential three sections of the Alexandrian form were selected as suggested by Riehle & Züllighoven [34]. The main reasons for presenting the FSM patterns using a limited set of Alexandrian form sections are that 1) FSM patterns definition is still a work in progress and 2) there is a need to keep the representation as simple as possible at this moment.

These three essential sections consist of:

- **Problem:** Describe succinctly the problem the pattern is about to solve;
- **Context:** Describe the circumstance when that problem occurs;
- **Solution:** Describe how to solve the problem in this specific context.

Furthermore, to be able to distinguish FSM patterns among each other, a unique name will be provided to each FSM Pattern, in accordance with the Alexandrian form to describe a pattern.

In short, each FSM patterns will be described by using this template form:

- Pattern Name
- Problem
- Context
- Solution (Generic COSMIC software model representation)

By using a such template, it will be easier to organize FSM patterns.

#### E. Evaluation of FSM patterns

We have applied FSM patterns in both IS and embedded real-time domains, mainly in training activities and to alleviate the measurement recording effort. Due to confidentiality issues, no explicit details of their usage can be reported. At this point in our research, FSM patterns and their usage have not yet been quantitatively evaluated against the solution objectives. Case studies and experimentations will need to be carried out. We intend to pursue our research with evaluation means and are hopeful to inspire other researchers to pursue work on FSM patterns.

#### F. Communication of FSM pattern concepts

This paper introduces the concepts of FSM patterns as the first means of communication. As research will progress on FSM patterns, it is expected that other communication means be used. The "Future work" section of this paper discusses the matter in greater details.

### III. DEMONSTRATION OF FSM PATTERNS WITH EXAMPLES

This section will succinctly demonstrate some FSM patterns for each category and illustrate them by providing examples. Micro, basic, composite and multi-composite FSM patterns will be addressed in this order; from the simplest to the most complex FSM pattern.

#### A. Micro FSM pattern examples

As mentioned earlier, a Micro FSM pattern applies to a fragment of a functional process, involving one or several data groups. The following examples are presented, along with their problem and context definitions:

- Display simple error messages;
- Display localized error messages;
- Asynchronous data sending to an external service;
- Data sending to an external service with confirmation;
- External service request with data group or error code return.

#### **Pattern Name:** Display simple error messages

**Problem:** How to measure a FUR that displays one or several occurrences of error messages?

**Context:** For one or several validations within functional processes, the FUR may generically state that “when [such error condition] occurs, the [system] displays an error message (1 eXit)”.

**Solution:** For Display simple error messages.

Functional Process	Data Group	Data Movements	Funct. Size (in CFP)
<Functional process>	Error message	X	1
<b>Total:</b>			<b>1</b>

#### **Pattern Name:** Display localized error messages

**Problem:** How to measure a FUR that displays one or several occurrences of error messages in the current language previously selected by the user?

**Context:** For one or several validations within functional processes, the FUR may generically state that “when [such error condition] occurs, the [system] reads the message #n (1 Read) and displays this error message in the user selected language (1 eXit)”.

**Solution:** For Display localized error messages; the selected language is part of the user profile in that case.

Functional Process	Data Group	Data Movements	Funct. Size (in CFP)
<Functional process>	User profile	E	1
	Error message	RX	2
<b>Total:</b>			<b>3</b>

#### **Pattern Name:** Asynchronous data sending to an external service

**Problem:** How to measure a FUR that sends data to an external service in an asynchronous manner?

**Context:** In a distributed architecture system with services (e.g. Web service or SOA type of service) or integration with external systems, the FUR may generically state that a “[functional process] sends a [data group] to [service A]” (1 eXit). [Service A] is external to the piece of software being measured. Later, [service A] will process the <data group> with no instant return to the [functional process] which has sent the data. In that context, [service A], or at least the piece of software in which it resides, is a functional user of the [functional process].

**Solution:** For asynchronous data sending to an external service.

Functional Process	Data Group	Data Movements	Funct. Size (in CFP)
<Functional process>	<Data group>	X	1
<b>Total:</b>			<b>1</b>

#### **Pattern Name:** Sending data to an external service with confirmation

**Problem:** How to measure a FUR that sends data to an external service and expects a confirmation code from that service (e.g. success or failure) before continuing processing?

**Context:** In a distributed architecture system with services, the FUR may generically state that a “[Functional process] sends a <data group> to [service A] (1 eXit) and the processing will wait until it receives a confirmation code from [service A] (1 Entry)”. The confirmation code may be of different values that the [functional process] will need to process upon reception.

**Solution:** For sending data to an external service with reception of confirmation code.

Functional Process	Data Group	Data Movements	Funct. Size (in CFP)
<Functional process>	<Data group>	X	1
	Confirmation code	E	1
<b>Total:</b>			<b>2</b>

#### **Pattern Name:** External service request with data group or error code return

**Problem:** How to measure a FUR that sends a request to an external service and expects a data group or an error message in return?

**Context:** In a distributed architecture system with services, the FUR may generically state that a “[functional process] sends a <data request> (1 eXit) to [service A] and receives either occurrence(s) of <data group> in case of success (1 Entry) or an error code in case of failure (1 Entry)”. It is later

expected that the [functional process] provides adequate processing for both success and failure possibilities.

**Solution:** For an external service request with data group or error code return.

Functional Process	Data Group	Data Movements	Funct. Size (in CFP)
<Functional process>	<data request>	X	1
	<data group>	E	1
	Error message	E	1
<b>Total:</b>			<b>3</b>

The Error message data group refers to a distinct object of interest. This specific FSM pattern can easily be scaled up with other data groups being sent by the requested service. As an example, a functional process may request various data from a provided account number, where it will receive the following data groups with as many Entries: Customer data, Account data, Customer Addresses, and List of other accounts. In any cases, there will only be one Entry for the Error message data group.

#### B. Basic FSM pattern examples

To illustrate basic FSM patterns, the following case will be used as the generic example: the software being measured manages a data group composed of one or several data attributes. This data group could be a simple reference data such as "Sales territories", "Countries", "States/Provinces", "Product types". The data group could also be some specific master data. The type of data group becomes irrelevant for measurement as most functional processes intended to manage these data groups will behave the same way.

Three basic FSM pattern examples are presented here: Basic Create, Basic Retrieve, Basic Update, and Basic Delete.

#### Pattern Name: Basic Create

**Problem:** How to measure a FUR that involves the persistence of a data group with a simple data existence validation?

**Context:** It is common in IS that data needs to be saved to be later used. The FUR may generically state that "the [functional user] enters data about the <data group> (1 Entry) and saves the occurrence; the [system] ensures that this occurrence does not already exist (1 Read) and persists the occurrence (1 Write); If the occurrence can't be persisted, an error message is displayed (1 eXit)".

**Solution:** For Basic Create

Functional Process	Data Group	Data Movements	Funct. Size (in CFP)	Remark
Create <data group>	<data group>	ERW	3	Creates a new occurrence
	Error message	X	1	
<b>Total:</b>			<b>4</b>	

#### Pattern Name: Basic Retrieve

**Problem:** How to measure a FUR that involves a retrieve to check what has been done for one or many data groups?

**Context:** It is common in IS that data needs to be retrieved to know what has been create, delete or update in a database. The FUR may generically state that "the [functional user] selects an occurrence of <data group> (1 Entry) to display. The [system] reads the selected <data group> (1 Read) and displays its attributes (1 eXit); If the occurrence can't be seen, an error message is displayed (1 eXit)".

**Solution:** For Basic Retrieve.

Functional Process	Data Group	Data Movements	Funct. Size (in CFP)	Remark
Retrieve <data group>	<data group>	ERX	3	Select, read and display existing occurrence
	Error message	X	1	
<b>Total:</b>			<b>4</b>	

#### Pattern Name: Basic Update

**Problem:** How to measure a FUR that involves the update of a data group with a data validation?

**Context:** It is common in IS that persisted data needs to be modified, then saved again to be later used. In case the data can't be updated, an error message will be generated. The FUR may generically state that "the [system] reads the selected occurrence of <data group> (1 Read); the [functional user] updates the value of the <data group> (1 Entry) and save the occurrence; the [system] persists the occurrence (1 Write); if the occurrence can't be persisted, the [system] displays an error message (1 eXit)".

**Solution:** For Basic Update

Functional Process	Data Group	Data Movements	Funct. Size (in CFP)	Remark
Update <data group>	<data group>	ERW	3	Update existing occurrence
	Error message	X	1	
<b>Total:</b>			<b>4</b>	

It is important to note that this FSM pattern example, as any other FSM pattern, is context specific. User interface ergonomic guidelines, stated or implied, should drive such context when applicable, and FSM patterns should be defined accordingly.

**Pattern Name:** Basic Delete

**Problem:** How to measure a FUR that deletes a data group and expects to display a confirmation message and a potential error message?

**Context:** It is common in IS that persisted data needs to be deleted from the storage when no longer useful. The FUR may generically state that “the [functional user] selects an occurrence of <data group> (1 Entry) to be deleted; the [system] prompts a confirmation message, waiting for the user confirmation (no data movement here, as it is considered as navigation); on confirmation (no data movement, considered as navigation), the [system] finds the occurrence in the storage (1 Read) and delete the occurrence (1 Write); upon successful delete of the occurrence, the [system] displays a confirmation [message] (1 eXit); if the occurrence can’t be deleted, the [system] displays an error [message] (same eXit than the confirmation message as the object of interest here is the [message], not its different occurrences)”.

**Solution:** For Basic Delete.

Functional Process	Data Group	Data Movements	Funct. Size (in CFP)	Remark
Delete a <data group>	<data group>	ERW	3	Delete an occurrence, reads it 1st
	Message	X	1	Confirmation or error
<b>Total:</b>			<b>4</b>	

**C. Composite FSM pattern example**

To illustrate composite FSM patterns, the following case will be used as the generic example: the software being measured manages a master data group composed of several data attributes, some of which are linked to other master or reference data groups. That would be the case to manage “Customers” when an attribute must be linked to the “Sales Rep” master data group and another attribute be linked to the “Customer Category” reference data group. The software needs to allow Create, Retrieve, Update, Delete and List operations on “Customer” (commonly identified as CRUDL). The same set of functional processes with the same number of data groups may also apply on the “Product” master data group, where a “Product” attribute must be linked to the “Product type” reference data group and another attribute must be linked to the “Price and Discount List” master data group.

**Pattern Name:** Composite CRUDL-3DG

**Problem:** How to measure a group of FUR (e.g. CRUDL) related to a given data group (DG) having the same functional processes as another similar set of FUR, but handling also two other DGs?

**Context:** It is common in IS that FUR are repeated many times with variations of DGs while the expected software behaviour is the same. The FUR, at a higher level of decomposition than preceding examples, may generically state that “the [systems] is required to Create, Retrieve, Update,

Delete, and List occurrences of <First DG>. A <First DG> must be linked with an existing <Second DG> [for attribute X] and to an existing <Third DG> [for attribute Y], chosen from a list on screen.

**Solution:**

Functional Process	Data Group	Data Mvnts	Funct. Size (in CFP)	Remark
Create <First DG>	<First DG>	ERW	3	Create new occurrence
	<Second DG>	RX	2	Read and display list
	<Third DG>	RX	2	Read and display list
	Error message	X	1	<b>Subtotal: 8 CFP</b>
Retrieve <First DG>	<First DG>	ERX	3	Select, read and display existing occurrence
	<Second DG>	RX	2	Must read its ID to display its name
	<Third DG>	RX	2	Same as above
	Error message	X	1	<b>Subtotal: 8 CFP</b>
Update <First DG>	<First DG>	ERW	3	Update existing occurrence
	<Second DG>	RX	2	Read and display list
	<Third DG>	RX	2	Read and display list
	Error message	X	1	<b>Subtotal: 8 CFP</b>
Delete a <First DG>	<First DG>	ERW	3	Delete an occurrence, reads it 1st, no other DG required
	Message	X	1	<b>Subtotal: 4 CFP</b>
List <First DG>	<First DG>	RX	2	Read and display list
	Filter	<b>E</b>	1	Search filter applicable to all DGs
	<Second DG>	RX	2	Read/display list (filter)
	<Third DG>	RX	2	Same as above
	Error message	X	1	<b>Subtotal: 8 CFP</b>
<b>Total:</b>			<b>36</b>	For this FSM pattern

<sup>a</sup> Legend: E=Entry; X=eXit; R=Read; W=Write; The “E” in bold represents the triggering Entry of its functional process.

