

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

**EXPÉRIMENTATION D'UNE TECHNIQUE DE MESURE
DE LA RÉUTILISATION FONCTIONNELLE
DANS DES PROJETS**

**RAPPORT D'ACTIVITÉ DE SYNTHÈSE PRÉSENTÉ
COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN INFORMATIQUE DE GESTION**

**PAR
RAFAEL VALENCIA**

FÉVRIER 1999

RAPPORT FINAL D'ACTIVITÉ DE SYNTHÈSE

approuvé par le directeur de recherche : Alain ABRAN

M. ABRAN est
Professeur au Département d'informatique de l'Université du Québec à Montréal
Directeur du Laboratoire de recherche en gestion des logiciels de l'Université du
Québec à Montréal

et par le codirecteur de recherche : Robert DUPUIS

M. DUPUIS est
Professeur au Département d'informatique de l'Université du Québec à Montréal
Directeur du programme de maîtrise en informatique de gestion à l'Université du
Québec à Montréal

REMERCIEMENTS

Je voudrais remercier M. Alain Abran, professeur au Département d'informatique et directeur de cette activité de synthèse, par son support à la réalisation de cette étude.

Mes remerciements à M. Robert Dupuis, professeur au Département d'informatique, directeur du programme de Maîtrise en informatique de gestion et codirecteur de cette activité de recherche, pour son soutien continu.

Mes remerciements à tous les membres du Laboratoire de recherche en gestion des logiciels de l'UQAM (LRGL).

Merci beaucoup à mes parents, ma sœur et mon frère.

Enfin, ces remerciements s'adressent à toutes les personnes qui de près et de loin ont su me motiver à entreprendre cette recherche.

TABLE DES MATIÈRES

LISTE DES FIGURES	VI
LISTE DES TABLEAUX.....	VII
LISTE DES ABRÉVIATIONS.....	IX
RÉSUMÉ.....	X
INTRODUCTION.....	1
CHAPITRE I	
PRÉSENTATION DU TRAVAIL	3
1.1 Problématique de recherche.....	3
1.2 Limites du modèle de mesure (Abran et Desharnais, 1995).....	4
1.3 Objectifs et limites de la recherche.....	5
1.4 Démarche méthodologique	6
1.5 Définition du projet de recherche.....	8
1.5.1 Motivation, objet propos et le domaine de la recherche	8
1.5.2 Les utilisateurs.....	9
1.6 Les étapes du projet, intrants et livrables.....	10
CHAPITRE II	
CONTEXTE THÉORIQUE.....	12
2.1 Introduction.....	12
2.2 Une proposition de mesure de la réutilisation fonctionnelle.....	13
2.3 Les enjeux de la réutilisation des composants logiciels.....	17
2.3.1 Introduction.....	17
2.3.2 Problèmes de mise en pratique d'un programme de réutilisation	18
2.3.3 Bénéfices de la réutilisation	20
CHAPITRE III	
VÉRIFICATION DE LA MÉTHODE PROPOSÉE	21
3.1 Introduction.....	21
3.2 Structure du guide de validation	22
3.2.1 Cycle de vie d'une méthode de mesure.....	23
3.2.2 Vérification de la méthode proposée par l'entremise du guide de validation	26
3.3 La mesure utilisée dans cette approche	29
3.3.1 Informations requises aux mesures des points de fonction	30
CHAPITRE IV	
ÉLABORATION DE DONNÉES HISTORIQUES DE POINTS DE FONCTION	33
4.1 Caractéristiques du site industriel	33
4.2 Approche utilisée.....	33
4.3 Élaboration de bases de données historiques de points de fonction	37
4.3.1 La collecte de données	40

4.3.2	La validation de données.....	43
4.4	Analyse descriptive de base de données	44
4.4.1	Base de données Institution gouvernementale du Québec	45
4.4.2	Base de données Institution Financière (Valencia et Nguyen-kim)	46
4.4.3	Base de données institution financière (Abran et Robillard)	48
4.4.4	Base de données ISBSG.....	49
CHAPITRE V		
SCÉNARIOS DE TEST		53
5.1	Élaboration de scénarios de test	53
5.2	Application des scénarios de test.....	59
5.2.1	Simulation du scénario de test des fonctionnalités majeures	61
5.2.2	Impact de la méthode de mesure proposée dans le coût unitaire.....	71
5.3	Conclusion sur les scénarios de test utilisés.....	73
CHAPITRE VI		
INTERPRÉTATION DE RÉSULTATS.....		74
6.1	Contexte d'interprétation.....	74
6.2	Approche d'analyse empirique.....	74
6.3	Extrapolation des résultats.....	77
6.4	Conclusion	79
6.5	Suggestion de recherche	81
6.6	Autres suggestions de recherche	82
APPENDICE A		
CADRE DE BASILI SOMMAIRE DE L'EXPÉRIENCE		84
APPENDICE B		
SCÉNARIOS DE TEST DE FONCTIONNALITÉS MOYENNES		
SIMULATION DU SCÉNARIO DE TEST DES FONCTIONNALITÉS MOYENNES		88
APPENDICE C		
SCÉNARIOS DE TEST DES FONCTIONNALITÉS MINEURES		
SIMULATION DU SCÉNARIO DE TEST DES FONCTIONNALITÉS MINEURES		98
APPENDICE D		
SCÉNARIOS DE TEST DES FONCTIONNALITÉS PETITES		
SIMULATION DU SCÉNARIO DE TEST DES FONCTIONNALITÉS PETITES		108
APPENDICE E		
L'ANALYSE DE POINTS DE FONCTION DESCRIPTION SOMMAIRE DE		
L'ANALYSE DE POINTS DE FONCTION.....		118
APPENDICE F		
ARTICLE « MEASUREMENT OF FUNCTIONAL REUSE IN MAINTENANCE »		
(ABRAN ET DESHARNAIS, 1995).....		125
BIBLIOGRAPHIE.....		141

LISTE DES FIGURES

Figure	Page
3.1 Processus de mesure	23
3.2 Cycle de vie d'une méthode de mesure.....	24
5.1 Exemple qui illustre la formulation des besoins de l'utilisateur	54
5.2 Exemple qui illustre les différentes transformations des formules de la méthode de mesure proposée.....	55
5.3 Exemple qui illustre l'analyse des résultats de l'application du modèle de productivité.....	57
E.1 Matrices de complexité pour les entrées et les sorties	120
E.2 Matrices de complexité pour les fichiers logiques internes et externes.....	121

LISTE DES TABLEAUX

Tableau	Page
1.1 Les quatre phases du cadre de recherche de Basili.....	7
1.2 Les étapes, les intrants et les livrables du projet de recherche	10
4.1 Collecte de données d'un projet ou une application informatique	35
4.2 Pourcentage pour chaque type de fonction	36
4.3 Distribution fonctionnelle moyenne de points de fonction : une organisation gouvernementale et une institution financière canadienne.....	38
4.4 Distribution fonctionnelle moyenne de points de fonction dans une institution financière canadienne.....	39
4.5 Distribution fonctionnelle moyenne de points de fonction tirée du Worldwide Software Development The Benchmark Release 5 ISBSG (1998).....	40
4.6 Pays d'origine, nombre de projets et le pourcentage de projets par pays d'origine selon Worldwide Software Development The Benchmark Release 5 ISBSG (1998).....	50
5.1 Exemple qui illustre l'analyse des résultats de l'application du modèle de productivité	56
5.2 Analyse des résultats (Données Valencia et Desharnais).....	64
5.3 Analyse des résultats (Données Valencia et Nguyen-Kim).....	66
5.4 Analyse des résultats (Données Abran).....	68
5.5 Analyse des résultats (Données I.S.B.G.).....	70
5.6 Sommaire des résultats des différentes transformations de la méthode de mesure de réutilisation pour les différents scénarios de tests appartenant au regroupement des fonctionnalités majeures	72

B.1	Analyse de résultats (Données Valencia et Desharnais).....	90
B.2	Analyse des résultats (Données Valencia et Nguyen-Kim).....	92
B.3	Analyse des résultats (Données Abran).....	94
B.4	Analyse des résultats (Données I.S.B.G.).....	96
B.5	Sommaire des résultats des différentes transformations de la méthode de mesure de réutilisation pour les différents scénarios de tests appartenant au regroupement des fonctionnalités moyennes.....	97
C.1	Analyse de résultats (Données Valencia et Desharnais).....	100
C.2	Analyse des résultats (Données Valencia et Nguyen-Kim).....	102
C.3	Analyse des résultats (Données Abran).....	104
C.4	Analyse des résultats (Données I.S.B.G.).....	106
C.5	Sommaire des résultats des différentes transformations de la méthode de mesure de réutilisation pour les différents scénarios de tests appartenant au regroupement des fonctionnalités mineures.....	107
D.1	Analyse de résultats (Données Valencia et Desharnais).....	110
D.2	Analyse des résultats (Données Valencia et Nguyen-Kim).....	112
D.3	Analyse des résultats (Données Abran).....	114
D.4	Analyse des résultats (Données I.S.B.G.).....	116
D.5	Sommaire des résultats des différentes transformations de la méthode de mesure de réutilisation pour les différents scénarios de tests appartenant au regroupement des fonctionnalités petites.....	117
E.1	Pondération des composantes	121
E.2	Les 14 caractéristiques générales du système tel que déterminé par l'analyse de points de fonction.....	123
E.3	Modèle de mesure de l'analyse de points de fonction.....	124

LISTE DES ABRÉVIATIONS

ED	Éléments de données
FR	Fichiers références
GDE	Groupe de données externes
GDI	Groupe de données internes
IFPUG	International Function Point User Group
ISBSG	International Software Benchmarking Standards Group
M.I.S.	Management Information System
PDR	Project Delivery Rate
PF	Points de fonction
PFA	Points de fonction ajustés
PfnA	Points de fonction non ajustés
PWE	Project Work Effort
TDI	Total des degrés d'influence
3GL	Langage de troisième génération
4GL	Langage de quatrième génération
JP	Jours Personnes
JP/PF	Jours Personnes par point de fonction

RÉSUMÉ

L'article « Measurement of Functional Reuse in Maintenance » (Abran et Desharnais, 1995) identifie une méthode pour mesurer la réutilisation des composants logiciels dans les projets où des fonctionnalités ont été ajoutées aux applications existantes. La méthode proposée dans cet article, est basée sur la mesure de la réutilisation à partir d'une perspective fonctionnelle plutôt que technique. La présente étude se veut une vérification de cette méthode soit du point de vue théorique soit empirique. L'article en question est présenté dans l'appendice F.

Les objectifs de cette recherche sont au nombre de deux :

- Vérifier la généralisabilité de la méthode proposée par Abran et Desharnais (1995) à d'autres applications du domaine M.I.S.
- Identification à l'aide des scénarios de test (simulations) des situations dans lesquelles le modèle produit des prévisions positives et négatives (en terme de jours personnes) concernant le développement des applications informatiques qui utilisent des composants fonctionnels réutilisables.

Reliés aux objectifs, les livrables finaux du travail seront :

- Un cadre de vérification du modèle
- Différents scénarios de test (simulations).
- Deux bases de données historiques de points de fonction. Elles ont été élaborées en partie par l'auteur de ce travail de recherche, à partir des projets informatiques provenant d'une institution financière canadienne et d'un organisme gouvernemental de Québec.

Dans ce projet de recherche, on expérimente à partir de deux approches. La première approche théorique compare la méthode par rapport au processus de mesure et au cycle de vie d'une mesure proposé par Jacquet et Abran (1996). La deuxième approche consiste à vérifier les différents résultats de l'application de la méthode à partir de différents scénarios de tests. Ce travail de recherche permet de constater que la méthode de mesure est conforme au processus de mesure et au cycle de vie d'une mesure proposée par Jacquet et Abran (1996). Concernant la généralisation à d'autres contextes, le modèle proposé s'adapte facilement aux applications M.I.S. et permet de dégager certains bénéfices budgétaires(jours personnes) sur tous les contextes de travail utilisés dans cette recherche. Néanmoins, cette recherche se limite seulement aux applications M.I.S. car on n'a pas les données nécessaires pour sa vérification à d'autres domaines.

Mots clés : informatique, méthode, mesure, réutilisation, points de fonction, modèle d'estimation, modèle de productivité

INTRODUCTION

La réutilisation des composantes informatiques, suscite un intérêt pour les organisations afin de limiter la croissance de budgets de plus en plus importants dédiés au volet logiciel.

Dû à l'envergure des budgets alloués, les gestionnaires des projets informatiques envisagent la réutilisation comme une solution afin de mieux gérer leur budget. Ce contexte pose des défis importants au génie logiciel. Parmi eux, l'utilisation d'un modèle qui permettra de mieux estimer l'utilisation des composants réutilisables. Cette approche offre des voies intéressantes, incluant l'amélioration d'un modèle de mesure de la réutilisation fonctionnelle présenté par Abran et Desharnais (1995).

Dans ce projet de recherche, nous proposons un cadre de travail afin de vérifier la méthode de mesure de la réutilisation fonctionnelle décrite dans l'article "Measurement of Functional Reuse in Maintenance" (Abran et Desharnais, 1995). Les concepts de cette méthode seront analysés à partir des concepts tirés du processus de mesure (Jacquet et Abran, 1997) et testés avec des bases de données historiques de points de fonction. Ces données compilées pour ce travail proviennent de deux institutions financières canadiennes d'un organisme gouvernemental de Québec et d'une autre tirée de la littérature (ISBSG, 1998). L'utilisation de ces bases de données permettra d'éclaircir et déterminer les limites du modèle.

Dans ce document, nous présentons la problématique, ainsi que la démarche méthodologique qui guidera la mise en œuvre de ce projet de recherche. Cette démarche est basée sur le cadre de recherche expérimentale dans le domaine du génie logiciel proposé par Basili et al (1986). Nous formulons dans le chapitre I, les objectifs de la recherche, la motivation, l'objet d'étude, ainsi que les propos et les divers utilisateurs à qui s'adresse cette activité de recherche.

Le deuxième chapitre porte sur la présentation de la méthode de mesure de réutilisation fonctionnelle proposée par Abran et Desharnais (1995), ainsi que sur les enjeux principaux de la réutilisation des composants logiciels.

Le troisième chapitre porte sur la revue de la littérature. Les références sont regroupées de manière à ce qu'elles soient sous-jacentes à la problématique de cette étude. On décrit le processus de vérification de la mesure proposée, ainsi qu'un bref aperçu de la méthode de points de fonction qui a été la source d'inspiration dans l'élaboration de la méthode de la réutilisation fonctionnelle proposée.