When replacing <First DG> by “Customer”, the <Second DG> by “Sales Rep”, and the <Third DG> by “Customer Category”, the resulting COSMIC software model becomes an adequate representation of the FUR for all five related functional processes. The same applies when replacing <First DG> by “Product”, <Second DG> by “Price and Discount List”, and <Third DG> by “Product type”. This composite FSM pattern applies for any other given data group for which the software is required to process these five functional processes with 2 other DGs.

These five operations could represent five distinct functional processes in COSMIC. They also represent five basic FSM patterns. When measuring relevant functional processes for a specific reference data group, such as "Customer Category", the same five functional processes apply, while the functional size will be smaller as this other CRUDL does not handle as many data groups (in fact, there will be only <First DG>, <Error message>, and <Filter>). Therefore, the five basic FSM patterns apply again, but with another set of data groups and related movements. Therefore, basic FSM patterns can be grouped together to form a new pattern that can be used again and again, by referring to another data group.

CRUD or CRUDL are far from being the only possible examples of composite FSM pattern. More elaborated composite FSM patterns can be defined, such as "Ordering a new credit card" from a bank's website, which involves a series of functional processes and a rather large set of DGs. It could also be the set of functionalities required to "Manage the Audio Feature" in a modern car from the provided touch screen in an embedded/real-time context.

#### D. Multi-composite FSM pattern example

A multi-composite FSM pattern could represent a whole module, such as "Inventory Control", or even a whole software, such as an "Accounting system" that would comprise "General Ledger", "Financial Statements", "Accounts Receivables", and "Accounts Payables" modules. Every module can then be defined as a series of composite FSM patterns, plus a series of basic FSM patterns for specific types of functionalities.

With reusing some of the preceding examples and adding new ones, suppose a new "Accounts Receivables" module, with two master DGs from the A/R module (Customer and Sales Rep), one master DG from the GL module (GL account), five reference DGs (Customer category, Account aging parameters, Invoicing parameters, and Cash receipt parameters, Journals list from GL module), and four transactional DGs (Invoice, Cash receipt, Sales journal transaction, Cash receipt journal transaction). The whole pattern will not be detailed but summarized as follows.

#### Pattern Name: Multi-composite module-3DG

**Problem:** How to measure a whole module related to two primary DGs having the same functional processes as another similar module, but handling also one secondary master, three reference, and four transactional DGs?

**Context:** It is common in IS that popular modules are repeated many times with variations of DGs while the expected software behaviour is basically the same. The FUR, at a module level of decomposition, may generically state that "the [systems] is required to "Manage" occurrences of <DG1> and <DG2> ("Manage" being equivalent to CRUDL in this case), and also "Manage" 2nd, 3rd, and 4th reference DGs", and so on until all expected functionalities for that module have been described.

**Solution:** Please note that is only a summary view for such module type.

FSM Pattern	Category	Funct. Size (in CFP)	Remark
CRUDL-3DG	Composite	36	Ex. for "Customer"
CRUDL-1DG	Composite	20	Ex. for "Sales Rep"
CRUDL-1DG	Composite	20	Ex. for "Customer category"
CRUD-2DG	Composite	22	Ex. for "Account aging parameters"
CRUD-3DG	Composite	26	Ex. for "Invoicing parameters"
CRUD-3DG	Composite	26	Ex. for "Cash receipt (C/R) parameters"
Transaction-7DG	Basic	12	Ex. for "Enter manual invoices"
Transaction-6DG	Basic	10	Ex. for "Enter a manual cash receipt"
Transaction-8DG	Basic	14	Ex. for "Enter adjustment on Invoice or C/R"
Report-3DG	Basic	7	Ex. for "Report on customer sales"
Report-4DG	Basic	9	Ex. for "Customer aging report"
Report-5DG	Basic	11	Ex. for "Customer statement of account"
Milestone-2DG	Basic	10	Ex. for "End of month A/R processing"
	<b>Total:</b>	<b>223</b>	<b>For this FSM pattern</b>

And there could be a very large variety of multi-composite FSM patterns to be defined, probably as many as there are different software contexts.

#### IV. CONCLUSION AND FUTURE WORK

##### A. Conclusion

This paper introduced the concepts of FSM patterns as a means to help inexperienced measurer to ramp-up their learning curve faster, to avoid measurement mistakes, to reduce measurement effort and to assist in measuring projects at an early phase.

The concept of FSM pattern was designed and developed by applying a DSRM framework. Thereby, the problems involving FSM were first exposed. These problems led to motivation related to training novices, optimizing measurement effort, ensuring quality of measurement results and estimating projects more accurately in their early phases. From problems and motivations, the following four objectives were defined: (1) The solution must help an inexperienced measurer to apply the rules governing the COSMIC method; (2) The solution should contribute in reducing measurement effort; (3) The solution should allow for an early sizing, even without detailed written FUR; (4) The solution should provide a more accurate sizing by helping to avoid common measurement mistakes.

Based on COSMIC expert experiences, supported by researches on patterns [32][33][34] and especially inspired by design patterns [33], the concept of FSM pattern was designed and developed, and FSM patterns were categorized (Micro, Basic, Composite and Multi-Composite). A pattern form to

describe FSM patterns was proposed. Finally, the FSM pattern concept was demonstrated through a series of examples.

The discovery and the definition of FSM patterns arise many new opportunities in software measurement. One of the most hopeful opportunities is the development of tools supporting FSM patterns. In addition to FSM patterns, domain specific FSM patterns, derived or not from FSM patterns provided by a tool, could be created by measurers (i.e. the FSM tool users) to meet their own specific measurement needs. By using better measurement support tools, one hypothesis is that inexperienced measurers will be able to adequately size software by having an environment to guide them with the support of FSM patterns.

Another hypothesis is that the measurement effort will be reduced with the usage of FSM patterns but also by generating COSMIC compliant measurement records instead of having to type them manually while measuring a piece of software. Also, supported by such efficient tools, FSM popularity could increase by allowing novices to use the tool and FSM patterns to measure software. In a research perspective, the utilization of software tools supporting FSM patterns will generate measurement data that could be analysed to evaluate the effectiveness of FSM patterns or to discover complex new patterns.

Furthermore, if evaluation activities or measurer communities later confirm the value of FSM patterns, then the opportunity can be taken by the COSMIC organization to include this new concept as a part of the COSMIC Measurement Pattern Guideline [35]. More specifically, this guideline could include detailed FSM patterns on top of measurement strategy patterns.

Micro and basic FSM patterns should be included in training material for apprentice measurers, to guide them in applying the COSMIC rules and provide them with some basic know-how in achieving correct measures. Composite FSM patterns should be useful to accelerate measurement. Multi-composite FSM patterns should allow to make projections of what should be expected out of analysis, but also as an input to high-level estimates, quite early in the software lifecycle.

When using approximation techniques for sizing at an early stage, a person "estimates" the software size obtaining a number of CFP, often without details [23]. By applying FSM patterns, the same person should obtain detailed measurement recordings, which can later be verified and updated against written requirements, if any, or the developed software.

#### B. Future Work

Since the FSM pattern concept is new and still at a conceptual stage, the objectives achievement stays theoretical. Further research is required to validate the defined aspects of the problem, such as to confirm the achievement of objectives defined in section II-B, but also to explore new fields of research that can be derived from FSM patterns. Specifically, experiments will need to be carried out to evaluate the effectiveness of FSM patterns on the learning curve of inexperienced COSMIC measurers and their ability to avoid measurement mistakes.

At this point, the following two assumptions need to be evaluated:

- Recurring FSM patterns represent up to 80% of measurement cases in some contexts. Therefore, measuring the remaining 20% of cases requires expertise.
- FSM patterns can help to understand the COSMIC rules.

Other FSM patterns need to be investigated regarding data movements due to validation within functional processes, as these movements can have a substantial impact on the resulting measures. The same applies to the "report" type and "batch" type of functional processes, where novices could learn from well-defined FSM patterns.

Finally, work could also be done in defining a set of standard COSMIC FSM patterns that should be made freely available to the measurement community, along with a measurement-recording tool supporting these FSM patterns.

#### ACKNOWLEDGMENT

Special thanks to Stéphane Pineault and his team at EXFO for inspiring the concepts of FSM patterns in the context of embedded/real-time software. Special thanks to Michel Martel and his team at Analytik for inspiring FSM patterns examples in an IS context.

#### REFERENCES

- [1] Fairley, R (2009). *Managing and leading software projects*. New Jersey : Wiley-IEEE Computer Society Press.
- [2] Albrecht, A (1979). *Measuring Application Development Productivity*. Proceeding of the Joint SHARE/GUIDE/IBM Application Development Symposium, pp. 83-92.
- [3] ISO/IEC 14143-1:1998 (1998) *Information technology - Software measurement - Functional size measurement - Part 1: Definition of Concepts, JTC 1 / SC 7*, Geneva: International Organisation for Standardization.
- [4] ISO/IEC 20926:2009 (2009). *Software and systems engineering – Software measurement – IFPUG functional size measurement method*. Geneva: International Organisation for Standardization.
- [5] ISO/IEC 20968:2002 (2002). *Software engineering – Mk II Function Point Analysis – Counting Practices Manual*. Geneva: International Organisation for Standardization.
- [6] ISO/IEC 24570:2005 (2005). *Software engineering – NESMA functional size measurement method version 2.1 – Definitions and counting guidelines for the application of Function Point Analysis*. Geneva: International Organisation for Standardization.
- [7] ISO/IEC 29881:2010 (2010). *Information technology – Systems and software engineering – FISMA 1.1 functional size measurement method*. Geneva: International Organisation for Standardization.
- [8] ISO/IEC 19761:2011 (2011). *Software engineering – COSMIC: a functional size measurement method*, Geneva: International Organisation for Standardization.
- [9] Abran, A. et al. (2015). *The COSMIC Functional Size Measurement Method – Measurement Manual, version 4.0.1*, The COSMIC group, April 2015, available from : <http://www.cosmic-sizing.org>.
- [10] Abran, A. (2010). *Software Metrics and Software Metrology*, Wiley.
- [11] Boisvert, M. and Trudel, S. (2011) *Choisir l'agilité: Du développement logiciel à la gouvernance*. Paris: Dunod. p. 240.