Le quatrième chapitre est consacré à la présentation de deux nouvelles bases historiques de points de fonctions. On décrira les moyens utilisés, leur provenance, ainsi que la procédure à suivre pour les créer.

Dans le cinquième chapitre nous présentons l'élaboration des différents scénarios de tests qui serviront de simulations dans la vérification de la méthode de mesure de la réutilisation proposée par Abran et Desharnais (1995).

Dans le dernier chapitre, nous présentons l'analyse des résultats, ce qui constitue la dernière phase du cadre de Basili, et qui porte sur l'interprétation des résultats, leur extrapolation, ainsi que l'identification de pistes futures qui pourraient surgir à partir des résultats de cette recherche.

CHAPITRE I

PRÉSENTATION DU TRAVAIL

1.1 PROBLÉMATIQUE DE RECHERCHE

Le besoin d'un programme de mesures comme faisant partie du processus d'amélioration du développement et l'amélioration du logiciel, est un fait reconnu dans la littérature. La mesure du logiciel est une composante fondamentale afin d'arriver à une maturité de la technologie du logiciel. Une des techniques qui est présentée dans la littérature comme étant la clé pour améliorer et pallier au coût croissant du développement logiciel est la réutilisation des composants logiciels. Cependant, peu d'études ont analysé comment mesurer cette réutilisation du point de vue fonctionnel, ainsi que les avantages et les inconvénients qui en découlent. Abran et Desharnais (1995), ont proposé une technique de mesure de la réutilisation de composants logiciels à partir d'une perspective fonctionnelle, plutôt qu'à partir d'une perspective technique. La réutilisation logicielle du point de vue fonctionnel consiste à utiliser des artefacts logiciels existants lors du développement d'une nouvelle application informatique selon les besoins exprimés par les utilisateurs.

La mesure proposée par Abran et Desharnais (1995) possède un potentiel intéressant dans le développement logiciel. Cependant la mesure proposée, n'a pas été vérifiée dans un autre contexte que celui de son élaboration initiale. Ce fait devient la problématique à laquelle ce travail de recherche doit s'attaquer. Cette étude se veut une vérification de cette méthode de mesure, car on ne sait dans quel contexte cette méthode peut être généralisée. Dans cette vérification, la conformité avec le processus de mesure et le cycle de vie d'une mesure proposé par Jacquet et Abran (1996) est le premier axe à explorer. Le deuxième axe est de vérifier la mesure proposée avec un ensemble de données industrielles. Ces deux axes exploratoires sont à

la base de la vérification de la plausabilité, la complétude, la robustesse et la fiabilité de la mesure de la réutilisation fonctionnelle proposée.

La question de recherche à laquelle nous contribuons à répondre découle de ces deux axes de recherche et elle est la suivante : Est-ce que la méthode de mesure de la réutilisation fonctionnelle, qui a été développée dans un seul contexte expérimental tel que proposé par Abran et Desharnais (1995), peut être généralisée à d'autres contextes de travail du domaine M.I.S. ?

1.2 LIMITES DU MODÈLE DE MESURE (ABRAN ET DESHARNAIS, 1995)

Les limites de la méthode de mesure de la réutilisation fonctionnelle proposée par Abran et Desharnais (1995) identifiées dans cette étude de recherche sont au nombre de trois :

- La première limite est le fait que la méthode a été vérifiée à partir d'un exemple d'un modèle de productivité dans lequel le nombre de jours personnes nécessaires soit dans le développement, soit dans la maintenance, n'est pas nécessairement le même qui a été utilisé dans l'étude faite par Abran et Desharnais (1995). Nous utilisons des données provenant de la base de données ISBSG (ISBSG, 1998).
- La deuxième limite concerne les bases de données historiques de points de fonction utilisées dans la méthode de mesure. Nous considérons dans cette étude deux limitations spécifiques à l'utilisation de ces bases de données historiques. Premièrement, il faut signaler la variation des pourcentages des composants des différentes bases historiques de points de fonction, car ces pourcentages influencent les résultats du modèle. Cela est dû aux différents types d'applications qui ont servi à former les bases de données historiques de points de fonction, ainsi que le secteur d'activité dans lequel l'organisation se positionne. En deuxième lieu, ces bases de données historiques, ne représentent qu'un pourcentage limité des diverses applications informatiques du domaine M.I.S.
- La troisième limite concerne le nombre de scénarios de tests à partir desquels la méthode sera vérifiée. Originellement, un seul scénario fait sur mesure a été utilisé afin de tester la méthode proposée. Dans la présente étude nous avons élaboré quatre scénarios de test dans le but de tester la méthode proposée. Néanmoins ces quatre scénarios de tests ne sont pas exhaustifs. Ils ont été élaborés afin de refléter les fonctionnalités demandées dans deux organisations observées. Deux bases de données historiques de points de fonction utilisées dans cette étude ont été élaborées avec des données issues de ces deux organisations.

1.3 OBJECTIFS ET LIMITES DE LA RECHERCHE

Les objectifs de ce projet de recherche sont au nombre de deux :

- Vérifier la généralisation de la méthode proposée à d'autres applications du domaine M.I.S.
- L'identification des scénarios de travail dans lesquels la méthode de réutilisation fonctionnelle proposée présente des avantages budgétaires exprimés en termes de jours personnes.

De plus, ce travail a permis l'élaboration de deux nouvelles bases de données historiques de points de fonction. Elles ont été élaborées à partir de plusieurs projets informatiques qui ont été mesurés dans une institution financière canadienne et un organisme gouvernemental du Québec. Tous ces projets informatiques mesurés appartiennent au domaine M.I.S. Ces bases de données historiques de points de fonction permettent de « tester » la méthode proposée par Abran et Desharnais (1995) à partir de données autres que celles utilisées dans son contexte initial.

En ce qui concerne les limites de la présente étude, on a identifié les suivantes comme étant les principales :

- Cette étude est basée seulement sur les concepts élaborés dans l'article « Measurement of Functional Reuse in Maintenance » (Abran et Desharnais, 1995)
- Ce projet est vérifié à partir de quatre bases de données de points de fonction qui proviennent de deux institutions financières canadiennes, d'un organisme gouvernemental du Québec et une autre au niveau international (ISBSG, 1998).
- Les différents scénarios de travail sont élaborés à partir d'exemples de situations qui se sont présentées dans plusieurs entreprises.
- Les scénarios de travail de la présente étude sont limités à des situations où des applications M.I.S. ont été développées, car les projets mesurés afin de tester la méthode proposée appartiennent à ce domaine. Jusqu'à présent la méthode de points de fonction est très

adéquate pour les projets M.I.S. Il existe actuellement peu d'études et recherches sur l'application de la méthode de points de fonction à d'autres domaines tels que le temps réel (St-Pierre et al., 1997) et l'orienté objet (Fetcke et al., 1996), et il faudra les vérifier rigoureusement avant de les appliquer dans l'industrie.

1.4 DÉMARCHE MÉTHODOLOGIQUE

La démarche méthodologique à suivre dans cette recherche est basée sur le cadre d'expérimentation élaboré par Basili et al. (Basili et al., 1986). Ce cadre conceptuel comprend quatre phases de déroulement d'une expérience : la définition, la planification, l'exécution et l'interprétation. À l'intérieur de chaque phase, certaines activités sont proposées. Ce cadre fournit un protocole de recherche simple et facile à utiliser. Un gabarit de ce cadre est présenté dans le tableau 1.1. Il permettra de réaliser et de documenter ce projet de recherche exploratoire.

Ce cadre de recherche de Basili est composé de quatre parties principales : la définition, la planification, l'exécution et l'interprétation des résultats. Chacune de ces parties est composée à son tour de plusieurs étapes à réaliser. Dans chaque étape, plusieurs activités sont nécessaires afin de bien encadrer son contenu.

On a choisi ce cadre de recherche, car on considère qu'il s'adapte très bien au type de recherche qui sera réalisée dans la présente étude. En plus, ce cadre présente toutes les phases et activités de la méthode scientifique, ce qui permet de bien encadrer toutes sortes de recherches et en particulier celles qui sont faites dans le domaine du logiciel.

Tableau 1.1

Les quatre phases du cadre de recherche de Basili

Cadre de recherche de Basili					
Phase 1. Définition					
Motivation	Objet	But	Perspective	Domaine	Portée
Phase 2. Planification					
Conception expérimentale		Critères	Sélection des mesures		
Phase 3. Réalisation					
Préparation		Déroulement		Analyse des données	
Phase 4. Interprétation					
Contexte d'interprétation			Extrapolation des résultats		

La partie définition est composée des étapes suivantes : la motivation, l'objet, l'objectif, la perspective, le domaine et l'étendu de l'étude. La conception expérimentale, les critères et la sélection des mesures, ainsi que ses définitions font partie de la phase de la planification. La phase de l'exécution est formée des parties suivantes : la préparation, l'exécution et l'analyse de données. Finalement la quatrième phase est celle de l'interprétation qui est composée de deux parties : le contexte d'interprétation et l'extrapolation des résultats.

À l'intérieur de toutes les phases de développement du cadre de Basili, on énonce d'une manière générale les grandes lignes qui guideront les différentes activités qui seront développées dans cette recherche. L'ensemble des quatre phases sera présenté dans cette étude. Un résumé du gabarit du cadre dûment rempli est aussi présenté dans l'appendice A.

1.5 DÉFINITION DU PROJET DE RECHERCHE

Dans cette section on présente la première étape du cadre méthodologique de Basili. Cette première phase appelée la définition est composée de quatre parties : la motivation, l'objet, le propos, l'étendu, le domaine et les utilisateurs. Dans cette partie, nous formulerons les objectifs et les limites de cette recherche qui seront liés aux différentes étapes et aux livrables du projet.

1.5.1 Motivation, objet propos et le domaine de la recherche

Dans la plupart des organisations avec un service informatique, le budget alloué aux technologies de l'information, et spécialement au développement et à la maintenance du logiciel est assez élevé.

Afin de mieux contrôler le pourcentage du budget dédié au développement et à la maintenance, la réutilisation des composants informatiques est une avenue qui fait des progrès dans la communauté informatique. Cette approche, malgré tout, n'a pas été évaluée suffisamment en ce qui concerne sa rentabilité économique. En ayant pour objectif de montrer cette rentabilité économique de l'utilisation des composants informatiques, Abran et Desharnais (1995) ont élaboré une méthode de mesure de la réutilisation fonctionnelle. Toutefois, cette méthode n'a été développée qu'avec un seul contexte expérimental.

L'objet de cette étude est d'évaluer la méthode de réutilisation fonctionnelle avec une série de cas concrets qui ont été élaborés à partir des données industrielles. Dans cette étude, nous évaluerons une mesure de réutilisation fonctionnelle, ainsi que le modèle de productivité contenu dans ce modèle. Ce dernier sert à montrer les gains potentiels pour le budget informatique, exprimés en jours personnes si on utilise la méthode proposée.

Le propos de ce travail de recherche concerne l'élaboration d'un cadre de travail qui permettra d'évaluer l'objet de cette étude, soit la méthode de mesure de la réutilisation fonctionnelle, telle que décrite.

Le domaine du projet nous permettra de connaître la nature des données industrielles et leur provenance. On identifiera aussi le domaine d'application du logiciel. Cette procédure nous permettra de bien identifier et classer les bases de données historiques de points de fonction qui ont été créées et utilisées.

1.5.2 Les utilisateurs

Les utilisateurs qui bénéficieront des différents livrables de ce projet seront les suivants :

- Les gestionnaires en informatique et plus particulièrement les chefs de projets qui travaillent avec des applications de développement et maintenance logiciel, car ils disposeront de nouveaux outils de gestion.
- Les différents intervenants dans le domaine de l'impartition. Cette étude leur permettra d'avoir un instrument de travail afin de mieux élaborer leur stratégie d'affaires.
- Les chercheurs en mesures et en modèles de productivité, qui profiteront des éléments contenus dans cette étude et qui trouveront de nouvelles pistes pour des recherches futures.

1.6 LES ÉTAPES DU PROJET, INTRANTS ET LIVRABLES

Dans cette étude, on a identifié trois étapes principales pour la réalisation du livrable final. Des considérations d'ordre théorique et pratique sont à la base des intrants de chaque étape. Tout au long de ce projet, l'auteur extrapolera des concepts théoriques et les transplantera aux cas pratiques vécus dans l'industrie. La partie théorique sera développée à partir d'une revue de la littérature spécialisée, tandis que pour la partie pratique par des expériences directes vécues dans l'industrie par l'auteur de cette recherche.

Le produit de chaque étape qui compose cette recherche sera un livrable. Les livrables associés à chaque étape de cette étude seront des livrables intermédiaires ou finaux par rapport aux résultats du projet. Les livrables de chaque étape sont assemblés entre eux afin de constituer un livrable final qui sera le résultat de ce projet de recherche. L'ensemble des livrables finaux correspondront aux objectifs de cette recherche. Une vue synthétique des étapes, les intrants et livrables de cette activité de recherche, est présentée dans le tableau 1.2.

Tableau 1.2

Les étapes, les intrants et les livrables du projet de recherche

Étapes	Intrants	Livrables
Étape 1 . Positionnement de la méthode proposée dans le développement logiciel	. Revue de la littérature . Cas concrets extraits des sites industriels, concernant la mesure de productivité dans les projets informatiques .Utilisation des bases historiques de points de fonction pour la vérification de la méthode proposée	. Justification de la généralisation de la méthode de mesure de la réutilisation fonctionnelle (Livrable final du projet)

Étape 2 . Élaboration de bases historiques de points de fonction	. Données provenant de plusieurs projets mesurés à l'aide de la technique de points de fonction	. Création de deux nouvelles bases historiques de points de fonction. (Livrable intermédiaire)
Étape 3 .Élaboration de scénarios de test de la méthode proposée à partir de cas industriels concrets et pertinents.	. Des éléments et situations concrètes vécues dans l'industrie.	. Création des scénarios de tests. (Livrable intermédiaire)

Le cadre de Basili et al. (1986) s'avère une démarche logique, rigoureuse, facile à comprendre et à mettre en pratique. Elle n'est pas trop lourde et positionne les activités de mesure. En plus le cadre de Basili et al. (1986) fait ressortir l'importance des activités connexes de définition, de planification et d'exploitation.

CHAPITRE II

CONTEXTE THÉORIQUE

2.1 INTRODUCTION

Nous présentons dans ce chapitre les références bibliographiques correspondant aux considérations exposées dans la partie de la problématique décrite dans le chapitre I, et qui forment la base de ce travail de recherche

Cette revue de littérature est composée de quatre sections qui structure la base fondamentale à la présente étude. En premier lieu, on présente un résumé de l'article « Measurement of Functional Reuse in Maintenance » (Abran et Desharnais, 1995). L'article au complet est présenté dans l'appendice E. Cet article se veut une méthode de mesure de la réutilisation fonctionnelle. Les concepts et les hypothèses de travail de cet article seront vérifiées dans la présente étude.

Dans la deuxième section, on évaluera la méthode proposée par rapport aux critères de vérification établis dans le guide de vérification des méthodes de mesure de taille fonctionnelle (Jacquet et Abran, 1997). Ce guide fait partie d'un document ISO/IEC DIS 14143-3 sur une proposition d'une norme pour les méthodes de mesure de taille fonctionnelle. Cela permettra de vérifier que la méthode est construite avec une base théorique solide, avec des propriétés appropriées.

En troisième lieu, nous présenterons une méthode de collecte de données industrielles qui nous permettra d'illustrer le comportement de la méthode proposée dans des contextes industriels. Dans cette section, on décrira aussi le processus de collecte de données de points de fonction qui sert à la construction de bases de données historiques de points de fonction. Ces données industrielles nous permettront d'illustrer le comportement de la méthode proposée, à partir de contextes industriels.

Finalement, on analysera la pertinence de la méthode de la réutilisation fonctionnelle, son utilisation dans les analyses de productivité et d'estimation, ainsi que différents bénéfices engendrés par son utilisation.

Cette revue de la littérature, en plus d'être le fondement de la présente étude, nous permettra aussi de suggérer dans un contexte industriel dans quelle direction la méthode devrait être utilisée et dans quelles directions pourraient s'orienter les travaux futurs.

2.2 UNE PROPOSITION DE MESURE DE LA RÉUTILISATION FONCTIONNELLE

L'approche proposée dans l'article « Measurement of Functional Reuse in Maintenance » (Abran et Desharnais, 1995), présente une stratégie qui identifie et mesure d'une part les transactions qui n'ont pas dû être développées et d'autre part, elle quantifie les bénéfices dérivés à partir du coût qui a été évité pour ne pas rédévelopper ces transactions.

L'article « Measurement of Functional Reuse in Maintenance » identifie et mesure la réutilisation des composants logiciels dans les projets où des fonctionnalités ont été ajoutées aux applications existantes. L'approche est basée sur la mesure de la réutilisation à partir

d'une perspective fonctionnelle plutôt que technique. La mesure proposée se veut une approche qui permet de déterminer les bénéfices de la réutilisation fonctionnelle.

Deux concepts clés sont introduits : un indicateur de réutilisation et un ratio prédicteur. L'indicateur de réutilisation est dérivé à partir des concepts définis dans l'analyse de points de fonction (IFPUG, 1994) et de la notion « black box reuse ». Le ratio prédicteur est basé sur la compréhension de la notion « coûts évités si on réutilise au lieu de développer » et la manière d'aller capturer ces « coûts évités » à partir de bases de données historiques de points de fonction (PF).

Le premier concept clé, est l'indicateur de réutilisation, lequel est dérivé de l'arrimage de trois définitions. Les deux premières définitions font partie de la méthode de points de fonction, tandis que la troisième fait partie de la notion « black box reuse ».

La première définition est l'identification de la frontière du comptage. D'après le guide de comptage de points de fonction (IFPUG, 1994), une frontière indique les limites entre l'application ou le projet que l'on mesure et les applications externes ou le domaine de l'utilisateur. Une frontière établit quels sont les composants qui sont compris dans le comptage des points de fonction.

La deuxième définition est appelée Groupe logique de données externes (GDE). Un GDE est un groupe de données référencé dans l'application, identifiable par l'utilisateur comme logiquement lié, mais maintenu dans les frontières d'une autre application (IFPUG, 1994).

La troisième définition fait appel à la notion « black box reuse », qui permet de réutiliser une composante telle quelle, sans aucune modification. Les auteurs de l'article font l'arrimage entre ces trois concepts afin de définir l'indicateur de réutilisation. À chaque fois qu'un GDE se trouve à l'intérieur de la frontière de l'application à mesurer, il devient une composante

logicielle qu'on peut réutiliser sans aucune modification, afin de respecter la notion « black box reuse ».

Le deuxième concept clé, le ratio prédicteur représente comment le concept « coûts évités si on réutilise au lieu de développer » peut être capturé à partir de bases de données historiques de points de fonction. Ces bases de données historiques de points de fonction ont été collectées dans plusieurs sites industriels. Cette information a été intégrée dans des modèles de productivité par l'entremise de la combinaison de l'indicateur de réutilisation et du ratio prédicteur de la base de données référencée.

La mesure de la réutilisation fonctionnelle proposée, est composée d'une série de formules mathématiques, qui par l'entremise de certaines transformations mathématiques combinent le ratio prédicteur et l'indicateur de réutilisation. Dans les lignes suivantes, on fera un survol des différentes étapes qui composent la méthode de la réutilisation fonctionnelle proposée.

- Étape 1. Dans la méthode de points de fonction, on détermine le nombre total de points de fonction en appliquant la formule suivante :

Formule 1

$$PF = PF(GDI) + PF(GDE) + PF(ENTRÉES) + PF(SORTIES) + PF(INTERROGATION)$$

- Étape 2. L'indicateur de réutilisation est déterminé de la manière suivante :

Formule 2

$$\text{Indicateur de réutilisation} = PF(GDE)^E = \text{Nombre de GDI à réutiliser}$$

Cet indice offre une mesure des données réutilisées dans l'application mesurée. Ce sont des données qui ont été développées et qui existent en dehors de la frontière de l'application mesurée. La notion « black box reuse » est utilisée dans la détermination de cet indicateur.

- Étape 3. Le ratio prédicteur est nécessaire afin d'estimer le nombre de transactions réutilisées. Ce modèle peut être dérivé à partir de bases de données historiques de points de fonction, en identifiant combien points de fonction de chaque type de fonction sont nécessaires en moyenne pour chaque donnée interne de la base de donnée référencée. Ce ratio prédicteur informe sur les transactions qui ne doivent pas être développées, car elles ont déjà été développées en dehors de la frontière de l'application mesurée. Ce ratio prédicteur est dérivé à partir de la base de données historique de points de fonction en calculant le ratio des points de fonction des autres types de fonction (en excluant les points de fonctions des GDI), et en divisant par la suite, ce total par le nombre de points de fonction des GDI.

Formule 3

$$\text{Ratio prédicteur de réutilisation} = \frac{\text{PF(GDE)}^N + \text{PF(ENTRÉES)}^N + \text{PF(SORTIES)}^N + \text{PF(INTERROGATION)}^N}{\text{PF(GDI)}^N}$$

- Étape 4. Avec l'information disponible de la distribution des ratios des cinq types de fonctions extraite à partir des projets de développements antérieurs, ainsi que leur coût de développement il est possible de quantifier les fonctions à ne pas développer (évitées) dû à la réutilisation ainsi que leurs coûts. La formule suivante met en évidence les arguments ci dessus.

Formule 4

$$\text{Transactions évitées} = \text{Transactions réutilisées}$$

Formule 5

$$\text{Transactions évitées} = \text{Indicateur de réutilisation} \times \text{Ratio prédicteur de réutilisation}$$

Formule 6

$$\text{Transactions évitées} = \text{PF (indicateur)}^E \times \text{PF}^R \text{ (prédicteur)}$$

Quand une des ces formules qui utilise les transactions évitées est ajoutée à la formule du comptage de points de fonction, tel que défini par le guide de comptage de points de fonction, l'équation résultant est la suivante :

Formule 7

$$PF = PF \text{ (demandé par l'utilisateur)} + \text{transactions évitées}$$

De l'équation antérieure on dégage la formule finale suivante :

Formule 8

$$PF = PF \text{ (demandé par l'utilisateur)} + (PF \text{ (indicateur)} \times PF \text{ (prédicteur)})$$

À titre d'illustration on présente dans la section 5.2 du chapitre V un exemple de l'utilisation de cette méthode. Les appendices B, C et D illustrent les autres scénarios de test utilisés dans cette étude.

2.3 LES ENJEUX DE LA RÉUTILISATION DES COMPOSANTS LOGICIELS

2.3.1 Introduction

La réutilisation logicielle consiste à utiliser des artefacts logiciels existants lors du développement d'une nouvelle application informatique. Ceci inclut les fragments de code source, ainsi que les structures de conception et d'implantation, les spécifications, la documentation, etc. (Freeman, F., 1983 et Krueger, C.W. 1992). Il existe plusieurs raisons pour lesquelles la réutilisation est souhaitable. Les deux raisons majeures sont : 1) augmenter la productivité et 2) l'atteinte d'une meilleure qualité des logiciels (Tracz, W., 1987).

Plusieurs études sur la réutilisation démontrent que 40% à 60% du code est réutilisable d'une application informatique à une autre (Biggerstaff, T.J. et Perlis, A.J. 1984). D'après Lanergan, R.G. et Grasso, C.A. (1984) 60% de la conception et du code des applications d'affaires est réutilisable.

Plusieurs expériences en industrie suggèrent que la réutilisation peut atteindre un taux assez élevé en terme de productivité et de qualité (Basili, V.R. et Musa, J.D., 1991). Un autre auteur Lim, W.C. (1994) suggère que la réutilisation accélère la livraison des produits et un retour sur l'investissement très avantageux.

En général, théoriquement, les systèmes développés à partir de la réutilisation des composants logiciels coûtent moins cher. Des gains de productivité se font dans la réduction des coûts de développement. Deux facteurs permettent d'expliquer ces gains de productivité : un délai de livraison plus court et l'élaboration du code avec moins d'erreurs. Le programmeur écrit moins de code, la réutilisation réduit la quantité de documentation et de tests et le système est plus facile à maintenir, car les programmeurs sont plus familiers avec les composants réutilisables qui ont servi à concevoir le système.

2.3.2 Problèmes de mise en pratique d'un programme de réutilisation

Plusieurs chercheurs et praticiens sont d'accord pour dire que la mise en pratique d'un programme de réutilisation pose des problèmes de taille. Plusieurs facteurs empêchent de mettre en pratique un programme de réutilisation.

- La non-maturité du développement de logiciel, soit comme science ou discipline (Dijkstra, 1989; Shaw, 1990).
- La planification inadéquate du développement de logiciel (Brooks, 1975).

- La planification inadéquate du développement de la réutilisation (Woodfield, Embley et Scott, 1987).
- La gestion à court terme de la réutilisation (Gruman, 1988).
- Le manque de méthodologies et d'outils de support à la réutilisation (Fischer, 1987)

La réutilisation systématique est difficile et risquée à gérer car dans son processus elle nécessite une cohérence efficace parmi les différents intervenants. Du point de vue organisationnel, la réutilisation a besoin d'un support à long terme. Ce support doit regarder la réutilisation comme un investissement amortissable à moyen et à long terme.

Du point de vue économique, la réutilisation doit prendre en compte la notion de coûts et de bénéfices. Le coût encouru pour l'élaboration des composants réutilisables doit être amorti par l'utilisation de ces composants.

Du point de vue légal, il faut définir quels sont les droits et les obligations des fournisseurs et des consommateurs des composants réutilisables.

Du point de vue cognitif, les concepteurs et les utilisateurs ont besoin d'un modèle ou d'une représentation mentale de la composante réutilisable afin de la concevoir et de l'utiliser convenablement.

Du point de vue technique, deux approches de réutilisation sont généralement utilisées : l'approche construction de blocs, qui est basée sur le développement du nouveau logiciel à partir des produits existants, et l'approche « génératif » qui est basée sur la réutilisation des processus qui sont encodés dans un processeur tel qu'un compilateur, un générateur de code, etc. Ces approches posent un certain nombre de problèmes concernant leur construction et leur utilisation (Horowitz et Munson, 1984). Ces différents points de vue ne feront pas l'objet de la présente étude. On s'attardera uniquement sur la mesure de la réutilisation fonctionnelle et sur son utilisation pour des analyses de productivité.

2.3.3 Bénéfices de la réutilisation

Dans les sous-sections qui suivent on fera un survol des bénéfices que les différents intervenants dans le domaine logiciel pourraient réaliser avec la réutilisation. On abordera cette partie à partir de deux approches. La première approche est le développement de produits réutilisables, tandis que la deuxième est le développement avec des composantes réutilisables.

2.3.3.1 Réutiliser les composants déjà existants

La réutilisation de composants déjà existants dans une application informatique, présente pour une organisation les avantages suivants :

- Accélérer la création, l'entretien, le support et la formation;
- respecter les standards;
- optimiser les investissements;
- diminuer les risques du produit logiciel final;
- assurer une meilleure qualité.

CHAPITRE III

VÉRIFICATION DE LA MÉTHODE PROPOSÉE

3.1 INTRODUCTION

L'expression « métriques » est une expression courante dans la littérature en génie logiciel. Cette expression métrique est utilisée d'une manière très ambiguë. Depuis plusieurs années plusieurs métriques ont été élaborées dans le but d'avoir un meilleur contrôle des processus de production de logiciels. Dû à la prolifération de métriques, le besoin d'analyser la validité de ces métriques s'impose et plusieurs méthodes de validation ont été proposées. Des chercheurs et praticiens se voient proposer plusieurs options de validation de métriques quand le besoin de valider ou vérifier une métrique se présente.

D'une façon générale il n'y a pas eu jusqu'à date un consensus par rapport à la définition « validation ou vérification d'une métrique » et plusieurs types de validation ont été proposés, parfois distincts ou contradictoires. Il existe dans la littérature plusieurs auteurs qui affirment que plusieurs méthodes de mesure du logiciel ont peu de bases théoriques et une carence des propriétés appropriées (Kearney J.K. et al., 1986). D'autres auteurs Prater (1984) et Weyuker (1988) remarquent que les métriques de complexité logiciel ne possèdent pas les propriétés mathématiques appropriées. Cela suggère que les métriques doivent être construites avec un haut degré de rigueur théorique et mathématique.