- [12] Ugan, E., Demirörs, O., Top, Ö. Ö., & Özkan, B. (2009). An experimental study on the reliability of COSMIC measurement results. In *Software Process and Product Measurement* (pp. 321-336). Berlin Heidelberg: Springer.
- [13] Trudel, S., & Abran, A. (2009). Functional size measurement quality challenges for inexperienced measurers. In *Software Process and Product Measurement* (pp. 157-169). Berlin Heidelberg: Springer.
- [14] Dumas-Monette, J.-F., and Trudel, S. (2014). Requirements Engineering Quality Revealed Through Functional Size Measurement: A Case Study in an Agile Context, *IWSM*
- [15] Jenner, M. S. (2016). 5.1 Automation of Counting of Functional Size Using COSMIC FFP in UML. *COSMIC Function Points: Theory and Advanced Practices*, 276.
- [16] Cohn, M. (2004). User stories applied: For agile software development. Addison-Wesley Professional.
- [17] Peffers, K., Tuunanen, T., Rothenberger, M.A. and Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research, *Journal of Management Information Systems / Winter 2007-8*, Vol. 24, No. 3, pp. 45-77.
- [18] Desharnais, J.-M., (2003). Application de la mesure fonctionnelle COSMIC-FFP: une approche cognitive, in *Informatique*. Montreal: UQAM, pp. 187.
- [19] Desharnais, J.M., Abran, A., Mayers, A., Buglione, L., and Bevo, V. (2002). Knowledge Modeling for the Design of a KBS in the Functional Size Measurement Domain, presented at KES 2002, Crema, Italy.
- [20] Jenner, M.S.: Automation of Counting of Functional Size Using COSMIC-FFP in UML(2002). In: 12th International Workshop Software Measurement, pp. 43-51
- [21] Marin, B., Giachetti, G., Pastor, O. (2008): Automating the Measurement of Functional Size of Conceptual Models in a MDA Environment. In: Jedlitschka, A., Salo, O. (eds.) PROFES 2008. LNCS, vol. 5089, pp. 215-229. Springer, Heidelberg
- [22] Pastor, O., Gómez, J., Insfrán, E., Pelechano, V. (2001): The OO-Method Approach for Information Systems Modelling: From Object-Oriented Conceptual Modeling to Automated Programming. *Information Systems* 26, 507-534
- [23] Vogezang, F.W. et al. (2015). Guideline for Early or Rapid COSMIC Functional Size Measurement by Using Approximation Approaches, The COSMIC Group, June 2015, available from <http://www.cosmic-sizing.org>.
- [24] Desharnais, J.-M. (1996). Post-Measurement Validation Procedures of Function Point Counts, in 6th International Workshop on Software Metrics, University of Regensburg, Germany.
- [25] Desharnais, J.-M. and Morris, P. (1996). Validation Process in Software Engineering: an Example with Function Points, in Forum on Software Engineering Standards (SES'96), Montreal, available from: <http://www.totalmetrics.com/function-points-downloads/Function-Points-Validation-Auditing.pdf>
- [26] Chisholm, M. (2006) Master Data versus Reference Data, *Information Management Magazine*, Information Management, April 2006.
- [27] Otto, B., Hüner, K.M., and Österle, H. (2011). Toward a functional reference model for master data quality management, *Information Systems and E-Business Management*, Springer-Verlag, DOI 10.1007/s10257-011-0178-0.
- [28] Otto, B., and Hüner, K.M. (2009). Functional Reference Architecture for Corporate Master Data Management, Institute of Information Management, University of St. Gallen, Switzerland, Report number: BE HSG/CC CDQ/21, version 1.0, May 31st 2009.
- [29] Abran, A. et al. (2011). The COSMIC Functional Size Measurement Method: Guideline for assuring the accuracy of measurements, version 1.0, The COSMIC Group, available from <http://www.cosmic-sizing.org>.
- [30] Desharnais, J.M., Abran A. (2010). Assessment of the Quality of Functional user requirements documentation using criteria derived from measurement with COSMIC – ISO 19761, *IWSM Metrikon* 2010.
- [31] Kocatürk, B., Desharnais, J.M., Abran A. (2011), Using the COSMIC Method to Evaluate the Quality of the Documentation of Agile User Stories, Joint Conf of 21st Int'l Workshop on Software Measurement and the 6th Int'l Conference on Software Process and Product Measurement, *IWSM/Mensura* 2011, Nara, Japan, November 3-4, 2011.
- [32] Christopher Alexander, Sara Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdahl-King, and Shlomo Angel (1977). *A Pattern Language*. New York: Oxford University Press.
- [33] Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1993). *Design Patterns: Abstraction and Reuse of Object-Oriented Design*. ECOOP '93, LNCS-707, Conference Proceedings (pp. 406-431). Berlin-Heidelberg: Springer-Verlag.
- [34] Riehle, D., & Züllighoven, H. (1996). Understanding and using patterns in software development. *TAPOS*, 2(1), 3-13.
- [35] Symons, C.R. (2013). Guideline for 'Measurement Strategy Patterns': Ensuring that COSMIC size measurements may be compared, version 1.0, The COSMIC group, March 2013, available from: <http://www.cosmic-sizing.org>.



## ANNEXE B.

### RÉSULTAT DES REQUÊTES DE LA RSL

Dans le tableau B.1, la première colonne « ID » représente un identifiant unique correspondant à la requête. Cette information a été utile pour lier chacun des articles trouvés à la requête correspondante. La seconde colonne, comme son nom l'indique, correspond à la base de données interrogée. La troisième colonne contient le filtre utilisé pour la date. En effet, la majorité des bases de données interrogées permettait de filtrer l'information entre autres par date. La quatrième colonne liste les mots clefs utilisés pour former la requête. La cinquième colonne représente la requête dans le format utilisé par le moteur de recherche de la base de données interrogée. Cette information permet, entre autres, de reproduire le résultat des recherches de cette revue systématique en tenant compte du fait que de nouveaux articles ont pu être publiés depuis notre recherche initiale. Ces ajouts pourraient expliquer une disparité entre le nombre d'articles trouvés. Enfin, la dernière colonne intitulée « NB » correspond au nombre d'articles obtenus avec la requête exécutée.

Tableau B.1. – Résultat des requêtes

ID	Base de données	Filtre	Mots clefs	Requête	NB
ACM6	ACM	depuis 2003	COSMIC, Functional size measurement, Tool	(+COSMIC, +Functional +size +measurement, +Tool)	15
ACM7	ACM	depuis 2003	COSMIC, FSM, Tool	(+COSMIC, +FSM, +tool)	6
DES1	Digital Scholary Editions	COSMIC	COSMIC	0	

ID	Base de données	Filtre	Mots clefs	Requête	NB
GCL1	ACM - Guide to computing literature	depuis 2003	COSMIC, Functional size measurement, Tool	(+COSMIC, +Functional +size +measurement, +tool)	39
SCO1	Scopus	depuis 2003	COSMIC, Functional size measurement, Tool	( ALL ( cosmic ) AND ALL ( tool ) AND ALL ( "Functional Size Measurement" ) ) AND PUBYEAR > 2002	161
SCO2	Scopus	depuis 2003	COSMIC, FSM, Tool	ALL ( cosmic ) AND ALL ( tool ) AND ALL ( fsm ) ) AND PUBYEAR > 2002	64
SCO3	Scopus	depuis 2003	COSMIC, Functional size measurement, Tool	( TITLE-ABS-KEY ( cosmic ) AND TITLE-ABS-KEY ( tool ) AND TITLE-ABS-KEY ( "Functional Size Measurement" ) ) AND PUBYEAR > 2002	26
SCO4	Scopus	depuis 2003	COSMIC, FSM, Tool	( TITLE-ABS-KEY ( cosmic ) AND TITLE-ABS-KEY ( tool ) AND TITLE-ABS-KEY ( fsm ) ) AND PUBYEAR > 2002	10
SCI1	ScienceDirect	depuis 2003	COSMIC, FSM, Tool	pub-date > 2002 and COSMIC AND Tool AND FSM	50

ID	Base de données	Filtre	Mots clefs	Requête	NB
SCI2	ScienceDirect	depuis 2003	COSMIC, Functional size measurement, Tool	pub-date > 2002 and COSMIC AND TOOL AND "Functional Size Measurement"	37
WOJ01	Whiley Online Library	depuis 2003	COSMIC, FSM, Tool	COSMIC(all Fields) AND Tool(all Fields) AND FSM(all Fields), Between 2003 and 2018	13
WOJ02	Whiley Online Library	depuis 2003	COSMIC, Functional size measurement, Tool	COSMIC(all Fields) AND Tool(all Fields) AND "Functional Size Measurement"(all Fields), Between 2003 and 2018	19
IEEE01	IEEE XPLORE	entre 2003 et présent	COSMIC, FSM, Tool	(Full Text & Metadata" : COSMIC AND Tool AND FSM	180
IEEE02	IEEE XPLORE	entre 2003 et présent	COSMIC, Functional size measurement, Tool	(Full Text & Metadata" : COSMIC AND Tool AND "Functional size measurement"	162
SPL01	Springer Link	entre 2003 et 2018	COSMIC, Functional size measurement, Tool	COSMIC AND Tool AND "Functional size measurement"	92
SPL02	Springer Link	entre 2003 et 2018	COSMIC, FSM, Tool	COSMIC AND TOOL AND FSM	89



# ANNEXE C.

## APERÇU DU EXCEL RÉSULTANT DE LA RSL

Indexer ID	Indexer Name	Type	Filter	Keyword	Results Found
ACM1	ACM	Basic Search	Pub Since 2002	CCSMIC, Functional size measurement, software	144312 Page
ACM2	ACM	Basic Search	Pub Since 2002	CCSMIC, Functional size measurement, tool	136205 Page
ACM3	ACM	Basic Search	Pub Since 2002	CCSMIC, Functional size measurement, tool for software estimation	309346 Page
ACM4	ACM	Basic Search	Pub Since 2002	CCSMIC, Functional size measurement, prototype for effort estimation	296897 Page
ACM5	ACM	Advanced Search	Pub Since 2002	CCSMIC, Functional size measurement	212 Measurement
ACM6	ACM	Advanced Search	Pub Since 2002	CCSMIC, Functional size measurement, software for effort estimation	15,6000
ACM7	ACM	Advanced Search	Pub Since 2002	CCSMIC, Functional size measurement	9,6000
DB11	Digit of Software Estimation	CTR of page by Education		CCSMIC	2 Sample Page
SC01	ACM - Guide to computing literature	Advanced Search	Pub Since 2002	CCSMIC, Functional size measurement, tool	88 Measur
SC02	Search	All Fields	Pub Since 2002	(All (same) AND All (tool) AND All ("Functional Size Measurement")) AND PUBLYEAR > 2002	142 Measur, Year >
SC03	Search	All Fields	Pub Since 2002	(All (same) AND All (tool) AND All (tool)) AND PUBLYEAR > 2002	68 Measur, Year >
SC04	Search	Article Title, Abstr	Pub Since 2002	(TITLE-ABS-KEY (ccsmic) AND TITLE-ABS-KEY (tool) AND TITLE-ABS-KEY ("Functional Size Measurement")) AND PUBLYEAR > 2002	24 Measur, Year >
SC05	Search	Article Title, Abstr	Pub Since 2002	(TITLE-ABS-KEY (ccsmic) AND TITLE-ABS-KEY (tool) AND TITLE-ABS-KEY (tool)) AND PUBLYEAR > 2002	15 Good, Year >
SC06	Search	Abstract	Pub Since 2002	CCSMIC AND Tool AND Year	108 Measur
SC07	Search	Abstract	Pub Since 2002	Pub Since > 2002 AND CCSMIC AND TOOL AND "Functional Size Measurement"	17 Measur
SC08	Whitney Online Library - Journals	Advanced Search		CCSMIC AND Functional Family AND Change Family Between 2002 and 2012	12 Page
SC09	Whitney Online Library - Journals	Advanced Search		CCSMIC AND Functional Family AND Functional Size Measurement Tool Fields Between 2002 and 2012	18 Page
SC10	Whitney Online Library - Book Review	Advanced Search		CCSMIC AND Functional Family AND Functional Size Measurement Tool Fields Between 2002 and 2012	18 Page

FIGURE C.1. – Excel RSL : onglet Métrique

Indexer ID	Indexer Name	Type	Filter	Keyword	Results Found
ACM1	ACM	Basic Search	Pub Since 2002	CCSMIC, Functional size measurement, software	144312 Page
ACM2	ACM	Basic Search	Pub Since 2002	CCSMIC, Functional size measurement, tool	136205 Page
ACM3	ACM	Basic Search	Pub Since 2002	CCSMIC, Functional size measurement, tool for software estimation	309346 Page
ACM4	ACM	Basic Search	Pub Since 2002	CCSMIC, Functional size measurement, prototype for effort estimation	296897 Page
ACM5	ACM	Advanced Search	Pub Since 2002	CCSMIC, Functional size measurement	212 Measurement
ACM6	ACM	Advanced Search	Pub Since 2002	CCSMIC, Functional size measurement, software for effort estimation	15,6000
ACM7	ACM	Advanced Search	Pub Since 2002	CCSMIC, Functional size measurement	9,6000
DB11	Digit of Software Estimation	CTR of page by Education		CCSMIC	2 Sample Page
SC01	ACM - Guide to computing literature	Advanced Search	Pub Since 2002	CCSMIC, Functional size measurement, tool	88 Measur
SC02	Search	All Fields	Pub Since 2002	(All (same) AND All (tool) AND All ("Functional Size Measurement")) AND PUBLYEAR > 2002	142 Measur, Year >
SC03	Search	All Fields	Pub Since 2002	(All (same) AND All (tool) AND All (tool)) AND PUBLYEAR > 2002	68 Measur, Year >
SC04	Search	Article Title, Abstr	Pub Since 2002	(TITLE-ABS-KEY (ccsmic) AND TITLE-ABS-KEY (tool) AND TITLE-ABS-KEY ("Functional Size Measurement")) AND PUBLYEAR > 2002	24 Measur, Year >
SC05	Search	Article Title, Abstr	Pub Since 2002	(TITLE-ABS-KEY (ccsmic) AND TITLE-ABS-KEY (tool) AND TITLE-ABS-KEY (tool)) AND PUBLYEAR > 2002	15 Good, Year >
SC06	Search	Abstract	Pub Since 2002	CCSMIC AND Tool AND Year	108 Measur
SC07	Search	Abstract	Pub Since 2002	Pub Since > 2002 AND CCSMIC AND TOOL AND "Functional Size Measurement"	17 Measur
SC08	Whitney Online Library - Journals	Advanced Search		CCSMIC AND Functional Family AND Change Family Between 2002 and 2012	12 Page
SC09	Whitney Online Library - Journals	Advanced Search		CCSMIC AND Functional Family AND Functional Size Measurement Tool Fields Between 2002 and 2012	18 Page
SC10	Whitney Online Library - Book Review	Advanced Search		CCSMIC AND Functional Family AND Functional Size Measurement Tool Fields Between 2002 and 2012	18 Page

FIGURE C.2. – Excel RSL : onglet Papiers

Authors	Title	Year	Source Title	Volume	Issue	Art. No.	Page start	Page end	Page count	DOI	Link	Abstract	Author Key	Index	Keywords	ED
Lusky M., Po	Software cost	2018	Advances in	998			51	62	12	10.1007/978-https://www.springer.com/978-3-319-85026-4_909						
Sehn S.K., B	Research pap	2017	Information	91			1	21	21	10.1016/j.infhttps://www.computer-science.org/2017/05/001						
Quezada-Ló	Towards an	2017	ACM Internat	Part F131936			138	144	7	10.1145/314https://www.functional.si.computer-science.org/2017/05/001						
Soubre H., A	Functional si	2017	ACM Internat	Part F131936			122	128	7	10.1145/314https://www.functional.si.computer-science.org/2017/05/001						
Martin W., S	A survey of a	2017	IEEE Transac	43		9	7765038	817	847	10.1109/TSEhttps://www.app-store-analysis.com/2017/05/001						
Catalino G., I	A Set of Met	2017	Proceedings - 2017 IEEE/ACM 4th Intern				7972739	194	198	10.1109/MOhttps://www.in-this-work.com/2017/05/001						
Finkelstein A.	Investigating	2017	Information	87				119	139	10.1016/j.infhttps://www.computer-science.org/2017/05/001						
Quezada-Ló	CCSMIC Bas	2017	2016 IEEE 36th Central American and P				7942360			10.1109/CCPhttps://www.software-eff.com/2017/05/001						
Abuallish / A	study on th	2017	Information	86				1	19	10.1016/j.infhttps://www.background-base.com/2017/05/001						
Guimarães V	A reuse-base	2017	Proceedings - International Conference				7920978	712	719	10.1109/AINhttps://www.information-cost.com/2017/05/001						
Hovoes M., S	Functional cl	2017	Science of Co	135				88	104	10.1016/j.scihttps://www.computer-science.org/2017/05/001						
Oskan B., D	On the sever	2017	Proceedings - 26th International Works				7809590	45	51	10.1109/NWShttps://www.among-the.com/2017/05/001						
Symons C., A	Measurement	2017	Proceedings - 26th International Works				7809593	75	86	10.1109/NWShttps://www.information-cost.com/2017/05/001						
Tarhan A., O	A proposal o	2017	Proceedings - 26th International Works				7809610	195	200	10.1109/NWShttps://www.information-cost.com/2017/05/001						
Lavazza L., N	An empirical	2017	Proceedings - 26th International Works				7809592	65	74	10.1109/NWShttps://www.information-cost.com/2017/05/001						

FIGURE C.3. – Excel RSL : onglet SCO1

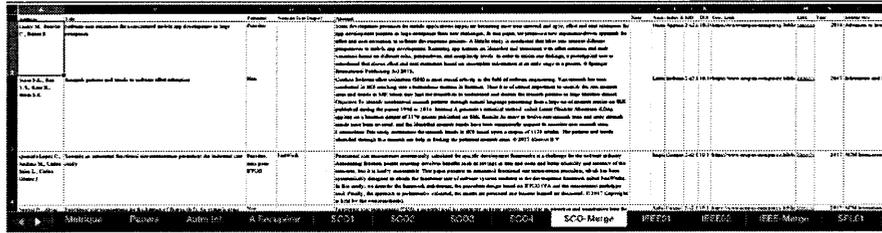


FIGURE C.4. – Excel RSL : onglet SCO-Merge

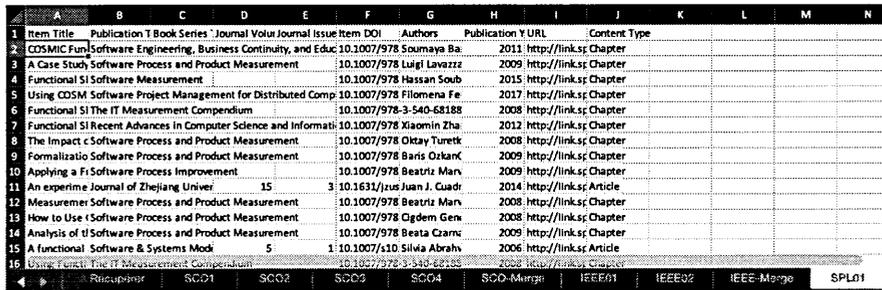


FIGURE C.5. – Excel RSL : onglet SPL01

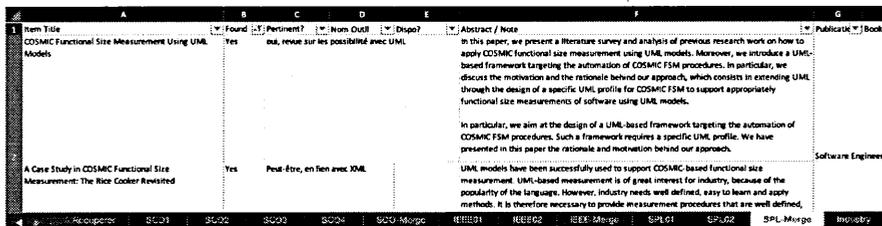


FIGURE C.6. – Excel RSL : onglet SPL-Merge

**ANNEXE D.**

**CARTE CONCEPTUELLE DES OUTILS SUPPORTANT LA  
MESURE**

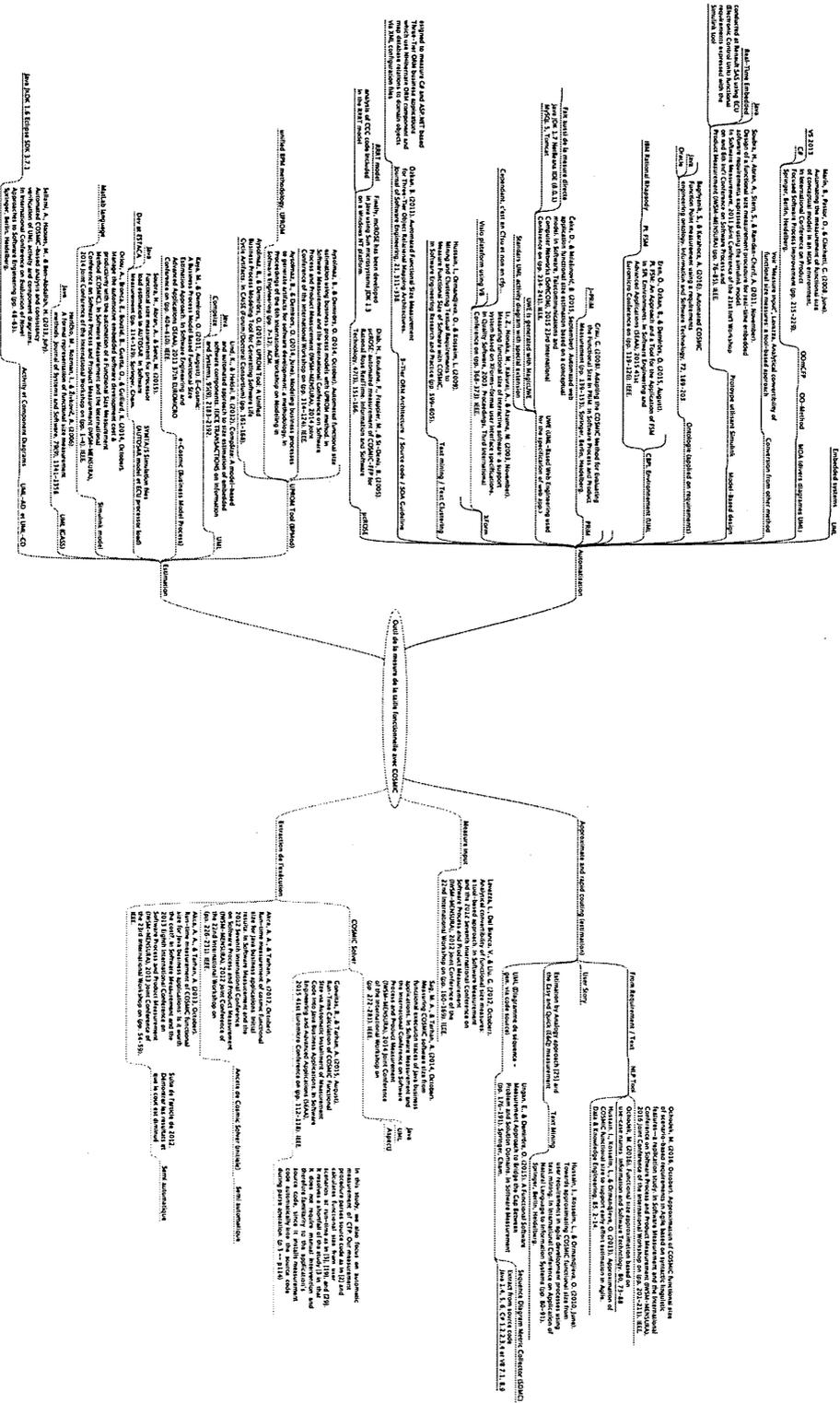


FIGURE D.1. – Outils supportant la mesure selon COSMIC : carte conceptuelle

## **ANNEXE E.**

### **CLASSIFICATION DES ARTICLES**

Table : Visualisation des regroupements de sujets des articles retenus



**ANNEXE F.**

***COSMIC-SIZING-TOOL BACKLOG***

number	title	body	tag
62	Recover password functionality		
58	SPIKE: Conformance to WCAG 2.0	Basic WCAG 2.0 should be applied for HTML structure.	
53	Bug: Behaviour when clicking on the top-left logo	Clicking on the top left logo should bring back the landing page (when not logged in) or the main page (when logged in).	bug
52	ISBSG: Generate ISBSG Data Submission Form		enhancement
51	SPIKE: DB: intégrer un DBMS/Integrate a DBMS	Integrate a known open-source DBMS (MySQL or PostgreSQL) along with data access mechanisms for other teams to use. Data access public classes and methods should be documented in such a way that any developer may use them.	
39	PROFIL: Recadrer photo/Cropping profile picture	FR: Lors de l'upload de la photo du profil, je veux être capable de la recadrer dans un carré de 2,5 cm x 2,5 cm. EN: When uploading a picture, I want to reframe that picture in a square of 2.5cm x 2.5 cm.  FR: La photo sélectionnée doit avoir une taille maximale de 1 MB (comme sur GitHub). Vérifier et afficher le message "Le fichier de photo est trop gros (>1Mo). SVP, sélectionnez un fichier plus petit." EN: The selected photo has a maximum size of 1 MB (as in GitHub). Verify and display message "The picture file is too large (>1MB). Please select a smaller picture pile."	
38	PROFIL: Page "Courriels"/"Emails"	FR: En tant qu'utilisateur, je veux pouvoir GÉRER MES ADRESSES DE COURRIEL afin d'en maintenir plusieurs et de les tenir à jour. EN: As a user, I want to MANAGE MY EMAIL ADDRESSES so I can keep several email addresses up-to-date.	