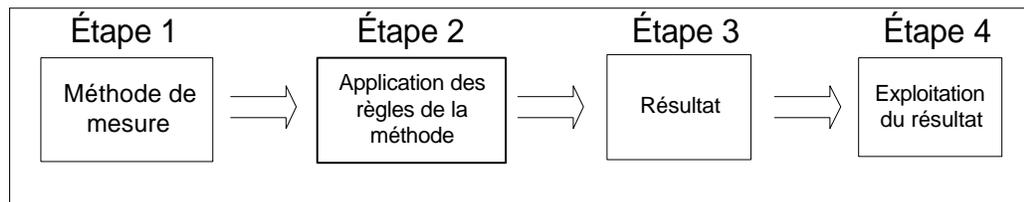
Jacquet et Abran (1996) présentent une guide de vérification des méthodes de mesures de taille fonctionnelle, afin de vérifier les différentes méthodes proposées. Dans la section suivante on

réalisera une évaluation de la méthode proposée par Abran et Desharnais (1995), pour la mesure de réutilisation fonctionnelle, par rapport au processus de mesure et le modèle détaillé du cycle de vie d'une mesure proposé par Jacquet et Abran (1996).

3.2 STRUCTURE DU GUIDE DE VALIDATION

Cette section se veut une vérification de la méthode de mesure de la réutilisation fonctionnelle proposée par Abran et Desharnais, en utilisant le guide de vérification des méthodes de mesure de taille fonctionnelle proposé par Jacquet et Abran (1996). Ce futur guide ISO a pour objet de fournir une méthode de vérification, aussi bien théorique que pratique, des méthodes de mesure de taille fonctionnelle. En plus, ce guide présente un ensemble de critères qui permettront de déterminer les points faibles et forts de chacune de ces mesures et donc de juger de leur qualité. Dans le but d'évaluer la qualité et la pertinence des différentes mesures candidates à la norme ISO, le guide doit contenir une liste de propriétés qu'une mesure « idéale » doit posséder. Dans la sous-section suivante on fera un survol des principaux concepts qui composent le guide de validation proposé par Jacquet et Abran (1996).

D'après Jacquet et Abran (1996) la structure de ce guide de validation est construite à partir des concepts sous-jacents au processus de mesure. Tout d'abord, avant de mesurer, il est nécessaire de définir une méthode de mesure. Elle sera appliquée sur le logiciel, ce qui va permettre d'obtenir des résultats, qui seront ensuite exploités dans différents modèles. La figure 3.1 permet de mieux comprendre la logique à partir de laquelle le modèle du processus de mesure est construit. Ce processus de mesure est composé de quatre étapes : La construction de la méthode de mesure, l'application de la méthode, l'obtention de résultats et l'exploitation des résultats.

**Figure 3.1**

Processus de mesure. (Tiré de Jacquet et Abran, 1997)

- Étape 1 : Afin de pouvoir mesurer, il est fondamental de construire une méthode de mesure.
- Étape 2 : Les règles qui composent la méthode de mesure doivent être appliquées sur le logiciel ou l'application logicielle à mesurer.
- Étape 3 : L'application des règles de la méthode de mesure permet d'obtenir un résultat. Ce résultat doit être analysé afin de pouvoir faire des conclusions sur son objectivité et sa valeur.
- Étape 4 : Les résultats fournis par la méthode de mesure doivent être exploités. Par exemple on peut les exploiter dans des modèles analytiques, qualitatifs ou quantitatifs.

3.2.1 Cycle de vie d'une méthode de mesure

La validation ou vérification d'une métrique ou mesure est importante tant pour les chercheurs que pour les praticiens. Le guide proposé par Jacquet et Abran (1997) qui sera utilisé dans ce travail de recherche permet de contrôler que la méthode de mesure soit bien construite, qu'elle soit applicable et qu'elle fournisse de « bons » résultats. Ce guide doit aussi nous permettre de

préciser quels sont les points à vérifier pour s'assurer que la méthode de mesure en plus d'être applicable, soit utilisée dans différents modèles.

En ce qui concerne la présente étude, la méthode proposée par Abran et Desharnais (1995) sera vérifiée à partir du cycle de vie d'une méthode de mesure telle que proposée par Jacquet et Abran (1997). Ce modèle détaillé du processus de mesure est détaillé en plusieurs sous-étapes dans le cadre de référence du processus de mesure. Les différentes sous-étapes qui composent le processus de mesure sont illustrées dans la figure 3.2.

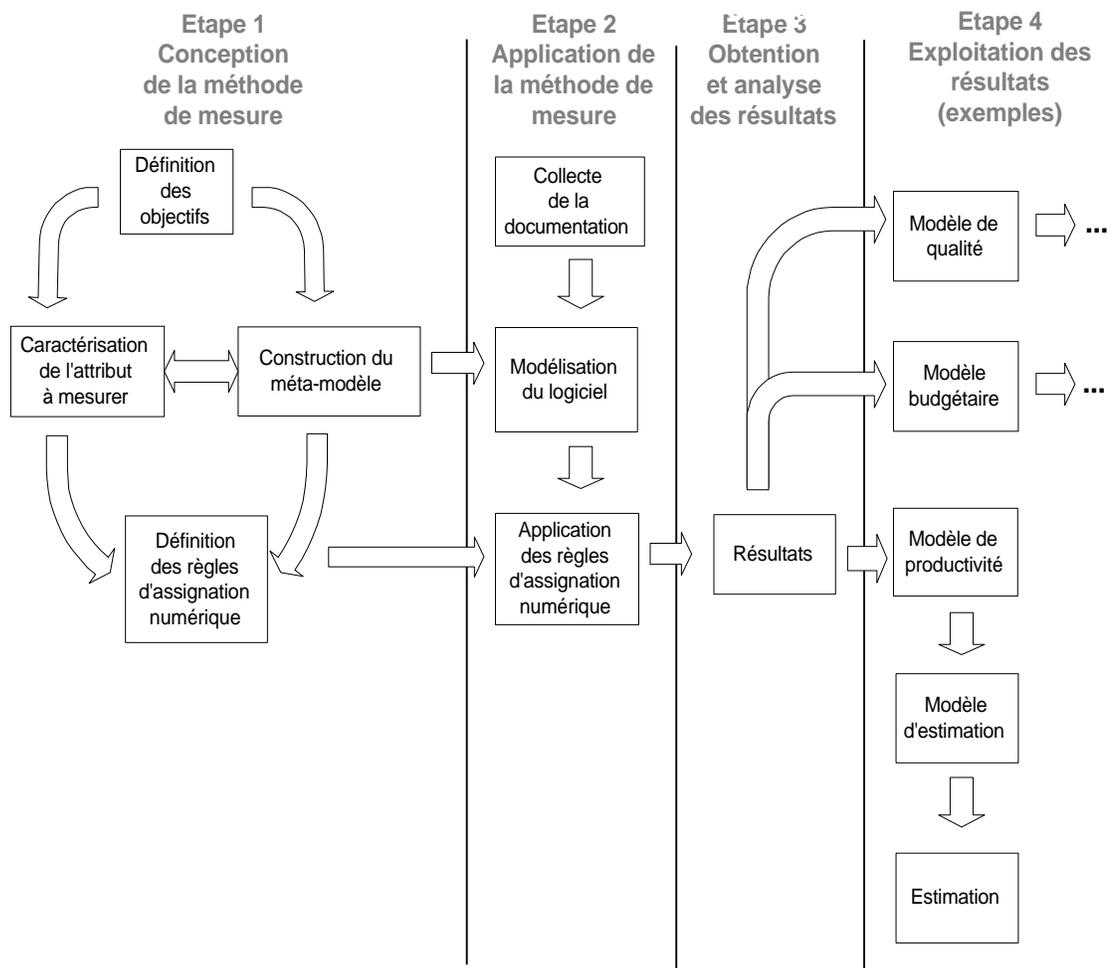


Figure 3.2

Cycle de vie d'une méthode de mesure. (Tiré de Jacquet et Abran, 1997)

Les quatre étapes du processus de mesure sont divisées en plusieurs sous-étapes (Jacquet et Abran, 1997) tel qu'il est illustré dans la figure 3.2. Ces sous-étapes sont les suivantes :

- Étape 1 : Conception de la méthode de mesure, les sous-étapes sont :
 - Sous-étape 1 : Définition des objectifs. Les objectifs de la méthode de mesure doivent être spécifiés dès le début de cette étape de la conception de la mesure.
 - Sous-étape 2 : Construction ou sélection du méta modèle. Un logiciel n'est pas un produit tangible et pour permettre sa mesure, il est souvent nécessaire de le modéliser.
 - Sous-étape 3 : Caractérisation de l'attribut mesuré. Une fois que les règles de modélisation du logiciel ont été choisies (construction du méta modèle), une définition opérationnelle (empirique) de l'attribut mesuré doit à son tour être construite.
 - Sous-étape 4 : Définition des règles d'assignation numérique. Après les sous-étapes 2 et 3 de l'étape 1, les règles permettant l'assignation de valeur(s) numérique(s) aux logiciels mesurés peuvent être construites.

- Étape 2 : Application de la méthode de mesure, les sous-étapes sont :
 - Sous-étape 1 : La documentation nécessaire à l'application de la mesure est réunie.
 - Sous-étape 2 : Le logiciel à mesurer est modélisé en utilisant les règles du concept et des attributs à mesurer.
 - Sous-étape 3 : Les règles d'assignation numérique sont appliquées.

- Étape 3 : Le résultat n'a pas de sous-étapes définies dans ce processus de mesure. Une hypothèse de travail sera soulevée dans cette section. Est-ce qu'il y a un moyen d'assurer que ce résultat soit crédible ?

- Étape 4 : Exploitation des résultats. Les différents résultats pour la méthode doivent être exploités dans l'élaboration des modèles : de qualité, budgétaire, de productivité, et d'estimation.

3.2.2 Vérification de la méthode proposée par l'entremise du guide de validation

La première phase du cycle de vie d'une méthode de mesure appelé « Mise au point de la méthode de mesure » est composée de quatre parties. La première est appelée la définition des objectifs. La méthode proposée par Abran et Desharnais (1995) se conforme à cette première partie, car elle est bâtie à partir d'une idée claire et précise du concept à mesurer du domaine dans lequel s'effectuera la mesure, ainsi que la direction que prendra l'exploitation de leurs résultats. Autrement dit, la méthode proposée permet de dégager le domaine dans lequel la méthode de mesure sera utilisée, ainsi que les objectifs à atteindre.

La deuxième partie, appelée la construction du méta modèle permet de décrire l'ensemble de caractéristiques sélectionnées pour représenter le logiciel et fournir la liste des règles qui permettront l'identification des entités utilisées lors de cette modélisation. La troisième partie doit permettre de décomposer le concept à mesurer en sous-concepts. Ceux-ci doivent à leur tour être décomposés jusqu'à ce que la méthode puisse se baser sur des observations directement mesurables. La méthode proposée par Abran et Desharnais (1995), adhère à cette partie, car elle assure que la fonctionnalité (la décomposition du concept à mesurer) sera cohérente par rapport au concept et aussi par rapport à l'objet sur lequel porte ce concept (le développement et la maintenance logiciel). En plus le modèle permet de vérifier que tous les sous-concepts et observations soient définies clairement, sans ambiguïté.

La troisième partie, celle de la définition des règles d'assignation numérique permet de déterminer quels sont les différents concepts et observations qui vont prendre part à la mesure. Plusieurs points de contrôle sont effectués à ce niveau. Dans la présente étude, et dans le modèle d'Abran et Desharnais, on utilise le guide de comptage des points de fonction, version

4.0 IFPUG (1994) dans l'élaboration de certains concepts qui servent à construire la méthode, ainsi que pour réaliser la collecte de données. Cette méthode est reconnue au niveau international et elle fera l'objet de la section suivante. Cette méthode est conforme aux divers points de contrôle définis dans cette partie du guide proposé par Jacquet et Abran (1996).

La deuxième phase du cycle de vie d'une méthode de mesure est celle de la vérification de l'étape d'application de la méthode de mesure. Elle est composée à son tour de trois parties : la récolte de la documentation, la modélisation du logiciel et l'application des règles d'assignation numérique. Dans ces trois parties, le guide est focalisé surtout sur la vérification de ces trois activités. Dans la présente étude, toutes les règles et procédures de la méthode de points de fonction ont été respectées. En plus, dans certains cas on a élaboré des règles propres qui s'adaptent à la culture de chaque organisation où la collecte de données a été réalisée. Ces règles propres suivent la philosophie de la méthode de points de fonction.

La troisième phase du cycle de vie d'une méthode de mesure est celle de la vérification des résultats. Le guide propose plusieurs contrôles qui doivent être faits sur les résultats fournis par une méthode de mesure.

Le premier est le contrôle de « non-monotonie » du résultat. Le guide demandera de vérifier que la méthode ne produit pas le même résultat quel que soit le logiciel sur lequel elle est appliquée. La méthode de mesure de la réutilisation fonctionnelle proposée, respecte la non-monotonie des résultats, car elle génère différentes classes d'équivalence selon l'application informatique mesurée (Shepperd, M. et Ince, D., 1993).

Le deuxième contrôle est la variation du résultat par rapport au contexte de la mesure. Le guide doit identifier les quatre contextes suivants. Ces contextes de travail doivent répondre dans l'ordre aux quatre questions suivantes : Est-ce que la mesure est répétable ? Est-ce que la mesure est-elle reproductible ? Pour un même contexte, est-ce que la méthode de mesure

produit le même résultat à chaque fois qu'elle est appliquée sur des éléments (des logiciels) de même taille ? Dans des contextes différents, est-ce que la méthode de mesure produit le même résultat à chaque fois qu'elle est appliquée sur des éléments (des logiciels) de même taille ? La méthode de mesure proposée par Abran et Desharnais, est elle conforme et répond elle d'une manière sans équivoque aux questions ci-dessus ?

Nous pouvons répondre aux questions ci-dessus de la manière suivante. Premièrement cette méthode renvoie toujours la même valeur quand une même application informatique est mesurée par des personnes de compétences équivalentes. Deuxièmement, l'application répétée de cette méthode sur une même application renvoie toujours les mêmes valeurs, si elle est mesurée par des personnes différentes. Ensuite, elle produit le même résultat quand elle est appliquée soit dans un même contexte, soit dans des contextes différents. En ce qui concerne le point de contrôle qui porte sur les erreurs de la mesure proposée aucune méthode de test n'est proposé dans le guide et ce contrôle n'a pas été effectué dans cette étude. Néanmoins, dans l'éventualité d'erreurs personnelles, elles ne sont pas imputables à la méthode de mesure.

Valider un résultat c'est essayer d'évaluer sa justesse en le comparant à d'autres résultats connus. Dans cette section, la procédure de vérification suivante a servi d'outils d'aide à la détection d'erreurs. Est-ce que le « compteur » de l'application dispose de données lui permettant d'évaluer le résultat de l'application d'une méthode de mesure sur un logiciel ? Dans cette étude, les résultats bruts de la méthode de mesure de la réutilisation fonctionnelle ont permis de connaître le rapport moyen entre le nombre de fichiers logiques d'une application mesurée et son nombre de points de fonction. Toujours dans le même contexte, le nombre de points de fonction moyen de plusieurs équipes de conception dans un temps déterminé n'a pas eu des écarts considérables (Morris, P.M. et Desharnais, J.-M., 1996).