20	Chercher un mesureur	<p>En tant que membre du public, je veux chercher un mesureur dans ma région afin de lui offrir un mandat de mesure. "Trouver un mesureur"/Find a certified measurer" est une option dans le menu de gauche accessible au public non authentifié. On a les filtres de recherche dans le haut de la page:- Méthode (Drop down avec COSMIC, IFPUG, NESMA, FISMA, Mark II de la table MeasurementMethod ou "Toutes/All") et sa version optionnelle (de la table MeasurementMethodVersion) --&gt; ne pas demander la version si aucune méthode choisie.- Pays/Country (dropdown de Country, ou "Tous/All")- État/Province / State/Province (Drop down selon le pays ou "Tous/All", vide si le pays n'est pas spécifié --&gt; ne devrait être actif que si un pays a été spécifié: le libellé de la division de CountryDivisionType + Contenu de DivisionPerCountry)- Ville/City (Drop down selon le pays et la province mais seulement à partir des valeurs des mesureurs ayant ces valeurs, en ordre alphabétique du nom de ville) Bouton "Chercher"/"Search" --&gt; Cherche seulement ceux qui ont marqué "Disponible pour mandats de mesure". Les résultats s'affichent en-dessous dans un tableau avec les colonnes suivantes:- Nom, prénom- Courriel (cliquable "mailto:email"), à condition que le courriel soit "isVisible"- Liste des certifications selon les critères du filtre (Méthode from MeasurementMethod + version de certification from MeasurementMethodVersion, + date AAAA-MM concaténée from UserCertification) Si on ne trouve personne, alors afficher "Aucun mesureur trouvé correspondant à ces critères"/"No measurer found using these criteria". Afin d'éviter que l'information ne soit pas récupérée par un robot, il faudra implémenter un principe pour s'assurer que la demande d'information est justifié</p>	enhancement
19	LOG: Visualiser l'historique d'entrée/modification des mesures	<p>FR: En tant que Mesureur, je veux voir le log de l'historique des mesures d'un projet afin d'auditer ce qui a changé, par qui et quand.  EN: As a Measurer, I want to view the measurement history of a project so I can audit what was changed, by whom and when.</p>	Mesures

18	IMPORT/EXPORT: Excel	<p>En tant que Mesureur, je veux importer des mesure d'Excel afin d'accélérer l'entrée de mes mesures.En tant que Mesureur, je veux exporter des mesures vers Excel, afin de faciliter certaines analyses.Ça prend un onglet pour les données de projet:- Nom du projet- Stratégie de mesure (raison de la mesure)- Utilisateurs fonctionnels- Niveau de granularité- Niveau de décompositionUn onglet pour les mesures détaillées où on a, sur chaque ligne:- Type de changement (Ajout/Mod/Sup: obligatoire)- Système (nom court, obligatoire)- Couche (peut être vide ou carrément absente)- Référence (pourrait être vide)- Processus fonctionnel (obligatoire)- Groupe de données (obligatoire)- 1 colonne pour l'ensemble des mouvements (EXRW: obligatoire)- 4 colonnes pour les mouvements en PFC (en PFC, donc vide ou 0 si absent, 1 si présent)- Total par ligne- Commentaire/hypothèseHiérarchie d'options pour l'Import (cas d'affaires possibles):1- On importe un fichier Excel parfaitement bien formaté (pour l'onglet des mesures détaillées), les colonnes sont dans l'ordre attendu,2- Le fichier excel a toutes les colonnes au bon endroit mais certaines lignes sont vides. Les lignes complètement vides doivent être ignorées,3- Le fichier excel a toutes les bonnes colonnes mais certains champs obligatoires (changement, système, PF) sont vides: on émet l'hypothèse que leur valeur devrait être la même que la ligne précédente.4- Le fichier Excel a toutes les colonnes mais pas dans l'ordre attendu: donc on doit détecter les colonnes avec leur titre (peu importe la langue) et avoir un mécanisme pour associer [manuellement, semi-automatiquement, automatiquement] chaque colonne avec la donnée qu'elle contient (ex. colonne "GD" = Groupe de donnée, autant an anglais qu'en français).Possible que ça change dans le temps.Fonctionnement attendu:Option no.1: Il est de la responsabilité du mesureur de fournir un chiffrier adéquatement formaté.a) Le système demande un fichier (drag-n-drop ou navigateur de fichier avec sélection). b) L'utilisateur donne le fichier. c) Le système l'ouvre, détecte les onglets, les affiche dans une liste et demande dans quel onglet se trouvent les mesures détaillées. d) L'utilisateur sélectionne l'onglet parmi la liste d'onglets. e) Le système lit les données de mesure dans l'onglet spécifié et les importe dans sa structure de données dans l'ordre suivant: Software, Layer, FunctionalProcess (incluant le type de changement), DataGroup (incluant les mouvements de données, la référence et le commentaire). Si un processus fonctionnel est vide alors qu'il y a un groupe de données et des mouvements de données, alors le PF = le PF précédent.f) Le système affiche</p>	enhancement
----	----------------------	---	-------------

		<p>les données importées dans un tableau (pour que l'utilisateur puisse les voir et naviguer à travers sans rien pouvoir changer) et demande de sélectionner une organisation et un projet.g) L'utilisateur sélectionne une organisation et un projet de cette organisation (dropdown), puis il confirme l'importation des données.h) Le système affecte l'organisation et le projet aux données importées et il les persiste dans la BD.</p>	
--	--	---	--

17	MESURES: Entrer/Modifier des mesures	<p>En tant que Mesureur, je veux entrer et modifier des mesures de projet/application, [afin d'analyser ces mesures plus tard].</p> <p>La saisie des mesures est au coeur de cette application. Elle se doit d'être la plus efficiente possible en créant les occurrences d'objets au fur et à mesure qu'ils sont saisis, tout en permettant l'auto-complétion avec les données préalablement entrées.</p> <p>Le GUI doit être relativement compact.</p> <p>0) Déclencheur: À partir de la page "Dashboard", l'utilisateur clique sur un projet existant (lien avec le nom du projet) pour éditer les mesures OU l'utilisateur clique sur le lien "Entrer des mesures".</p> <p>1) L'utilisateur sélectionne une organisation ou entre un nouveau nom d'organisation.</p> <p>2) L'utilisateur créé ou choisit un projet.</p> <p>3) L'utilisateur créé ou choisit un logiciel:</p> <ul style="list-style-type: none"> <li>- Nom du logiciel</li> <li>- "Utiliser des couches/Use layers?", --&gt; Booléen</li> <li>- Type de changement, parmi "Nouveau/New", "Modifié/Modified" ou "Supprimé/Deleted".</li> </ul> <p>4) L'utilisateur créé ou choisit une couche (si utilisé):</p> <ul style="list-style-type: none"> <li>- Nom de la couche</li> </ul> <p>5) L'utilisateur crée ou modifie des processus fonctionnels:</p> <ul style="list-style-type: none"> <li>- Nom du processus fonctionnel (texte, 64 char)</li> <li>- Référence (à une section de la documentation utilisée pour mesurer ce PF, texte, 16 char)</li> <li>- Type de changement (si le logiciel est "New", --&gt; affecter "New" au PF et griser le champ, si le logiciel est "Deleted", affecter "Deleted" au PF et griser le champ, si le logiciel est "Modified" --&gt; saisir n'importe laquelle es 3 valeurs pour ce PF)</li> <li>- Cote de qualité/Quality Rating: valeur parmi "Inconnue/Unknown", "a", "b", "c", "d", "e". Valeur par défaut: "Inconnue/Unknown".</li> <li>- Commentaire (Texte, 64 char)</li> </ul> <p>6) Puis, par PF, l'utilisateur crée ou choisit un groupe de données avec ses mouvements de données:</p> <ul style="list-style-type: none"> <li>- Nom du groupe de données</li> <li>- Mouvements (string de 4 char pouvant contenir une seule occurrence max</li> </ul>	
----	--------------------------------------	---	--

de EXRW)

Autres champs optionnels:

- *Commentaire/hypothèse*

Voir la page Wiki "User Interface Principles" sur le site GitHub du repository.

16	RAPPORTS: Effort de mesure	<p>En tant que mesureur, je veux analyser l'effort de mesure de mes projets afin de pouvoir m'améliorer. Je veux voir la liste pour chacun de mes projets:- la taille totale (en PFC)- l'effort (en heures:minutes)- la productivité de mesure (en #PFC/heure)- un indicateur selon la valeur de productivité (voir les settings par organisations) . En vert si &gt;= valeur cible . En jaune si 0-20% en-dessous de la valeur cible . En rouge si très en dessous de la valeur cible. 2e niveau de détail:- Effort par contributeur (mêmes données), si plus d'un contributeur.</p>	Effort
15	EFFORT: Mesurer l'effort de saisie	<p>Chrono: En tant que Mesureur, je veux mesurer l'effort de mesure (le mien et celui de mes collaborateurs) afin de surveiller ma performance de mesure.</p> <p>L'effort est calculé en minutes à partir du moment où un projet de mesure est créé. L'effort se cumule tant et aussi longtemps que je suis actif dans ce projet. Le compteur d'effort s'arrête si je change de projet ou que je fais "Pause".</p> <p>Des boutons Start/Pause (type toggle) sont présents en tout temps dans le haut de l'écran afin que je puisse arrêter le compteur. Si j'ai fait "Pause", je ne peux plus entrer de nouvelles mesures dans ce projet tant que je ne fais pas "Start".</p>	Effort

14	RAPPORTS: Visualiser les mesures	<p>En tant que Mesureur, je veux visualiser mes mesures (sommaires/détaillées) afin d'avoir un aperçu rapide d'où j'en suis dans ma mesure.</p> <p>Sommaire par (imbriqués dans cet ordre):</p> <ul style="list-style-type: none"> <li>- Organisation</li> <li>- Projet/Application</li> <li>- Morceau de logiciel (si plus d'un)</li> <li>- Couche (si plus d'une)</li> </ul> <p>1er niveau de détail: Informations sommaires requises (par ligne):</p> <ul style="list-style-type: none"> <li>- Ajouts: Nombre de PF, #E, #X, #R, #W, Total</li> <li>- Modifications: Idem</li> <li>- Suppression: Idem</li> </ul> <p>2e niveau de détail: Par PF:</p> <ul style="list-style-type: none"> <li>- Afficher une ligne par PF selon ces 3 catégories (Ajouts, Modifications, Suppression) seulement si elles ne sont pas vides.</li> </ul> <p>3e niveau de détail: Introduire les Groupes de données entre le nom des PF et les mouvements.</p>	Mesures
----	----------------------------------	---	---------

13	PATRON: Gérer les patrons génériques/Manage generic patterns	En tant qu'Admin, je veux gérer les patrons génériques de mesure dans l'outil afin que les utilisateurs puissent les utiliser. Cette fonctionnalité devrait être visible seulement aux utilisateurs ayant les droits Admin sur l'application COSMIC-sizing-tool. Même structure que les patterns de l'utilisateur mais spécialement identifiés à COSMIC. Ils ne sont pas modifiables par les utilisateurs qui peuvent seulement les copier pour les modifier. Bouton vert en bas: Enregistrer le patron/Save pattern Bouton jaune: Annuler/Cancel	Patterns
----	--	---	----------

12	PATRON: Gérer mes patrons de mesurage/Manage my measurement patterns	<p>En tant que Mesureur, je veux gérer mes patrons de mesurage afin d'accélérer la saisie de la mesure plus tard.</p> <p>Les patrons servent pour générer de la mesure détaillée. Une fois utilisés, ils peuvent être modifiés pour un prochain usage sans affecter les mesures déjà faites avec ce patron (Release-1).</p> <p>Ça prend 2 fonctionnalités:</p> <ul style="list-style-type: none"> <li>- "Manage my patterns"/"Gérer mes patrons" qui affiche la liste des patrons (Pattern_Name, Pattern_Desc from table Pattern [header], #Func.Proc from table Pattern_FP., Size from table Pattern_DGDM). Cette fonctionnalité contient un bouton/lien "New pattern"/"Nouveau patron" qui mène vers la fonctionnalité "Créer/Modifier un patron". Chaque Nom et description de patron existant sont cliquables afin de modifier ce patron avec "Créer/Modifier un patron". Pour supprimer un ou plusieurs patrons, l'utilisateur sélectionne une ou plusieurs lignes de patrons et le bouton "Delete"/"Supprimer" s'active en rouge et il le clique (sinon, il est grisé) --&gt; Message de confirmation --&gt; Suppression des données de ce.s patron.s dans les 3 tables (Pattern, Pattern_FP, Pattern_DGDM). La sélection/Suppression/Modification sont à tester pour une ergonomie fluide. Si un patron a isAValidPattern=False alors le texte de la ligne de ce patron est affiché en rouge.</li> <li>- "Créer/Modifier un patron"/"Create/Modify my pattern". <ul style="list-style-type: none"> <li>Mode création: L'utilisateur entre un nom de patron (15 char) et une description du patron (char 64). Le système affiche son user_ID vis-à-vis (non modifiable). La taille est à zéro et le nombre de processus fonctionnels aussi. L'utilisateur entre un nom d'un PF &lt;Tab&gt;. Le système incrémente le #PF. L'utilisateur entre un nom de DG (Data group) &lt;tab&gt;, une string de DM (Data Movements) --&gt; Seulement EXRW, toute autre lettre sera ignorée, &lt;tab&gt;. Le système incrémente la taille du patron. Le système positionne le curseur sur une nouvelle ligne de DG. Si l'utilisateur fait &lt;tab&gt; sur un DG vide, le système positionne son curseur sur le champ FP de cette ligne pour entrer un nouveau FP. Et ainsi de suite.</li> <li>Mode Modification: Le système affiche l'entête du patron (avec #PF et taille en CFP), les FP (en ordre alphabet les DG/DM. Chaque ligne est "sélectable" (elle devient bleue), on peut en choisir plusieurs (Shift ou Ctrl). Dès qu'au</li> </ul> </li> </ul>	Patterns
----	--	---	----------

moins une ligne est sélectionnée, le bouton "Delete"/"Supprimer" est activé en rouge (sinon, il est grisé). Pour ajouter un PF entre 2 PF, positionner le curseur sur un PF, faire <Enter> et le système insère une ligne en-dessous. Pour ajouter un DG entre 2 DG, on positionne le curseur sur un DG et on fait <Enter>. Sinon, on ajoute à la fin, comme en mode création.

Les 2 modes: dès qu'une entrée est modifiée/ajoutée, le bouton "Save"/"Enregistrer" s'active en vert (sinon il est grisé).

Règles de validité d'un patron:

- 1) Un pattern doit avoir au moins un PF.
- 2) Un PF doit avoir au moins un DG.
- 3) Un DG doit avoir au moins un DM.
- 4) Un PF doit contenir au moins une Entrée ET (une sortie-X OU une écriture-W).

À l'enregistrement, détecter la règle et afficher un message approprié.

Les patrons de mesurage doivent être liés à l'utilisateur.

Un patron devrait avoir les 3 tables suivantes:

Pattern (entête)

- Pattern\_Name (char 15)
- User\_ID (char??) --> lié au user\_ID du "profile"
- Pattern\_Desc (char 64)
- isAValidPattern (Booléen)

Pattern\_FP (1er niveau de détail d'un pattern, relation 1 à n avec l'entête de patron)

- Pattern\_ID char15
- Pattern\_FP\_ID char64

Pattern\_DGDM

- Groupe de données (char 64) 1 variable par groupe entre <>
- Mouvements de données (EXRW) --> 4xint ou 4 booléens (on veut les additionner).

Les données devraient être saisies dans un tableau assez compact à l'écran,

resizable, sortable (souhaitable).

Fonctionnalité souhaitable: Champs en haut de la page: "Copier d'un patron"/"Copy from pattern":  
ComboBox avec la liste des patrons fournis par le COSMIC-sizing-tool plus tous ceux créés par l'utilisateur. Ça copie le détail mais pas l'entête que l'utilisateur doit saisir (champs obligatoires).

Bouton vert: Enregistrer le patron/Save this pattern

Bouton jaune: Annuler/Cancel

11	METHODE: Visualiser le glossaire COSMIC/View COSMIC Glossary	<p>Libellé de l'onglet: FR"Glossaire COSMIC"/EN"COSMIC Glossary"Disponible au grand public et aux utilisateurs authentifiés dans le menu de gauche.Cette page affiche les termes du glossaire sous forme de liste (selon la langue active de la session):- Terme- DéfinitionTous les termes sont pigés directement des manuels de mesurage de COSMIC, en anglais et en français pour le Release-1.Petit espace en entête pour filtrer avec 3 comboBox "Contient"/"Contains" avec radio boutons "ET", "OU" afin de filtrer. Bouton Vert "Filtrer"/"Filter", Bouton Gris "Effacer le filtre"/"Clear filter". Le filtre est minimisable pour occuper moins d'espace à l'écran. Par défaut: "Collapsed". Icône de filtre pour l'activer: en rouge si filtre non vide, en bleu/gris si filtre vide.Quand les éléments du glossaire s'affichent, le système affiche une ligne avec les lettres de l'alphabet pour naviguer à travers les termes. Les lettres utilisées sont cliquables en bleu et soulignées. Les lettres non utilisées sont grisées et non cliquables. Les termes s'affichent en ordre alphabétique de la langue courante. 2 colonnes: "Terme" /"Term", "Définition"/"Definition".La page est resizable: les colonnes Terme et Définition doivent s'ajuster jusqu'à un certain point, ça doit rester utilisable.</p>	enhancement
10	MESURE: Gérer des application/Manage Pieces of Software	<p>Un projet de mesure comprend une ou plusieurs applications (au moins une). Pour les informations détaillées, voir le formulaire ISBSG.</p> <p>Une application peut être divisée en plusieurs couches qu'on doit pouvoir mesurer séparément (ou pas).</p>	Measures

9	<p>MESURE: Gérer mes projets [de mesure]/Manage my [measurement] projects</p>	<p>Les informations saisies ici vont permettre de générer un rapport de données pour ISBSG (base de données de benchmarking). Les informations à saisir doivent être les mêmes que sur le formulaire disponible à <a href="http://www.isbsg.org">www.isbsg.org</a> (un exemple priorisé de ce questionnaire a été placé sous <a href="https://github.com/cosmic-sizing-tool/cosmic-sizing-tool/blob/master/server/public/images/COSMIC-ISBSG%20Benchmarking%202014%20-%20Concise%20DCQuestionnaire-PrioritizedForTool.docx">https://github.com/cosmic-sizing-tool/cosmic-sizing-tool/blob/master/server/public/images/COSMIC-ISBSG%20Benchmarking%202014%20-%20Concise%20DCQuestionnaire-PrioritizedForTool.docx</a> ). L'information est divisée selon les sections suivantes: A. Submitter Information / FR-"Information du soumissionnaire" B. Process /FR-"Processus" C. Technology /FR-"Technologie" D. People and Work Effort /FR-"Personnel impliqué et Effort de travail" E. Product /FR-"Produit logiciel" F. COSMIC Project Functional Size /FR-"Taille Fonctionnelle COSMIC du Logiciel" G. Project Completion /FR-"Clôture du projet" Quand l'utilisateur démarre un nouveau mesurage (voir fonctionnalité "Create/Edit measures of a project", seules quelques informations essentielles vont être créées:- Nom du projet: textBox- Type de projet: dropdown (Enhancement, New development, Redevelopment) Le reste des champs requis doit être saisi ici par la fonctionnalité "Gérer mes projets". Cette fonctionnalité aura besoin de plusieurs domaines de valeurs (ex. Country [1][64], SoftwareProjectType [5], DomainType [6] and used for [87], ReusableType [7], ProcessImprovementType [12], TechnologyType [58], EnvironmentType [60], PrimaryImplementationPlatformType [63], EmbeddedSoftwareType [63], PrimaryEndUserDomainType [71], TimesheetType [78], EffortQualityType [85], AppDomainType [87], MeasurementApproachType [94], FunctionalUserType [95], MeasurementPhaseType [110], MeasurementAccuracyType [112], FunctionalSizeSourceType [114], DocumentationQualityType [115], SizingTechnologyType [116], MeasurerExperienceType [119], ), en plus de certaines données d'affaires (Organization [1], UserProfile [1], Project [2] and all other required fields, SizingMethod [92], SizingMethodVersion [93]). Les chiffres entre crochets ([]) représentent les sections du questionnaire ISBSG où ces informations seront requises.</p>	Measures
---	---	---	----------

8	<p>GUI: Changer la langue d'affichage/Change the displayed language</p>	<p>FR: Nonobstant la préférence de langage de l'utilisateur, dans le haut de l'écran, on doit pouvoir sélectionner la langue d'affichage (drop down en ordre alphabétique de la langue).</p> <p>EN: Whatever is the preferred user language (as defined in his profile), any user may select another language in the top of screen (dropdown, in alphabetical order of language).</p> <p>FR: Pour le Release-1, on commence par le français et l'anglais. Les manuels de mesurage sont disponible dans les 2 langues et contiennent un glossaire à la fin (annexe F). L'anglais doit être la langue par défaut (et celle de référence) quand on arrive sur la page d'accueil. On doit pouvoir changer la langue même si on ne s'est pas authentifié.</p> <p>EN: For Release-1, we begin by French and English. Measurement manuals are available in both languages with a glossary in annex F. English must be the default language when getting to the landing page. Any user may change the language, even if not logged in.</p> <p>FR: Quand on change la langue courante, les libellés se changent sans reloader la page.</p> <p>EN: When changing the current language, text labels are displayed in the selected language without reloading the page.</p>	enhancement
---	---	---	-------------

7	GUI: Page d'accueil/Landing Page	<p>La page d'accueil (Landing Page) doit être visible à tout internaute qui désire en savoir davantage sur l'outil et la méthode. / The Landing Page must be visible to anyone who wishes to know more about the COSMIC tool or method (public page).Bande du haut de la fenêtre/Top of screen (ribbon):- Logo COSMIC/ COSMIC LOGO- Nom de l'outil/Tool name: "COSMIC-sizing-tool"- DropDown pour changer la langue: En "English" par défaut, pour le moment, on rend disponible "English" et "Français", Voir les autres règles d'affaires dans la user story traitant du changement de langue / Dropdown to change the language: "English" is default, for now, both "English" and "Français" should be available. See other user story for rules when changing language.- Bouton "S'inscrire"/"Register" button (or link)- Bouton "Accéder"/"Log in" button (or linkMenu à gauche si "Mouse over" sur le logo de COSMIC/Mouse over menu on the left:À propos du COSMIC-sizing-tool / About the COSMIC-sizing-tool (text to be done)À propos de la méthode COSMIC / About the COSMIC method (text to be done)Glossaire COSMIC / COSMIC GlossaryChercher un mesureur / Find a measurer</p>	enhancement
---	----------------------------------	---	-------------

6	MESURE: Gérer mes organisations/Manage my organizations	<p>Onglet du navigateur: "Créer/Modifier/Supprimer une organisation"/"Create/Modify/Delete an Organization"</p> <p>Priorité no1: Create/Modify, Priorité no.2: Delete.</p> <p>Create/Modify:</p> <p>0) Cette fonctionnalité commence quand l'utilisateur a cliqué sur une organisation existante du dashboard OU il a choisi "Créer une organisation/Create an organization" dans ce même dashboard.</p> <p>1) Le système affiche la section "Profil d'organisation"/"Organization profile" avec les champs suivants:</p> <ul style="list-style-type: none"> <li>- Photo/Logo (comme pour le compte utilisateur) --&gt; Optionnel.</li> <li>- Code de l'organisation/Organization code (Nom court donné par le User) --&gt; éviter les doublons "Cette organisation existe déjà/This organization already exists".</li> <li>- Nom de l'organisation/Organization's Name</li> <li>- Description</li> <li>- URL</li> <li>- Contact/Contact info: Nom, téléphone, courriel,</li> <li>- Emplacement/Location: adresse-1, adresse-2, Ville, Pays (dropdown de Country), État/Province/Région (libellé de CountryDivisionType + Contenu de DivisionPerCountry), Code postal/Zip code.</li> </ul> <p>2) L'utilisateur saisit/modifie les champs et il appuie sur le bouton vert: "Enregistrer l'organisation"/"Save organization".</p> <p>3) Le système persiste l'organisation.</p> <p>Delete:</p> <p>4) Le système affiche les informations de l'organisation sélectionnée,</p> <p>5) L'utilisateur appuie sur le bouton ROUGE "Supprimer/Delete",</p> <p>6) Le système vérifie qu'il n'y a pas de projet dans cette organisation --&gt; "Impossible de détruire cette organisation car il reste des projets actifs. Vous devez d'abord détruire les projets sous cette organisation avant de pouvoir la supprimer./Not able to delete this organization due to existing projects. You need to delete the projects first, then delete the organization."</p> <p>7) Le système supprime l'organisation.</p>	Measures
---	---	---	----------