La quatrième phase l'exploitation du résultat a été utilisée dans la mesure proposée par l'entremise d'un modèle de productivité. Il permet de comparer les budgets informatiques (en terme de jours personnes) engagés dans le développement d'une application informatique. En

extrapolant les différents éléments et concepts de la mesure de la réutilisation fonctionnelle proposée et le guide de validation des méthodes de mesure de taille fonctionnelle, on trouve que la mise au point de la première a été faite à partir du cycle de vie d'une méthode de mesure. Ce cycle retrace les différentes étapes et sous-étapes allant de la conception de la méthode jusqu'à son utilisation.

En conclusion la méthode de mesure de la réutilisation fonctionnelle proposée remplit les différents points de contrôle présentés dans le cycle de vie d'une méthode de mesure et son utilisation est donc recommandée, car elle respecte les différents critères d'une méthode de mesure candidate à une norme ISO.

3.3 LA MESURE UTILISÉE DANS CETTE APPROCHE

La mesure utilisée dans la méthode de mesure de la réutilisation fonctionnelle est faite à partir de deux approches qui font appel à l'analyse de points de fonction version 4.0 (IFPUG, 1994). Dans l'appendice E on présente un résumé de cette méthode de points de fonction.

La première approche est l'arrimage fait par les auteurs de la méthode avec les notions de composants réutilisables et la notion de groupe de données externes et de frontière telle que présentée par l'analyse de points de fonction version 4.0.

La deuxième approche, utilise les résultats de l'application de l'analyse de points de fonction pour la construction de bases de données historiques de points de fonction. Ces dernières servent à la vérification des résultats du modèle qui fera l'objet de ce travail. La section 4.3 et 4.4 décrivent en détail la manière précise d'élaborer ces bases de données.

Pour les projets de très petites tailles, le lecteur est prié de se référer à la technique étendue des points de fonction dans la construction des modèles de productivité en maintenance adaptative (Abran et Maya, 1995).

Pour les projets en temps réel, le lecteur doit se référer à l'étude d'adaptation ou d'arrimage de l'analyse de points de fonction à la mesure de projets développés en temps réels (St-Pierre et al., 1997).

En ce qui concerne les projets développés avec la méthode orientée objet, le lecteur devra se référer à l'étude réalisée par Fetcke, Abran, et Nguyen (1996).

Dans toutes ces études, il faudra vérifier rigoureusement les méthodes proposées avant de les appliquer dans l'industrie.

Il faut noter que la technique de points de fonction a été élaborée d'abord pour mesurer les applications du domaine M.I.S. C'est dans ce domaine où l'analyse de points de fonction s'avère jusqu'à date un outil acceptable dans l'industrie.

3.3.1 Informations requises aux mesures des points de fonction

Afin de mesurer les fonctionnalités demandées par les utilisateurs, la méthode de points de fonction est l'approche la plus répandue dans l'industrie. Dans le but de tester la méthode proposée par Abran et Desharnais on utilise plusieurs ensembles de bases de données historiques de points de fonction. Ces données sont collectées en mesurant les applications informatiques existantes dans divers projets informatiques. Dans les sections suivantes on donnera un aperçu de la procédure à suivre dans cette collecte de données.

3.3.1.1 Généralités

Les points de fonction sont des unités fonctionnelles sur la taille d'une application informatique du point de vue des usagers. Les informations requises sont en général les données (les entités internes et les interfaces) ou encore les transactions permettant de traiter ou consulter ces données (matérialisées par les écrans, les interrogations ou les rapports). Les mesures des points de fonction se basent alors sur les biens livrables aux usagers. On s'intéresse au « quoi », ou les livrables finaux, et non au « comment », ou les moyens utilisés par l'informatique pour arriver aux résultats.

3.3.1.2 Documents nécessaires pour mesurer les points de fonction

Ces informations se retrouvent généralement dans les dossiers fonctionnels du projet (diagramme de flux de données, structure de bases de données). On les retrouve en particulier dans l'analyse des items suivants :

- La liste des entités ou fichiers logiques affectés (éléments ajoutés, modifiés ou enlevés) ou consultés, le cas échéant. Les entités peuvent être remplacées par les fichiers logiques, si le modèle de données n'est pas disponible.
- Pour chaque entité/fichier dans la liste, il faut prendre en compte :
 - Le nombre des éléments affectés;
 - le nombre de relations affectées;
 - spécifier si l'entité/fichier logique est utilisé en lecture seulement ou en mode de mise à jour;
 - la liste des écrans ou rapports affectés (nouveaux, modifiés ou enlevés). Pour chaque écran ou rapport, il faut préciser :

- s'il sert à la mise à jour des données;
 - s'il est le résultat des calculs;
 - s'il est un rapport périodique;
 - s'il est une interrogation des données sans aucune mise à jour.
-
- Les caractéristiques générales du système.

Il est nécessaire de bien évaluer ces caractéristiques afin d'ajuster le total de points de fonction d'une application informatique.

CHAPITRE IV

ÉLABORATION DE DONNÉES HISTORIQUES DE POINTS DE FONCTION

4.1 CARACTÉRISTIQUES DU SITE INDUSTRIEL

Dans cette recherche deux sites industriels ont été disponibles. Ceci a permis à l'auteur de cette recherche un accès privilégié à un ensemble de données provenant d'une institution financière canadienne et d'un organisme gouvernemental de Québec. Les sections suivantes décriront les principales caractéristiques de l'environnement où les mesures ont été réalisées. On décrira aussi les caractéristiques des ensembles de données provenant d'un autre site industriel (Institution financière canadienne) et un autre provenant de l'International Software Benchmarking Standards Groups Ltd. : Worldwide Software Development. The Benchmark Release 5. March 1998 (ISBSG, 1998).

4.2 APPROCHE UTILISÉE

Cette sous-section fera un survol sur la valeur ajoutée pour une entreprise d'avoir un historique sur la taille et l'effort des projets informatiques. Elle décrit aussi les stratégies suivies dans l'élaboration des bases de données historiques de points de fonction et illustre la manière d'aller collecter l'information nécessaire à l'élaboration de bases de données historiques de points de fonction.

Une base de données historique de points de fonction sert à analyser l'évolution du développement informatique d'une entreprise. Elle permet de dégager certains éléments qui sont utilisés par les gestionnaires de projets dans leur prise de décisions.

La construction de bases de données historiques de points de fonction s'avère très utile dans les cas suivants :

- Tester plusieurs modèles analytiques, qualitatifs ou quantitatifs.
- Garder une trace de la taille des projets ainsi que l'estimation de l'effort pour chaque projet. On peut aussi comparer le coût d'un projet fini par rapport aux estimés préliminaires.
- Réaliser des « benchmarking ».

Dans la présente étude la mesure des projets servant à l'élaboration des bases de données historiques de points de fonction utilisées a été faite à partir de deux approches.

La première approche est d'aller extraire dans la documentation de chaque projet ou application informatique mesurée, les données nécessaires à sa construction (les groupes de données internes, externes, les entrées, les sorties et les interrogations). Par la suite, on calcule le pourcentage de chaque type d'entité par rapport au nombre total de points de fonction de tous les projets mesurés.

Une deuxième approche, est utilisée lorsque la documentation est incomplète ou n'existe pas : on rencontre les personnes impliquées dans le projet et on les questionne soit sur les changements fonctionnels qui seront apportées ou qui ont été faits, soit sur les fonctionnalités dont disposera un nouveau logiciel. Pour le calcul du pourcentage de chaque type d'entité la procédure est la même.

Toutes les données collectées sont emmagasinées soit dans un chiffrier électronique, soit dans un logiciel de base de données ou sur un autre support informatique qui permet la manipulation

des données. Dans cette étude on a utilisé le chiffrier électronique Microsoft Excel 97 (Windows 95). Ces données sont « stockées » par projet. Ce total de points de fonction est divisé par le nombre de points correspondants à chaque type de fonction, afin de calculer le pourcentage moyen correspondant.

On présente dans le tableau 4.1 un exemple qu'illustre la manière de garder la trace des différents projets mesurés. Ce tableau permet d'une façon claire et précise de visualiser les données correspondantes aux points de fonction de chaque type de fonction (GDE, GDI, entrées, sorties, interrogations) pour un projet donné.

Tableau 4.1

Collecte de données d'un projet ou une application informatique

Numéro du projet	25
Description du projet	Serveur bancaire
Service ou unité	Service aux entreprises
Chef du projet	No. 1997-1
Points de fonction correspondants aux GDI	30
Points de fonction correspondants aux GDE	20
Points de fonction correspondants aux entrées	50
Points de fonction correspondants aux sorties	70
Points de fonction correspondants aux interrogations	30
Nombre total de points de fonction	200

En plus le tableau 4.1 permet aussi de se faire une idée de la taille du projet et des efforts encourus dans le projet. On peut aussi faire des comparaisons avec d'autres projets similaires ou à peu près similaires d'où l'importance de toujours garder une trace des différents projets mesurés.

Dans la présente étude on se sert du nombre total de points de fonction de tous les projets mis en contribution et on calcule par la suite la moyenne du pourcentage pour chaque type de fonction par rapport au nombre total de points de fonction de tous les projets.

Par exemple, dans une institution quelconque, on a mesuré 50 projets informatiques. Du nombre total de points de fonction des 50 projets on a trouvé les pourcentages selon le type correspondant. Le total en pourcentage doit être égal à 100% du nombre total de points de fonction des 50 projets mesurés. Par ailleurs le pourcentage pour chaque type de fonction peut varier selon l'environnement de travail dans lequel le projet est élaboré, néanmoins le total doit être toujours égal à 100%. Le tableau 4.2 illustre les pourcentages par type de fonction d'une application informatique mesurée. Il illustre aussi un ensemble des données calculées en pourcentage par rapport au nombre total de points de fonction de plusieurs projets mesurés.

Tableau 4.2

Pourcentage pour chaque type de fonction

Pourcentage correspondant aux GDI	20%
Pourcentage correspondant aux GDE	10%
Pourcentage correspondant aux Entrées	30%
Pourcentage correspondant aux Sorties	25%
Pourcentage correspondant aux Interrogations	15%
Total (%)	100 %

4.3 ÉLABORATION DE BASES DE DONNÉES HISTORIQUES DE POINTS DE FONCTION

Avant qu'une mesure puisse être développée, il faut formuler une description claire de ce qui doit être mesuré, ainsi que la raison pour laquelle on veut mesurer (Kearney, 1986). Ces éléments sont aussi vrais pour la collecte de données de points de fonction, à partir des projets informatiques. Ces considérations exprimées par Kearney (1986) ont été prises en compte dans la cueillette des données utilisées dans cette étude de recherche.

Les bases de données historiques de points de fonction qui ont servi dans cette étude sont au nombre de six. Les collectes de données qui ont servi à l'élaboration des différentes bases de données historiques de points de fonction ont été réalisées dans des institutions de calibre industriel.

La première collecte de données a été réalisée dans une institution gouvernementale du Québec en 1995. La base de données historique contient des mesures sur 50 applications informatiques. Ces données ont été recueillies par monsieur Jean Marc Desharnais et l'auteur de cette présente étude.

La deuxième collecte a été réalisée au cours de la période 1993–1997 dans une institution financière canadienne. Elle comprend des mesures sur 60 projets. Cette collecte de données a été réalisée par monsieur Hong Nguyen-Kim et par l'auteur de cette présente étude.

Le tableau 4.3 présente les deux bases de points de fonctions collectées en partie par l'auteur de cette présente étude.

Tableau 4.3

Distribution fonctionnelle moyenne de points de fonction : une organisation
gouvernementale et une institution financière canadienne

Types de fonction	Ajout de fonctionnalités Institution Gouvernementale (Valencia et Desharnais) 1995 50 applications logicielles	Ajout de fonctionnalités Institution financière (Valencia et Nguyen-kim) 1993 – 1997 60 projets
GDI	20%	25%
GDE	16%	14%
ENTRÉES	13%	15%
SORTIES	46%	30%
INTERROGATIONS	6%	16%

La troisième collecte de données correspond à un ensemble de données qui comprend deux bases de données historiques de points de fonction. Ces bases de données sont disponibles dans l'étude d'Abran et Desharnais (1995) et elles contiennent des mesures sur 38 projets réalisés dans la période 1986-1990 (Voir tableau 4.4). Cet ensemble de données comprend une base de données historique de points de fonction élaborés avec des données des applications ou projets réalisés à partir de zéro et une autre base de données construite avec des données des applications ou projets où des fonctionnalités ont été ajoutées.

Le tableau 4.4 présente les deux bases historiques de points de fonctions collectées et testées dans une institution financière canadienne (Abran et Desharnais, 1995).

Tableau 4.4
Distribution fonctionnelle moyenne de points de fonction
dans une institution financière canadienne

Types de fonction	Ajout de fonctionnalités Institution financière (Abran et Desharnais) 1990 38 projets	Nouveau développement Institution financière (Abran et Desharnais) 1990 38 projets
GDI	13 %	27 %
GDE	38 %	14 %
ENTRÉES	9 %	22 %
SORTIES	31 %	21 %
INTERROGATIONS	9 %	17 %

Le quatrième ensemble des données correspond aussi à deux bases de données contenant des données qui proviennent des applications ou des projets construits à partir de zéro et des applications ou projets où des fonctionnalités ont été ajoutées. Il a été extrait du Worldwide Software Development. The Benchmark, Release 5 March 1998. International Software Benchmarking Standards Group Ltd (ISBSG, 1998) Voir tableau 4.5.

Les données nécessaires à l'élaboration de toutes les bases de données historiques de points de fonction utilisées dans cette étude de recherche ont été recueillies selon les règles et procédures de la méthode de points de fonction, version 4 (IFPUG, 1994).

Tableau 4.5

Distribution fonctionnelle moyenne de points de fonction
tirées du Worldwide Software Development The Benchmark Release 5 ISBSG

Types de fonction	Ajout de fonctionnalités ISBSG 1998 451 projets	Nouveau développement ISBSG 1998 451 projets
GDI	5 %	22.3 %
GDE	15 %	3.8 %
ENTRÉES	36 %	37.2 %
SORTIES	32 %	23.5 %
INTERROGATIONS	12 %	13.2 %

L'écart entre les divers pourcentages existant dans les bases de données historiques de points de fonction qui sont utilisées dans cette étude est dû au secteur d'activité et aussi au degré d'interrelation entre les diverses applications dans les différents sites industriels où les données ont été collectées.