5	PROFIL: Mon compte/Account settings	<p>En tant que Mesureur, je veux gérer mon compte afin qu'il soit à jour./As a Measurer, I want to manage my account so that I can keep it up-to-date.</p> <p>Section: Changer de mot de passe/Change password- Mot de passe actuel/Current password- Nouveau mot de passe/New password- Confirmer le nouveau mot de passe/Confirm new password Si OK: message "Mot de passe changé avec succès"/"Successful password change". Si pas-OK: "La confirmation du mot de passe ne correspond pas"/"Invalid password confirmation". Dans les 2 cas, on reste sur cette page. L'utilisateur choisira d'aller où il voudra par la suite./In both cases, stay on that page. The user will choose to go wherever he wants afterwards.</p> <p>Section: Modifier mon identifiant ou courriel/Change ID or emailAffiche le ID et le courriel actuel de l'utilisateur./Display current user ID and email.L'utilisateur peut modifier son adresse courriel./The user can modify his email address.</p> <p>Bouton VERT/GREEN button: Mettre à jour/Update.</p> <p>Section Supprimer mon compte/Delete account (l'entête de section est en ROUGE/Section heading is in RED)- Mise en garde (si owner d'organisations dans l'outil)/Warning (if owner of at least one organization): "Votre compte est présentement administrateur principal de ces organisations:"/"Your account is currently owner of these organizations:" + liste des noms d'organisations où on est administrateur principal/+ list of organizations where this user ID is the main owner.- "Vous devez transférer d'administrateur principal ou détruire ces organisations d'abord"/"You must first transfer the ownership of these organizations to another account or delete these organizations."- Si aucune organisation, activer le bouton ROUGE "Supprimer ce compte"/If not owner of an organization, activate the RED button "Delete this account", suivi d'un message de confirmation/followed by a confirmation message.</p>	Security
---	-------------------------------------	---	----------

4	PROFIL: Page "Profil public"/"Public Profile"	<p>La page de profil public doit permettre à un utilisateur Mesureur de s'identifier et de rendre disponible au public un certain nombre d'informations le concernant. Ces informations pourraient être utilisées par un membre du public pour se chercher un consultant en mesure COSMIC. La page sera divisée en plusieurs sections, comme le profil public de GitHub.</p> <p>Section FR"Profil public"/EN"Public profile"  Sur l'onglet du navigateur:  FR"Votre profil"/EN&gt;Your Profile"  - Photo du profil/Profile picture: espace pour une photo (à peu près 1 pouce carré (2,5 cm) avec un bouton (gris très pâle) à côté: "Télécharger une photo"/"Upload new picture"  - Nom/Name: textBox  - Courriel/Email: Drop down avec la liste des courriels et l'option "Masquer mon courriel"/"Don't show my email address"  - URL: textBox  - Compagnie/Company: ComboBox (de liste des compagnies)  - Emplacement (Ville, Province/État, Pays)/Location (City, Province/State, Country): comboBox  - Langue d'utilisation par défaut/Default language: DropDown des langues avec le drapeau correspondant  - Bouton vert: Mettre à jour mon profil/Update profile.--&gt; Rester sur la page et afficher un message si les informations ont été sauvegardées correctement.</p> <p>Section FR"Profil de mesureur"/"Measurer Profile"  - Disponible pour projets de mesure COSMIC/Available for measurement projects: Check box  - Certifications (liste avec les éléments suivants, on commence avec une ligne seulement et il y a un lien pour Ajouter une certification/Add a certification):  - Méthode de mesure/Measurement Method: Drop down avec les choix --&gt; COSMIC, Mark II, IFPUG, NESMA, FiSMA  - Version/Version: pour COSMIC (2.0, 2.1, 2.2, 3.0, 4.0), pour IFPUG (4.1, 4.2, Inconnue/Unknown), pour NESMA (2.0, 2.1), pour Mark II (1.2, 1.3, 1.3.1, Inconnue/Unknown), pour FiSMA (Inconnue/Unknown) --&gt; Message d'erreur s'il y a un doublon Méthode/Version: FR"Cette certification existe déjà"/EN"This certification already exists".</p>	Security
---	---	---	----------

		<ul style="list-style-type: none"><li>- Date d'obtention (Mois/Année)/Date obtained (Month/Year): Deux drop down, un pour le mois (Janvier à Décembre), un autre pour l'année (en ordre décroissant à partir de [l'année courante, ex. 2016] jusqu'à 1985).</li><li>- Bouton vert "Mettre à jour mon profil de mesureur"/"Update my measurer profile" --&gt; Rester sur la page et afficher un message si les informations ont été sauvegardées correctement.</li></ul>	
--	--	---	--

3	SECURITE: Forgot ID/Password	<p>En tant qu'utilisateur, je veux recevoir mon mot de passe par courriel quand je l'ai oublié afin de pouvoir entrer dans l'outil. L'utilisateur clique sur le lien "J'ai oublié mon mot de passe"/"I forgot my password". a) Le système demande à l'utilisateur d'entrer son courriel (email), de saisir le captcha et de cocher la case "Je ne suis pas un robot"/"I am not a robot", avec un bouton "Envoyer"/"Send". b) L'utilisateur entre un courriel valide, saisit le captcha et coche la case anti-robot, puis clique sur Send. c) Le système envoie le mot de passe correspondant à l'utilisateur ayant cette adresse courriel. Si ce courriel n'existe pas pour aucun utilisateur, le système affiche FR "Cet utilisateur n'existe pas"/EN "This user does not exist". À l'étape b), tant que le courriel n'est pas valide, le système indique "! Ce courriel n'est pas valide"/"! This email is invalid".</p>	Security
2	SECURITE: Register/Sign up	<p>S'enregistrer dans l'outil (se créer un compte = Register)          Devrait être fait de la même façon que le "Sign up" dans GitHub, avec les informations suivantes:</p> <ul style="list-style-type: none"> <li>- ID ou email</li> <li>- password</li> <li>- Langue préférée/Preferred language: Liste déroulante en ordre alphabétique</li> </ul> <p>--&gt; Par défaut, ce sera la langue de l'application au moment d'entrer dans cette fonctionnalité.</p>	Security

1	SECURITE: Login/Logout	<p>S'authentifier (Login)/Se déconnecter (Logout)</p> <p>Login:</p> <p>Comme dans GitHub, ID et mot de passe.</p> <p>Si l'un ou l'autre sont invalides, afficher le message FR"Identifiant ou mot de passe invalide"/EN"Invalid ID or password" en ré-affichant la page.</p> <p>Après le 5e essai infructueux --&gt; Page "ID ou mot de passe oublié".</p> <p>Si succès --&gt; Page d'accueil</p> <p>Lien FR"J'ai oublié mon ID ou mot de passe"/EN"I forgot my ID or password"</p> <p>Logout:</p> <p>Comme dans GitHub.</p>	Security
---	------------------------	--	----------



## ANNEXE G.

### ***DEVELOPER'S GUIDE FOR THE COSMIC-SIZING-TOOL***

Remarque : Le contenu de cette annexe est une copie du guide du développeur que j'ai créé dans les pages wiki sous Github. Il vise à s'assurer qu'un nouveau développeur puisse être rapidement fonctionnel pour poursuivre avec succès le développement de l'application.

*COSMIC-Sizing-Tool* is a Web App that supports functional size measurement using the COSMIC method with *FSM patterns* embedded. The project is in its Alpha version.

The following is a minimal guide to help building, using and modifying the prototype. It is written for a Mac OS. Basic knowledge with command line and web project dependencies are required to understand the following guide.

#### **G.1. Get sources**

GitHub account<sup>1</sup> and Git client<sup>2</sup> are required.

---

```
git clone https://github.com/jimmycloutier/cosmicsizingtool-dev.git
```

---

#### **G.2. Folder structure**

The git repos contains two folders corresponding to two projects required to run *COSMIC-Sizing-tool*.

- 
1. <https://github.com>
  2. <https://git-scm.com/downloads>

### G.2.1. backend folder

This folder contains the *back end* project sources. It is a REST API using Python and the library Flask.

The subfolders *dataMovement*, *functionalProcess*, *organization*, *pattern* and *project* are the modules of the project.

These subfolders contain the following files to create a module :

- *init.py* : To Inform Python that the directory is containing packages, set variables and import classes, function, etc. For more detail, see Python documentation.
- *controllers.py* : Define the routes for the *REST API* (entry point) and the calls to the objects implementing the business logic.
- *[module name]\_utils.py* : Generic functions used by module objects
- other files : Class(es) implementing business logic related to the module purpose

The subfolder *models* contains the data class to communicate with the database and work with the data. They are grouped by logical entity name and all inherited from *Base* class (*basemodel.py*).

The subfolder *utils* is a module to group generic functions that can be used in the whole project.

The root of the *backend* folder contains few files :

- *config.py* : Tool configuration
- *cosmicsizingtool.py* : main script to execute the tool
- *cst.db* : SQLite database

### G.2.2. Frontend folder

This folder contains the *front end* project sources. It is a HTML 5 GUI using React/Redux javascript library and jQWidgets javascript components. To understand the folder structure, knowledge with *React* and *Redux* project is required.

Main sources are in *src/client/app*.

Subfolders :

- *actions* : Sources for the *actions* (Redux).

- components : Sources for the *components* (React). Components to display.  
NOTE : *fpdmGrid.js* is the grid used by the user to obtain the functional size measurement. It must be refactored to follow React/Redux best practices (architecture). Few documents are available to use jqWidgets components with React/Redux. Logic was implemented as stated in jqWidgets example.
- constante : Note used. Contants are in *constants.js* file.
- containers : Sources for the *containers* (React). React *main* structure.
- reducers : Sources for the *reducers* (Redux). Manage object states.

Files :

- configureStore.js : Config the *store* (Redux)
- constants.js : Constants to be used for component *state* management
- index.js : main React container
- reducer.js : initialize reducers (Redux)

## G.3. How to build

### G.3.1. Back end

**TIP :** Set up a virtual environment

Install :

---

```
pip install virtualenv
```

---

Create a virtual environment : From project folder,

---

```
virtualenv [virtual environment name]
```

---

Activate virtual environment : From project folder,

---

```
source bin/activate
```

---

Desactivate virtual environment :

---

`deactivate`

---

Python version 3.6.1 is used for the *back end*.

Verify Python version :

---

`python --version`

---

or

---

`python3 --version`

---

*Pip* is used to install packages (Note : if multiple *pip* versions are installed, use *pip3* instead of *pip* in the command line below).

See online documentation to install the required *Python* and *pip* version.

### Dependencies

The following dependencies are required and can be installed by using *pip*.

Example :

---

`pip install Flask`

---

The list was generated by using *pip freeze*. See online documentation for the details on how to install dependency with *pip* if required.

Dependency Name	Version
blinker	1.4
certifi	2017.7.27.1
chardet	3.0.4
click	6.7
Flask	0.12.2
Flask-Autodoc	0.1.2
Flask-Cors	3.0.3
Flask-Dance	0.11.1

Dependency Name	Version
Flask-Login	0.4.0
Flask-SQLAlchemy	2.2
idna	2.5
itsdangerous	0.24
Jinja2	2.9.6
lazy	1.3
MarkupSafe	1.0
oauthlib	2.0.2
requests	2.18.3
requests-oauthlib	0.8.0
six	1.11.0
SQLAlchemy	1.1.13
SQLAlchemy-Utills	0.32.14
urllib3	1.22
URLObject	2.4.3
Werkzeug	0.12.2

## Build

From project path,

---

```
python cosmic sizingtool.py
```

---

or

---

```
python3 cosmic sizingtool.py
```

---

```
[(cosmic sizingtool) JimmyC:backend jimmy$ python3 cosmic sizingtool.py ]
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 241-811-760
```

FIGURE G.1. – *Back end* running in command line

By default, the REST API will run on `http://127.0.0.1:5000/`. The address can be used to test. Here is the result :

---

```
{  
  "message": "test project API"  
}
```

---

See *controllers.py* contained in each module to see available functions and how to call them. (TODO : Implement <https://github.com/rochacbruno/flagger>)

### G.3.2. *Front end*

To use the project, *npm* is required.

To install all the dependencies, run from the project root : `npm install`

#### Build

From project root :

---

```
npm run dev
```

---

Then, in another console

---

```
npm run start
```

---

By default, project uses this address :

---

```
http://localhost:8080/
```

---

NOTE : *Back end* project needs to run simultaneously to use the *front end* project.