4.3.1 La collecte de données

La collecte de données est une tâche très difficile dans les différents domaines scientifiques. Les difficultés principales peuvent s'exprimer par l'entremise de deux points de vue. D'une part, le modèle du processus de développement peut être inadéquat pour permettre des niveaux adéquats pour quantifier les données (Verner et Tate, 1985) et d'autre part les données que ont été obtenues peuvent se prêter à être mis en question car son degré d'acceptation est spécifique à leur utilisation dans un contexte déterminé (Yau et Collofello, 1980). Ces difficultés ont été analysées dans cette étude et on est venu à la conclusion d'utiliser certaines procédures recommandées par Basili et Phillips (1981) et Weiss et Basili (1985).

Pour remédier aux appréhensions de quantification et rencontrer les critères adéquats d'acceptation de la collecte de données, plusieurs procédures recommandées ont été suivies. En premier lieu Basili et Phillips (1981) statuent que les données doivent être obtenues à partir des environnements de développement commerciaux, plutôt qu'à partir des travaux expérimentaux.

En deuxième lieu Weiss et Basili (1985) suggèrent la méthodologie suivante pour la collecte de données dans des environnements commerciaux :

- Déterminer les objectifs de la recherche;
- développer les questions afin de satisfaire ces objectifs;
- définir le format de la collecte de données;
- établir des procédures de collecte de données;
- valider et analyser les données collectées.

Cette méthodologie a été suivie et a été appliquée grâce à la coopération des sites industriels où la collecte de données a été réalisée.

Par ailleurs, Rault (1979) argumente que les problèmes qui sont rencontrés dans le choix de données industrielles sont le résultat de deux problématiques liées l'une à l'autre. D'une part, le manque d'exactitude et du « timing » des données collectées pour la validation d'un modèle proposé et d'autre part, l'absence des modèles de validation restreint le choix des données nécessaires dans une étude déterminée. Dans cette étude le « timing » dans le choix des

données historique de points de fonction est respecté, car ces bases de données historiques s'adaptent facilement à d'autres études de recherche. En ce qui concerne le modèle de validation, cette étude se veut un cadre de vérification de la méthode de mesure de la réutilisation fonctionnelle proposée par Abran et Desharnais (1995).

En suivant les critères proposés, les collectes de données dans les organisations choisies n'ont occasionné aucun problème majeur. La méthodologie suggérée par Weiss et Basili (1985) a été utilisée. Il est à noter que :

- La documentation de divers logiciels était disponible. Les personnes en charge de cette collecte sont des experts ou étaient sous la direction d'experts en la matière. En plus ces personnes étaient aussi supportées par des analystes et programmeurs à l'emploi de ces organisations.
- Pour cette cueillette de données, on a suivi les règles de comptage des points de fonction telle que décrites dans le manuel de l'IFPUG version 4.0 (IFPUG, 1994).
- Les bases de données recueillies possèdent la taille nécessaire et elles sont représentatives pour vérifier la méthode de mesure de la réutilisation fonctionnelle proposée. En ce qui concerne la collecte de données elle n'a pas été automatisée. Jusqu'à date il n'existe aucun outil fiable pour accomplir cette tâche.
- En ce qui concerne les bases de données historiques de points de fonction tirées d'Abran et Desharnais (Tableau 4.4) et d'ISBSG (Tableau 4.5), elles possèdent aussi toutes les caractéristiques suggérées par Basili et Phillippe (1981) et par Weiss et Basili (1985).

Dans les deux institutions financières canadiennes, la mesure de points de fonction a été utilisée depuis plusieurs années. Des gestionnaires spécialisés coordonnent le programme de mesures, ainsi que la mise en place de l'environnement nécessaire pour la construction des bases de données historique de points de fonction. Dans l'institution gouvernementale, la mesure des différents projets a été réalisée pendant les mois de septembre jusqu'à décembre 1995.

Dans ces sites industriels, le responsable de l'élaboration des différentes bases de données historiques a été aussi responsable de veiller à ce que les processus et la qualité de la collecte de données respectent les différentes normes propres aux processus de collectes de mesures.

En ce qui concerne la base de données tirée de l'International Software Benchmarking Standards Group, le lecteur est prié de se référer au document officiel (ISBSG, 1998).

4.3.2 La validation de données

Les données collectées en partie par l'auteur de cette étude dans l'institution financière canadienne et celle de l'organisation gouvernementale du Québec, elles ont été validées par l'entremise de certaines procédures de validation et en deux étapes de validation.

La première étape est la méthode proposée par Morris et Desharnais (1996). Cette méthode comprend deux phases : la première phase est la validation a priori qui consiste à contrôler toutes les étapes de l'application de la mesure, de la collecte de données jusqu'à l'application des règles d'assignation numérique. Cette validation vérifie la précision et la fiabilité de la documentation ainsi que le contrôle de la compétence de l'équipe chargée d'effectuer la mesure des différents projets ou applications informatiques.

La deuxième phase est la validation a posteriori qui consiste à s'assurer de la cohérence du résultat. Afin de s'assurer d'une cohérence des différents résultats de la mesure, on peut comparer avec d'autres résultats obtenus sur des applications informatiques de même taille ou à peu près de la même taille. On peut aussi comparer les résultats des différentes mesures faites par la même équipe de programmeurs dans un contexte similaire de travail (même type de programme, même application, etc.).

La deuxième étape permet de vérifier l'analyse de la pertinence des données, la consistance de la définition de données et l'analyse de la valeur des données (Abran et Taboubi, 1994). Cette validation est nécessaire afin d'assurer la qualité et l'intégrité des données collectées.

En plus de ces procédures, on a vérifié que les « compteurs » d'une application disposent de données permettant d'évaluer le résultat de l'application d'une méthode de mesure sur un logiciel. Par exemple, le rapport moyen entre le nombre de fichiers logiques des diverses applications mesurées et leur nombre de points de fonction, est relativement stable.

Par ailleurs, tous les projets mesurés et représentés dans les bases de données historiques de points de fonction sont considérés comme à l'intérieur des normes de qualité pour ces industries.

En ce qui concerne la validation des données dans l'institution financière canadienne (Tableau 4.4) et celles d'ISBSG (tableau 4.5) le lecteur doit se référer à Abran et Taboubi (1994) et ISBSG (1998).

4.4 ANALYSE DESCRIPTIVE DE BASE DE DONNÉES

Afin de connaître les différentes sources des données utilisées dans cette étude, nous allons décrire brièvement le contexte général de développement des projets ou des applications qui sont la source des données utilisées. Cette description se fera pour tous les projets ou applications qui forment les bases de données historiques de points de fonction utilisées dans cette étude de recherche.

En premier lieu nous décrivons le contexte général de développement du projet c'est à dire la région géographique, le type d'organisation ainsi que le domaine d'affaire de chaque site industriel où la collecte de données a été réalisée. En deuxième lieu nous décrivons les caractéristiques du projet : le type de l'application et de développement. En troisième lieu, nous ferons une description de l'environnement de développement : la plate-forme de développement, le matériel et le type de programmation utilisée. En quatrième lieu, nous présenterons les méthodes de développement utilisées : la méthodologie utilisée, si le projet a été développé avec des outils CASE ou autre technique de développement. Les deux dernières sections sont propres au contrôle de chaque site industriel. Autrement dit, chaque organisation a ses propres environnements de développement ainsi que ses propres techniques ou méthodes de développement.

4.4.1 Base de données Institution gouvernementale du Québec

L'information de cette section a été extraite d'une institution gouvernementale du Québec. La base de données historique de points de fonction a été élaborée à partir de 50 applications informatiques mesurées pendant la période comprise entre les mois de septembre à décembre 1995.

4.4.1.1 Contexte de développement du projet

L'institution gouvernementale d'où provient le total des applications utilisées dans cette section fait partie des institutions gouvernementales du Québec. Le domaine de toutes les applications est celui de l'administration publique.

4.4.1.2 Caractéristiques du projet

Toutes les applications utilisées et mesurées dans cette institution gouvernementale sont du domaine M.I.S. et elles sont des applications où des ajouts fonctionnels ont été demandés.

4.4.1.3 Description de l'environnement de développement

La plupart des applications mesurées utilisaient dans une grande proportion le langage COBOL, ainsi que des bases de données DB2 et des JCL, 99% des applications ont été élaborées avec des langages de troisième génération. La plate-forme de développement utilisée dans toutes les applications est celle du « mainframe ». Il faut noter que dans cette institution il existe une grande variété de langages utilisés.

4.4.1.4 Les méthodes de développement utilisées

La méthodologie de développement utilisée a été adaptée aux besoins de l'institution. La technique de développement utilisée par la plupart des applications est celle appelée « classique ».

4.4.2 Base de données Institution Financière (Valencia et Nguyen-kim)

Dans cette institution financière canadienne, 60 projets informatiques ont été mesurés. La mesure de ces projets a été réalisée au cours des années 1993 à 1997. Ces projets représentent le portefeuille des applications informatiques propres à une institution financière ou bancaire.

4.4.2.1 Contexte de développement du projet

La totalité de projets de cette institution financière canadienne a été développée au Canada. Ces projets appartiennent au domaine bancaire et financier.

4.4.2.2 Caractéristiques des projets

Un 5% des projets étudiés et appartenant à cette institution bancaire sont des projets de la catégorie nouveau développement. Le reste (95%) sont des projets d'ajouts fonctionnels. Un 80% des projets étudiés sont du domaine MIS. Le reste appartient à la catégorie client serveur et à la personnalisation (customisation) des « packages ». Tous les projets ont été développés à l'intérieur de l'unité du service informatique de l'institution.

4.4.2.3 Description de l'environnement de développement

Les langages de programmation utilisés dans les différents projets dans cette institution financière sont très nombreux. Parmi les plus utilisés sont COBOL, Telon, CICS, DB2. Ces langages ont été utilisés dans une proportion de 80% dans les différents projets mesurés.

Les plates-formes de développement utilisées dans les projets mesurés sont celles du « mainframe » avec une proportion de 80% et un 20% des projets ont été développés avec la plate-forme « micro-ordinateur ».

4.4.2.4 Les méthodes de développement utilisées

La méthodologie utilisée est une adaptation de la technique de développement appelée « classique ». Elle a été suivie par les chefs des projets dans les différents cycles de vie logiciel.

4.4.3 Base de données institution financière (Abran et Desharnais)

Cette base de données historique de points de fonction a été élaborée à partir de 38 projets informatiques. La mesure de ces projets a été réalisée pendant la période 1986 à 1990 dans une institution financière canadienne.

4.4.3.1 Contexte de développement du projet

Le développement de ces projets a été effectué à l'intérieur d'une institution financière. Ces projets appartiennent au domaine financier et ils ont été réalisés au Canada.

4.4.3.2 Caractéristiques du projet

Le type de développement de 21 des 38 projets dans ce site industriel sont des projets d'addition et des modifications majeures sur des applications existantes de type M.I.S. et de nature transactionnelle. La plupart des projets ont été développés à l'intérieur d'une unité du service de l'informatique appartenant à ce site industriel.

4.4.3.3 Description de l'environnement de développement

Tous les projets informatiques mesurés dans ce site industriel ont presque tous été faits avec des applications du domaine M.I.S. La plate-forme de développement qui a été utilisée dans la

majorité des projets est sur la plate-forme de l'ordinateur central de grande puissance (mainframe).

4.4.3.4 Les méthodes de développement utilisées

La méthodologie de développement des projets dans ce site industriel est une adaptation interne des méthodologies structurées. Cette méthodologie a été suivie par les chefs de projets à chaque phase du cycle de vie des projets.

4.4.4 Base de données ISBSG

L'information de cette section a été tirée du Worldwide Software Development. The Benchmark Release 5 (ISBSG, 1998).

La base de données historiques de points de fonction appartenant à ISBSG (1998) a été élaborée à partir de 451 projets. Ces projets proviennent de plus de 80 organisations à travers le monde.

4.4.4.1 Contexte de développement des projets

Les 451 projets proviennent de 13 pays. Le tableau 4.6 présente les pays par ordre alphabétique, le nombre de projets par pays ainsi que le pourcentage respectif.

Les trois regroupements de pays qu'ont le plus contribué avec ISBSG (1998) sont : l'Australie avec 223 projets, l'Amérique du Nord avec 135 projets et l'Europe avec 82 projets.

Tableau 4.6

Pays d'origine, nombre de projets et le pourcentage de projets par pays d'origine
Worldwide Software Development The Benchmark Release 5 ISBSG (1998)

Pays	Projets	Pourcentage(%)
Allemagne	3	0.7
Australie	223	49.4
Autriche	2	0.4
Canada	82	18.2
Danemark	1	0.2
États Unis	53	11.8
Hong Kong	3	0.7
Indes	4	0.9
Japon	1	0.2
Norvège	4	0.9
Nouvelle Zélande	3	0.7
Pays-Bas	35	7.8
Pologne	2	0.4
Royaume Uni	35	7.8

Le type d'organisation classifie le domaine d'affaire où le projet a été développé. Du nombre total de projets, le pourcentage des projets appartenant à l'administration publique est de 18%, le domaine de finances et des services d'affaires est de 15%; pour les projets du domaine

bancaire est de 14%; le domaine manufacturier a contribué avec un 12.5% et les organisations du domaine de l'électricité, l'eau et le gaz ont contribué avec un 10% des projets.

Dans le créneau d'affaires la majorité des projets a été concentrée dans le domaine bancaire avec 22% des projets, le domaine financier avec 13% et le domaine manufacturier avec 8% des projets.

4.4.4.2 Caractéristiques des projets

Le type de développement pour les projets d'ISBSG (1998) exprimé en pourcentage a été le suivant : 64% des projets appartiennent à la catégorie nouveau développement, tandis qu'un 5% ont été du re-développement et le reste appartiennent aux projets où des ajouts fonctionnels ont été demandés.

Une majeure partie des projets 42% sont du domaine M.I.S. Pour les systèmes Transaction/Production le pourcentage est de 32% et un 12% des projets sont du domaine des systèmes d'information (Office Information Systems).

La plupart des projets (76%) ont été développés à l'intérieur d'une unité du service d'informatique appartenant à l'organisation (internal business unit). Par ailleurs seulement 22% des projets ont été conçus avec une architecture client serveur.