## G.4. Architecture

### G.4.1. Overview

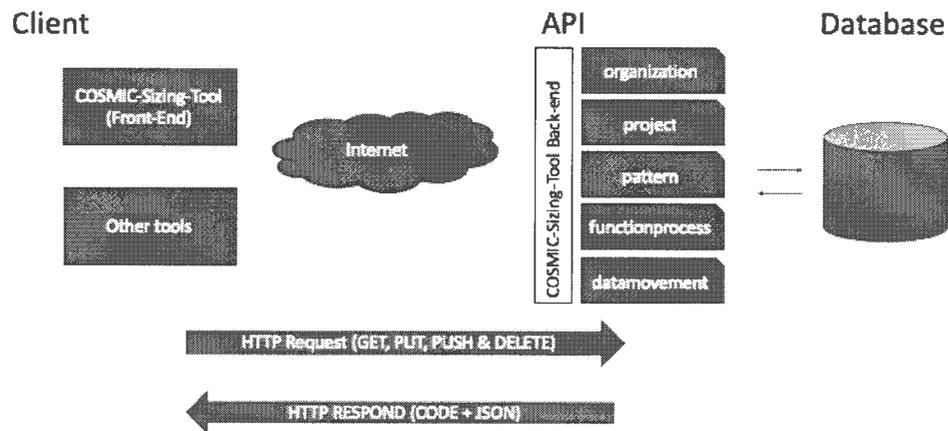


FIGURE G.2. – Architecture overview

### G.4.2. Back end

Go to <http://localhost:5000/apidocs> to get API documentation.

### G.4.3. Front end

The React *component* architecture and Redux *state container* architecture are described in *React + Redux : Architecture Overview*

## G.5. Coding standards

### G.5.1. Back end

Usually follow *PEP 8 - Style Guide for Python Code*<sup>3</sup>.

3. <https://www.python.org/dev/peps/pep-0008/>

### **G.5.2. Front end**

Usually follow *camelCase*<sup>4</sup>.

### **G.6. Dev. Environnement**

Both projects were developed using *IntelliJ IDEA* IDE<sup>5</sup>. However, a basic text editor can be used.

---

4. [https://en.wikipedia.org/wiki/Camel\\_case](https://en.wikipedia.org/wiki/Camel_case)

5. <https://www.jetbrains.com/idea/>

## BIBLIOGRAPHIE

Abran, A., Londeix, B., O'Neill, M., Santillo, L., Vogelezang, F., Desharnais, J., ... others. (2015). The COSMIC Functional Size Measurement Method. *Measurement Manual, Version, 4*(1).

Akca, A. A. et Tarhan, A. (2012, octobre). Run-time Measurement of COSMIC Functional Size for Java Business Applications : Initial Results. Dans *2012 Joint Conference of the 22nd International Workshop on Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement* (p. 226-231). doi : 10.1109/IWSM-MENSURA.2012.40

Akca, A. A. et Tarhan, A. (2013, octobre). Run-Time Measurement of COSMIC Functional Size for Java Business Applications : Is It Worth the Cost? Dans *2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement* (p. 54-59). doi : 10.1109/IWSM-Mensura.2013.19

Albrecht, A. J. (1979). Measuring application development productivity. Dans *Proc. of the Joint SHARE/GUIDE/IBM Application Development Symposium* (p. 83-92).

Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I. et Angel, S. (1977). A pattern language : Towns, buildings, construction (center for environmental structure). Oxford University Press Oxford.

Anderson, D. J. (2010). *Kanban : successful evolutionary change for your technology business.* (s. l.) : Blue Hole Press.

Aysolmaz, B. et Demirors, O. (2014). Modeling business processes to generate artifacts for software development : A methodology. *6th International Workshop on Modeling in Software Engineering, MiSE 2014 - Proceedings.* doi : 10.1145/2593770.2593775

Aysolmaz, B. et Demirörs, O. (2014). Automated Functional Size Estimation Using Business Process Models with UPROM Method. Dans *Proceedings of the 2014 Joint*

*Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement* (p. 114-124). Washington, DC, USA : IEEE Computer Society. doi : 10.1109/IWSM.Mensura.2014.10

Aysolmaz, B. et Demirörs, O. (2014). UPRoM Tool : A Unified Business Process Modeling Tool for Generating Software Life Cycle Artifacts. Dans *CAiSE (Forum/Doctoral Consortium)* (p. 161-168).

Bagriyanik, S. et Karahoca, A. (2016 avril). Automated COSMIC Function Point Measurement Using a Requirements Engineering Ontology. *Inf. Softw. Technol.*, 72(C), 189-203. doi : 10.1016/j.infsof.2015.12.011

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... others. (2001). Manifesto for agile software development.

Ceke, D. et Milasinovic, B. (2015). Automated web application functional size estimation based on a conceptual model. *2015 23rd International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2015*. doi : 10.1109/SOFTCOM.2015.7314074

Diab, H., Koukane, F., Frappier, M. et St-Denis, R. (2005). cROSE : automated measurement of COSMIC-FFP for Rational Rose RealTime. *Information and Software Technology*, 47(3), 151-166. doi : <https://doi.org/10.1016/j.infsof.2004.06.007>

Eren, Ö., Özkan, B. et Demirörs, O. (2015, août). PL FSM : An Approach and a Tool for the Application of FSM in SPL Environments. Dans *2015 41st Euromicro Conference on Software Engineering and Advanced Applications* (p. 119-126). doi : 10.1109/SEAA.2015.58

Gamma, E., Helm, R., Johnson, R. et Vlissides, J. (1993). Design patterns : Abstraction and reuse of object-oriented design. Dans *European Conference on Object-Oriented Programming* (p. 406-431). Springer.

Gonultas, R. et Tarhan, A. (2015). Run-Time Calculation of COSMIC Functional Size via Automatic Installment of Measurement Code into Java Business Applications. *Proceedings - 41st Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2015*. doi : 10.1109/SEAA.2015.30

Grau, G. (2008). Adapting the COSMIC Method for Evaluating the Functional Size in PRiM. Dans J. J. Cuadrado-Gallego, R. Braungarten, R. R. Dumke, et A. Abran (Dir.)

(dir.), *Software Process and Product Measurement* (p. 139-153). Berlin, Heidelberg : Springer Berlin Heidelberg.

Heričko, M., Rozman, I. et Živkovič, A. (2006 septembre). A Formal Representation of Functional Size Measurement Methods. *J. Syst. Softw.*, 79(9), 1341-1358. doi : 10.1016/j.jss.2005.11.568

Hussain, I., Kosseim, L. et Ormandjieva, O. (2010). Towards approximating COSMIC functional size from user requirements in agile development processes using text mining. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6177 LNCS, 80-91. doi : 10.1007/978-3-642-13881-2\_8

Hussain, I., Kosseim, L. et Ormandjieva, O. (2013). Approximation of COSMIC functional size to support early effort estimation in Agile. *Data and Knowledge Engineering*, 85, 2-14. doi : 10.1016/j.datak.2012.06.005

Hussain, I., Ormandjieva, O. et Kosseim, L. (2009). Mining and Clustering Textual Requirements to Measure Functional Size of Software with COSMIC. Dans *Software Engineering Research and Practice* (p. 599-605).

ISO/IEC19761 :2011. (2011). *Software engineering – COSMIC : a functional size measurement method*. Geneva : International Organisation for Standardization.

Kaya, M. et Demirörs, O. (2011). E-cosmic : A business process model based functional size estimation approach. *Proceedings - 37th EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA 2011*. doi : 10.1109/SEAA.2011.60

Kitchenham, B. et Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Récupéré de Citeseer

Lavazza, L., Bianco, V. del et Liu, G. (2012). Analytical Convertibility of Functional Size Measures : A Tool-based Approach. Dans *Proceedings of the 2012 Joint Conference of the 22Nd International Workshop on Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement* (p. 160-169). Washington, DC, USA : IEEE Computer Society. doi : 10.1109/IWSM-MENSURA.2012.32

Lavazza, L., Del Bianco, V. et Liu, G. (2012). Analytical convertibility of functional size measures : a tool-based approach. Dans *Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement (IWSM-*

*MENSURA*), 2012 Joint Conference of the 22nd International Workshop on (p. 160-169). IEEE.

Li, Z., Nonaka, M., Kakurai, A. et Azuma, M. (2003). Measuring functional size of interactive software : A support system based on XForms-format user interface specifications. *Proceedings - International Conference on Quality Software*. doi : 10.1109/Q-SIC.2003.1319123

Lind, K. et Heldal, R. (2012). Comp size : A model-based and automated approach to size estimation of embedded software components. *IEICE Transactions on Information and Systems, E95-D(9)*, 2183-2192. doi : 10.1587/transinf.E95.D.2183

Marín, B., Pastor, O. et Giachetti, G. (2008). Automating the Measurement of Functional Size of Conceptual Models in an MDA Environment. Dans *Proceedings of the 9th International Conference on Product-Focused Software Process Improvement* (p. 215-229). Berlin, Heidelberg : Springer-Verlag. doi : 10.1007/978-3-540-69566-0\_19

Ochodek, M. (2016). Approximation of COSMIC functional size of scenario-based requirements in Agile based on syntactic linguistic features—a replication study. Dans *Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2016 Joint Conference of the International Workshop on* (p. 201-211). IEEE.

Ochodek, M. (2016). Functional size approximation based on use-case names. *Information and Software Technology, 80*, 73-88. doi : 10.1016/j.infsof.2016.08.007

Oriou, A., Bronca, E., Bouzid, B., Guetta, O. et Guillard, K. (2014, octobre). Manage the Automotive Embedded Software Development Cost Productivity with the Automation of a Functional Size Measurement Method (COSMIC). Dans *2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement* (p. 1-4). doi : 10.1109/IWSM.Mensura.2014.45

Özkan, B. (2011). Automated Functional Size Measurement for Three-Tier Object Relational Mapping Architectures. *Journal of Software Engineering, 21*, 311-338.

Peffer, K., Tuunanen, T., Rothenberger, M. A. et Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems, 24(3)*, 45-77. Récupéré de Taylor & Francis

Riehle, D. et Züllighoven, H. (1996). Understanding and using patterns in software development. *Theory and practice of object systems*, 2(1), 3-13. Récupéré de Wiley Online Library

Sag, M. A. et Tarhan, A. (2014). Measuring COSMIC Software Size from Functional Execution Traces of Java Business Applications. Dans *Proceedings of the 2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement* (p. 272-281). Washington, DC, USA : IEEE Computer Society. doi : 10.1109/IWSM.Mensura.2014.29

Sellami, A., Haoues, M. et Ben-Abdallah, H. (2013). Automated COSMIC-Based Analysis and Consistency Verification of UML Activity and Component Diagrams. *Communications in Computer and Information Science*, 417 CCIS, 48-63. doi : 10.1007/978-3-642-54092-9\_4

Soubra, H., Abran, A. et Sehit, M. (2015). Functional size measurement for processor load estimation in AUTOSAR. *Lecture Notes in Business Information Processing*, 230, 114-129. doi : 10.1007/978-3-319-24285-9\_8

Soubra, H., Abran, A., Stern, S. et Ramdan-Cherif, A. (2011). Design of a Functional Size Measurement Procedure for Real-Time Embedded Software Requirements Expressed Using the Simulink Model. Dans *Proceedings of the 2011 Joint Conference of the 21st International Workshop on Software Measurement and the 6th International Conference on Software Process and Product Measurement* (p. 76-85). Washington, DC, USA : IEEE Computer Society. doi : 10.1109/IWSM-MENSURA.2011.52

Trudel, S. et Abran, A. (2009). Functional size measurement quality challenges for inexperienced measurers. Dans *International Workshop on Software Measurement* (p. 157-169). Springer.

Trudel, S., Desharnais, J.-M. et Cloutier, J. (2016). Functional size measurement patterns : A proposed approach. Dans *Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2016 Joint Conference of the International Workshop on* (p. 23-34). IEEE.

Ungan, E. et Demirörs, O. (2015). A Functional Software Measurement Approach to Bridge the Gap Between Problem and Solution Domains. Dans A. Kobylński, B. Czarnacka-Chrobot, et J. Świerczek (Dir.) (dir.), *Software Measurement* (p. 176-191). Cham : Springer International Publishing.

Vogelezang, F. (2015). Guideline for Early or Rapid COSMIC Functional Size Measurement by Using Approximation Approaches.