4.4.4.3 Description de l'environnement de développement

Les langages de programmation utilisés dans les projets appartenant à ISBSG (1998) sont au nombre de 73. Les langages plus utilisés sont au nombre de 12 : Cobol, Natural, PL/1, Telon, MS-Access, C, SQL, Cobol II, Oracle, Visual Basic, Easytrieve et Powerbuilder. Ces langages ont été utilisés dans 81% des projets.

Un pourcentage de 44% des projets ont été développés avec des langages de troisième génération (3GL), tandis qu'un 45% des projets ont utilisé les langages de quatrième génération (4GL).

La plate-forme de développement qui a été la plus utilisée dans la majorité des projets (66% des projets) est celle du « mainframe ».

4.4.4.4 Les méthodes de développement utilisées

Dans cette étude, la méthodologie de développement dans un projet est considérée seulement si elle a été suivie rigoureusement pendant le cycle de vie du logiciel.

La majorité des projets (43%) ont utilisé une méthodologie de développement adaptée à la culture de chaque organisation. Un 9% des projets n'ont pas utilisé de méthodologie.

La technique de développement utilisée dans un 44% des projets est la technique de modélisation de données appelée « classique ». Un 7% des projets ont utilisé la technique orientée objets. Le prototypage a été utilisé dans 27% des projets.

CHAPITRE V

SCÉNARIOS DE TEST

5.1 ÉLABORATION DE SCÉNARIOS DE TEST

L'objectif fondamental de cette étude est de vérifier le comportement de la méthode proposée et ses effets dans des situations autres que celles du contexte initial. Le résultat visé par cette étude est de vérifier que la méthode de mesure de la réutilisation fonctionnelle soit décrite et construite correctement en suivant le processus de mesure proposé par Jacquet et Abran (1997) et que leur résultat soit « cohérent » par rapport à l'objectif visé. Dans le cas contraire il faut identifier les situations qui font varier le comportement de la méthode de mesure proposée.

Cette étude sert à illustrer le potentiel de l'utilisation de ce modèle avec un ensemble de données industrielles (six bases de données historiques de points de fonction) et quatre scénarios de tests (chaque scénario comporte quatre simulations de tests pour un total de seize simulations de tests). Pour ce faire, nous utilisons des données représentatives d'applications informatiques dans le domaine M.I.S.

Un scénario de test est le processus qui se déroule selon un plan préétabli. Ce processus est une suite d'actions à prendre qui permet d'arriver à un résultat d'une manière ordonnée et cohérente.

Un scénario de test dans la présente étude est un contexte de travail composé de trois parties :

- La première partie est composée de toutes les coordonnées concernant les besoins de l'utilisateur. La figure 5.1 illustre ces coordonnées. Ce besoin est défini en terme de fonctionnalités qui sont exprimées en points de fonction. Dans cette partie, on définit aussi le nombre de jours personnes par point de fonction nécessaires à l'aboutissement de chaque projet. Le nombre de ressources en terme de jours personnes est défini selon les normes particulières de chaque organisation. Par exemple dans l'étude d'Abran et Desharnais (1995) le coût par point de fonction dans une nouvelle application est de deux jours personnes / PF. Pour les projets où des ajouts fonctionnels ont été effectués le coût moyen est de trois jours personnes / PF.

- | |
|--|
| <ul style="list-style-type: none"> • Ajout de 200 Points de fonction à une application existante
De 3000 PF • Coût moyen de 2 jours par point de fonction ajouté dans
Les nouvelles applications • Coût moyen de 3 jours par PF^E ajoutés dans les projets avec
Des ajouts fonctionnels • Coût total pour une application nouvelle
$200 \text{ PF}^N \times 3 \text{ jours} / \text{PF}^E = 600 \text{ jours}$ • Coût total pour une application avec des ajouts fonctionnels
$200 \text{ PF}^N \times 2 \text{ jours} / \text{PF}^N = 400 \text{ jours}$ |
|--|

Figure 5.1

Exemple qui illustre la formulation des besoins de l'utilisateur

- Deuxièmement, la méthode de mesure de la réutilisation fonctionnelle proposée par Abran et Desharnais (1995) est appliquée aux coordonnées illustrées dans la figure 5.1. Les différentes données, équations et transformations de la méthode, sont mises à contribution afin d'obtenir un résultat. Ce résultat se veut une mesure alternative à la méthode traditionnelle de points de fonction version 4.0 (IFPUG, 1994), car elle prend en compte la réutilisation des composantes fonctionnelles réutilisables. On utilise dans les transformations de la méthode les bases de données historiques de points de fonction. Une de ces bases doit correspondre aux applications nouvelles et l'autre aux applications où des fonctionnalités majeures ont été effectuées.

Le figure 5.2 illustre les trois phases fondamentales des diverses transformations de la méthode de mesure proposée. Le but de la première transformation est d'obtenir l'indicateur de réutilisation. Le résultat de la deuxième transformation est le ratio prédicteur. La troisième transformation donne comme résultat le total de points de fonctions, incluant les transactions évitées dans un projet.

<ul style="list-style-type: none"> • Transformation No. 1 $\begin{aligned} \text{PF (Indicateur)} &= \text{PF}^E \times \text{PF}_{\text{GDE}}^E(\text{ratio}) \\ &= 200 \times 38 \% \\ &= 76^E_{\text{GDE}} \end{aligned}$ • Transformation No. 2 $\begin{aligned} \text{Ratio prédicteur} &= \frac{\text{PF}_{\text{GDE}}^N + \text{PF}_{\text{ENT}}^N + \text{PF}_{\text{SOR}}^N + \text{PF}_{\text{INT}}^N}{\text{PF}_{\text{GDI}}^N} \\ &= 2.70 \end{aligned}$ • Transformation No. 3 $\begin{aligned} \text{PF}^{\text{ER}} &= \text{PF}^E + \text{Transactions évitées} \\ &= \text{PF}^E + (\text{PF}_{\text{GDE}}^E \times \text{PF}^{\text{R}}(\text{Prédicteur})) \\ &= 200 + (76 \times 2.70) \\ &= 405 \text{ PF}^{\text{ER}} \end{aligned}$
--

Figure 5.2

Exemple des transformations dans la méthode proposée
pour la mesure de la réutilisation fonctionnelle

- La troisième partie consiste à intégrer dans le tableau 5.1 les différents paramètres illustrés dans les figures 5.1 et 5.2. Ce tableau représente un modèle de productivité qui permet de comparer les résultats de l'utilisation de la mesure alternative de la réutilisation fonctionnelle.

Tableau 5.1

Exemple qui illustre l'analyse des résultats de l'application
du modèle de productivité

Concept	Type de projet	Coût unitaire	Taille en points de fonction	Coût total (jours)
Sans tenir compte de la mesure de la réutilisation (mesure traditionnelle de la méthode de points de fonction)	Nouvelle application	2 jours/P.F.	200	400
	Maintenance	3 jours/P.F.		600
En tenant compte de la mesure de la réutilisation (Mesure proposée qui prend en compte les transactions à ne pas développer)	Maintenance	1.5 jours/P.F.	405	600
	Nouvelle application	2 jours/P.F.		810

- La quatrième partie consiste à décrire l'analyse des résultats à partir du tableau qui synthétise les différentes options étudiées. D'abord les résultats de développer une application sans tenir compte de la mesure de la réutilisation (mesure traditionnelle de la méthode de points de fonction). Ensuite le développement qui tient compte de la mesure de la réutilisation fonctionnelle (mesure proposée qui prend compte du coût évité pour les transactions à ne pas développer, car elles existent déjà dans l'application). Cela est illustré à l'aide de la figure 5.3.

Le tableau 5.1 nous permet d'établir que lorsqu'on ne tient pas compte de la mesure de la réutilisation, l'addition de fonctions (maintenance) semble coûter plus cher (3 jours / PF pour la maintenance au lieu de 2 jours / PF pour Nouvelle application), mais lorsqu'on tient compte de la mesure de la réutilisation fonctionnelle, le coût unitaire par point de fonction décroît à 1.5 jours par point de fonction pour la maintenance (600 / 405 FP) par rapport à 3 jours par point de fonction quand la mesure de la réutilisation n'est pas prise en compte.

L'alternative de développer une nouvelle application qui a besoin de rédévelopper les 205 PF évités dans le projet de maintenance pourrait être d'un total de 405 PF. Si on prend le coût unitaire de 2 points de fonction pour développer une nouvelle application, cette alternative de développer 405 PF coûterait 810 jours ($405 \text{ PF} \times 2 \text{ jours / PF} = 810 \text{ jours}$). À partir de cette perspective l'exemple ci-dessus identifie un avantage économique de 210 jours-personnes (810 jours – 600 jours) pour un projet d'ajouts fonctionnels sur son alternative d'un projet à développer à partir de zéro.

Figure 5.3

Exemple de l'analyse des résultats de l'application
du modèle de productivité.

Les scénarios de test servent dans un premier temps à simuler un ensemble de circonstances liées les unes aux autres et à partir desquelles nous vérifions le résultat de l'application de la mesure de réutilisation proposée par Abran et Desharnais (1995). Le résultat de cette simulation est l'élément fondamental qui sert dans cette étude à déterminer ou à extrapoler la

pertinence de l'utilisation de la mesure de réutilisation fonctionnelle proposée (exprimée en termes de jours-personnes épargnés).

Les différents scénarios de travail sont élaborés à partir de situations qui se sont présentées dans plusieurs entreprises. Cependant afin de garder la confidentialité de ces institutions, il est préférable de ne pas les identifier.

Pour l'élaboration des différents scénarios de test (simulations), on a procédé de la manière suivante. En premier lieu, la provenance des différents scénarios de tests est celle du domaine M.I.S. Ceci exclue toutes les applications en dehors de ce domaine (temps réel, orienté-objet, etc.).

Deuxièmement, on a élaboré plusieurs scénarios de test pour évaluer la méthode de mesure. La stratégie choisie est simple et nous facilite la compréhension et la réalisation des différents tests. Cette stratégie consiste simplement à l'établissement de quatre groupes de projets qui représentent d'une manière générale les demandes globales en terme de fonctionnalités dans plusieurs entreprises. Ces quatre groupes ont été construits à partir de 16 scénarios de travail choisis qui représentaient des caractéristiques différentes.

Nous avons décidé de regrouper ces 16 scénarios de travail en 4 grands regroupements. Dans l'établissement de ce regroupement de projets, nous allons procéder de la manière suivante. Nous avons observé dans les entreprises ciblées par cette étude que la plupart des projets avaient des fonctionnalités différentes les uns par rapport aux autres. Nous avons décidé de regrouper les projets qui avaient un nombre similaire de fonctionnalités ou avec un écart autour de 10% du total de fonctionnalités ciblées. Cette manière de procéder a permis d'observer que la plupart de demandes de fonctionnalités étaient comprises entre 500 et 10 points de fonction. L'analyse de notre base de données historiques sur 110 projets a permis d'identifier quatre regroupements de projets représentatifs des besoins les plus fréquents dans

les entreprises étudiées. Il est à noter que ces projets appartiennent aux portefeuilles informatiques de grands sites industriels.

Nous avons donc retenu quatre scénarios qui sont représentatifs des diverses fonctionnalités demandées par les utilisateurs dans les organismes où les collectes de données ont été réalisées.

Les cas ont été choisis selon la taille des fonctionnalités demandées. Ils sont regroupés ainsi :

- | | |
|-----------------------------|---|
| 1. Fonctionnalités majeures | Demandes de 5000 points de fonction. |
| 2. Fonctionnalités moyennes | Demandes comprises entre 200 et 500 points de fonction. |
| 3. Fonctionnalités mineures | Demandes comprises entre 75 et 200 points de fonction |
| 4. Fonctionnalités petites | Demandes comprises entre 10 et 75 points de fonction |

5.2 APPLICATION DES SCÉNARIOS DE TEST

Les applications des différents scénarios de test seront présentées en ordre décroissant selon la taille de l'application exprimée en points de fonction. En ce qui concerne les bases de données historiques de points de fonction, elles sont utilisées dans l'ordre suivant.

Premièrement les données qui ont été collectées par l'auteur de cette étude : celles de l'organisation gouvernementale de Québec (Valencia et Desharnais) en premier. Ces données ont été collectées à partir de 50 applications informatiques. Ensuite celles collectées dans l'institution financière canadienne (Valencia et Nguyen-Kim). Dans cette institution les données ont été recueillies à partir de 60 projets informatiques.

Deuxièmement, nous utilisons les données (38 projets informatiques) utilisées par Abran et Desharnais (1995) et qui appartiennent à une autre institution financière canadienne. En troisième lieu on s'en sert de la base de données (451 projets informatiques) tirée d'ISBSG (1998). Ces bases historiques de points de fonction sont présentées dans les tableaux 4.1, 4.2 et 4.3 respectivement.

La procédure suivie est celle qui a été décrite dans l'étude d'Abran et Desharnais (1995). Cette étude est présentée dans l'appendice E. Les changements introduits se trouvent surtout au niveau de la base de données historiques de points de fonction à utiliser dans chaque scénario de test et les coordonnées concernant le nombre de jours-personnes par point de fonction.

Le premier changement est l'utilisation de la base de données de points de fonction (nouveau développement) provenant d'ISBSG (1998). Le lecteur est prié de se référer au tableau 4.5 du chapitre IV de cette étude.

En deuxième lieu, le nombre de jours-personnes à utiliser sera celui établi dans ISBSG (1998). Ces hypothèses de travail tirées d'ISBSG (1998) nous permettront de solidifier la notion de généralisation des résultats de cette étude. Nous établissons les hypothèses suivantes :

- *Le nombre de jours-personnes par point de fonction dans les projets développés à partir de zéro est en moyenne de 22.75 heures par point de fonction soit 1 point de fonction par 3.25 jours-personnes.*
- *Le nombre de jours-personnes par point de fonction dans les projets d'ajouts fonctionnels (maintenance) est de 19.58 heures par point de fonction soit 1 point de fonction par 2.80 jours-personnes.*

Les mesures tirées d'ISBSG (1998) ont servi à déterminer le nombre de jours-personnes par point de fonction. Ces mesures sont la taille fonctionnelle mesurée en points de fonction, l'effort dans un projet mesuré en heures et le temps écoulé dans le projet mesuré en mois. De cela nous dérivons le taux de livraison du projet (mesuré en heures par point de fonction) et la vitesse de livraison du projet (mesure en points de fonction par mois) tel que défini dans ISBSG (1998).

« La mesure choisie pour quantifier les extrants dans un projet de développement est le taux de livraison du projet défini comme étant l'effort en heures personnes requis pour livrer un point de fonction du logiciel à l'utilisateur final. »

Ces taux de livraison dans ISBSG (1998) sont :

- *Taux de livraison = 3.25 jours / PF*
- *Taux de livraison = 2.80 jours / PF*

Les différents résultats de l'application des différents scénarios de tests permettront d'identifier et comparer le comportement du modèle classique de productivité par rapport à un modèle avec la mesure de la réutilisation fonctionnelle proposée.

5.2.1 Simulation du scénario de test des fonctionnalités majeures

Dans cette partie de notre étude, on présente le premier scénario de test correspondant à la catégorie de fonctionnalités majeures soit l'ajout fonctionnel de 5000 points de fonction. Ce scénario de test s'applique à un logiciel de ressources humaines d'une taille approximative de 50000 points de fonction. Les annexes B, C et D illustrent les autres regroupements soit :

- Fonctionnalités moyennes. Demandes comprises entre 200 et 500 points de fonction.
- Fonctionnalités mineures. Demandes comprises entre 75 et 200 points de fonction
- Fonctionnalités petites. Demandes comprises entre 10 et 75 points de fonction.

L'hypothèse à vérifier dans la simulation de la méthode de mesure de la réutilisation fonctionnelle est la suivante : Est-ce que l'application de la méthode de mesure de la réutilisation fonctionnelle proposée par Abran et Desharnais (1995) permet de dégager des bénéfices budgétaires (exprimés en terme de jours-personnes) quel que soit le scénario de test utilisé et la base de données historiques de points de fonction utilisée ? Cette hypothèse de

travail est la même pour toutes les simulations des différents regroupements des scénarios de test réalisé dans cette étude.

La procédure à utiliser dans tous les différents scénarios de tests est la suivante.

1. Nous décrivons d'abord les besoins de l'utilisateur exprimés en points de fonction (PF).
2. Par la suite, on détermine le nombre de jours-personnes pour les applications informatiques. Le nombre de jours-personnes est le même pour tous les scénarios de test, soit 3.25 jours-personnes pour un projet développé à partir de zéro, soit 2.80 jours-personnes pour un projet d'ajout de fonctionnalités (maintenance).
3. Troisièmement, on fait les calculs de toutes les équations de la méthode de mesure de la réutilisation fonctionnelle.
4. Finalement et à l'aide d'un tableau nous analysons et comparons les résultats en fonction de son coût et selon l'option prise, soit utiliser ou ne pas utiliser la méthode de mesure de la réutilisation fonctionnelle.

5.2.1.1 Scénario qui utilise les données de l'institution gouvernementale de Québec (Valencia et Desharnais)

Dans ce scénario de test, nous voulons expérimenter le comportement de la méthode de mesure de la réutilisation fonctionnelle avec la base de données historique de points de fonction construite à partir des projets informatiques de l'institution gouvernementale de Québec. Dans ce scénario de test, l'utilisateur demande des ajouts fonctionnels de 5000 points de fonction à une application existante de 50000 points de fonction.

Regroupement des projets de 5000 points de fonction

- Base de données historiques de points de fonction à utiliser : Institution gouvernementale
- Besoin de l'utilisateur : Ajout de 5000 PF à une application de 50000 PF
- Durée en jours-personnes : 3.25 jours par PF dans les nouvelles applications
2.80 jours par PF dans les projets de maintenance

- Coût pour les nouvelles applications : $5000 \text{ PF} \times 3.25 \text{ jours} / \text{PF} = 16250 \text{ jours}$
- Coût pour les applications de maintenance : $5000 \text{ PF} \times 2.80 \text{ jours} / \text{PF} = 14000 \text{ jours}$
- Application de la méthode de mesure de la réutilisation fonctionnelle

$$\begin{aligned} \text{PF (Indicateur)} &= \text{PF}^E \times \text{PF}_{\text{GDE}}^E(\text{ratio}) \\ &= 5000 \times 16 \% \\ &= 800^E_{\text{GDE}} \end{aligned}$$

$$\begin{aligned} \text{PF}^R(\text{Ratio prédicteur}) &= \text{PF}_{\text{GDE}}^N + \text{PF}_{\text{ENT}}^N + \text{PF}_{\text{SOR}}^N + \text{PF}_{\text{INT}}^N / \text{PF}_{\text{GDI}}^N \\ &= 4.05 \end{aligned}$$

$$\begin{aligned} \text{PF}^{\text{ER}} &= \text{PF}^E + \text{Transactions évitées} \\ &= \text{PF}^E + (\text{PF}_{\text{GDE}}^E \times \text{PF}^R(\text{Prédicteur})) \\ &= 5000 + (800 \times 4.05) \\ &= 8240 \text{ PF}^{\text{ER}} \end{aligned}$$

Le tableau 5.2 permet d'analyser la méthode de mesure de la réutilisation fonctionnelle en utilisant les données historiques de points de fonction qui appartiennent à l'institution gouvernementale de Québec. Cette analyse est réalisée par l'entremise d'un modèle de productivité et elle identifie deux points principaux :

- Lorsqu'on ne tient pas compte de la réutilisation, l'addition de fonctions (maintenance) coûte légèrement moins cher (2.8 jours par PF pour la maintenance au lieu de 3.25 jours par PF pour une nouvelle application), et lorsqu'on tient compte de la mesure de la réutilisation fonctionnelle, la différence de coût unitaire 1.7 jours par PF pour la maintenance (14000 jours / 8240 jours) est maintenant beaucoup plus grande et toujours par rapport à 3.25 jours par PF pour nouvelle application.
- L'alternative de développer une nouvelle application qui n'a pas besoin de ré développer les 3240 PF évités dans le projet de maintenance pourrait être d'un total de 8240 PF. Si on prend le coût unitaire de 3.25 points de fonction pour développer une nouvelle application, cette alternative de développer 8240 PF coûterait 26780 jours ($8240 \text{ PF} \times 3.25 \text{ jours}/\text{PF} = 26780 \text{ jours}$). À partir de cette perspective l'exemple ci-dessus identifie un avantage économique de 12780 jours personnes ($26780 \text{ jours} - 14000 \text{ jours}$) pour un projet d'ajouts fonctionnels sur son alternative un projet à développer à partir de zéro.

Tableau 5.2
Analyse des résultats

Concept	Type de projet	Coût unitaire	Taille en points de fonction	Coût total (jours)
Sans tenir compte de la mesure de la réutilisation (Modèle traditionnel avec la méthode de points de fonction)	Nouvelle application	3.25 jours/P.F.	5000	16250
	Maintenance	2.80 jours/P.F.		14000
En tenant compte de la mesure de la réutilisation (Mesure proposée qui prend en compte les transactions à ne pas développer)	Maintenance	1.7 Jours/P.F.	8240	14000
	Nouvelle application	3.25 jours/P.F.		26780

5.2.1.2 Scénario qui utilise les données de l'institution financière canadienne (Valencia et Nguyen-Kim)

- Base de données historiques de points de fonction à utiliser : Institution financière canadienne (Valencia et Nguyen-Kim)

- Besoin de l'utilisateur : Ajout de 5000 PF à une application de 50000 PF
- Durée en jours personnes : 3.25 jours par PF dans les nouvelles applications
2.80 jours par PF dans les projets de maintenance
- Coût pour les nouvelles applications : 5000 PF x 3.25 jours / PF = 16250 jours
- Coût pour les applications de maintenance : 5000 PF x 2.80 jours / PF = 14000 jours
- Application de la méthode de mesure de la réutilisation fonctionnelle

$$\begin{aligned} \text{PF (Indicateur)} &= \text{PF}^E \times \text{PF}_{\text{GDE}}^E(\text{ratio}) \\ &= 5000 \times 14 \% \\ &= 700^E_{\text{GDE}} \end{aligned}$$

$$\begin{aligned} \text{PF}^R (\text{Ratio prédicteur}) &= \text{PF}_{\text{GDE}}^N + \text{PF}_{\text{ENT}}^N + \text{PF}_{\text{SOR}}^N + \text{PF}_{\text{INT}}^N / \text{PF}_{\text{GDI}}^N \\ &= 3 \end{aligned}$$

$$\begin{aligned} \text{PF}^{\text{ER}} &= \text{PF}^E + \text{Transactions évitées} \\ &= \text{PF}^E + (\text{PF}_{\text{GDE}}^E \times \text{PF}^R(\text{Prédicteur})) \\ &= 5000 + (700 \times 3) \\ &= 7100 \text{ PF}^{\text{ER}} \end{aligned}$$

Le tableau 5.3 permet d'analyser l'utilisation de la méthode de mesure de la réutilisation fonctionnelle en utilisant les données historiques de points de fonction qui appartiennent à l'institution financière canadienne (Valencia et Nguyen-Kim). Cette analyse est réalisée par l'entremise d'un modèle de productivité et elle identifie deux points principaux :

- Lorsqu'on ne tient pas compte de la réutilisation, l'addition de fonctions (maintenance) semble coûter moins cher (2.8 jours par PF pour la maintenance au lieu de 3.25 jours par PF pour une nouvelle application), et lorsqu'on tient compte de la mesure de la réutilisation fonctionnelle, la différence de coût unitaire 1.97 jours par PF pour la maintenance (14000 / 7100) est maintenant plus grande et toujours par rapport à 3.25 jours par PF pour une nouvelle application.
- L'alternative de développer une nouvelle application qui n'a pas besoin de ré développer les 2100 PF évités dans le projet de maintenance pourrait être d'un total de 7100 PF. Si on prend le coût unitaire de 3.25 points de fonction pour développer une nouvelle application, cette alternative de développer 7100 PF coûterait 23075 jours (7100 PF x 3.25 jours/PF =

23075 jours). À partir de cette perspective l'exemple ci-dessus identifie un avantage économique de 9075 jours personnes (23075 jours – 14000 jours) pour un projet d'ajouts fonctionnels sur son alternative un projet à développer à partir de zéro.

Tableau 5.3

Analyse des résultats

Concept	Type de projet	Coût unitaire	Taille en points de fonction	Coût total (jours)
Sans tenir compte de la mesure de la réutilisation (mesure traditionnelle de la méthode de points de fonction)	Nouvelle application	3.25 jours/P.F.	5000	16250
	Maintenance	2.80 jours/P.F.		14000
En tenant compte de la mesure de la réutilisation (Mesure proposée qui prend en compte les transactions à ne pas développer)	Maintenance	1.97 jours/P.F.	7100	14000
	Nouvelle application	3.25 jours/P.F.		23075

5.2.1.3 Scénario qui utilise les données de l'institution financière canadienne (Abran)

- Base de données historiques de points de fonction à utiliser : Institution financière canadienne (Abran)
- Besoin de l'utilisateur : Ajout de 5000 PF à une application de 50000 PF
- Durée en jours personnes : 3.25 jours par PF dans les nouvelles applications
2.80 jours par PF dans les projets de maintenance
- Coût pour les nouvelles applications : 5000 PF x 3.25 jours / PF = 16250 jours
- Coût pour les applications de maintenance : 5000 PF x 2.80 jours / PF = 14000 jours
- Application de la méthode de mesure de la réutilisation fonctionnelle

$$\begin{aligned} \text{PF}^{\text{E}} (\text{Indicateur}) &= \text{PF}^{\text{E}} \times \text{PF}_{\text{GDE}}^{\text{E}} (\text{ratio}) \\ &= 5000 \times 38 \% \\ &= 1900^{\text{E}}_{\text{GDE}} \end{aligned}$$

$$\begin{aligned} \text{PF}^{\text{R}} (\text{Ratio prédicteur}) &= \text{PF}_{\text{GDE}}^{\text{N}} + \text{PF}_{\text{ENT}}^{\text{N}} + \text{PF}_{\text{SOR}}^{\text{N}} + \text{PF}_{\text{INT}}^{\text{N}} / \text{PF}_{\text{GDI}}^{\text{N}} \\ &= 2.74 \end{aligned}$$

$$\begin{aligned} \text{PF}^{\text{ER}} &= \text{PF}^{\text{E}} + \text{Transactions évitées} \\ &= \text{PF}^{\text{E}} + (\text{PF}_{\text{GDE}}^{\text{E}} \times \text{PF}^{\text{R}} (\text{Prédicteur})) \\ &= 5000 + (1900 \times 2.74) \\ &= 10206 \text{ PF}^{\text{ER}} \end{aligned}$$

Le tableau 5.4 permet d'analyser l'utilisation de la méthode de mesure de la réutilisation fonctionnelle en utilisant les données historiques de points de fonction qui appartiennent à l'institution financière canadienne (Abran). Cette analyse est réalisée par l'entremise d'un modèle de productivité et elle identifie deux points principaux :

- Lorsqu'on ne tient pas compte de la réutilisation, l'addition de fonctions (maintenance) semble coûter moins cher (2.8 jours par PF pour la maintenance au lieu de 3.25 jours par PF pour une nouvelle application), et lorsqu'on tient compte de la mesure de la réutilisation fonctionnelle, la différence de coût unitaire 1.37 jours par PF pour la maintenance (14000 / 10206) est maintenant plus grande et toujours par rapport à 3.25 jours par PF pour une nouvelle application.

- L'alternative de développer une nouvelle application qui n'a pas besoin de ré développer les 5206 PF évités dans le projet de maintenance pourrait être d'un total de 10206 PF. Si on prend le coût unitaire de 3.25 points de fonction pour développer une nouvelle application, cette alternative de développer 10206 PF coûterait 33169 jours ($10206 \text{ PF} \times 3.25 \text{ jours/PF} = 33169 \text{ jours}$). À partir de cette perspective l'exemple ci-dessus identifie un avantage économique de 19169 jours personnes ($33169 \text{ jours} - 14000 \text{ jours}$) pour un projet d'ajouts fonctionnels sur son alternative un projet à développer à partir de zéro.

Tableau 5.4
Analyse des résultats

Concept	Type de projet	Coût unitaire	Taille en points de fonction	Coût total (jours)
Sans tenir compte de la mesure de la réutilisation (mesure traditionnelle de la méthode de points de fonction)	Nouvelle application	3.25 jours/P.F.	5000	16250
	Maintenance	2.80 jours/P.F.		14000
En tenant compte de la mesure de la réutilisation (Mesure proposée qui prend en compte les transactions à ne pas développer)	Maintenance	1.37 jours/P.F.	10206	14000
	Nouvelle application	3.25 jours/P.F.		33169

5.2.1.4 Scénario qui utilise les données tirées d'ISBSG (1998)

Regroupement des projets de 5000 points de fonction

- Base de données historiques de points de fonction à utiliser : ISBSG (1998)r
- Besoin de l'utilisateur : Ajout de 5000 PF à une application de 50000 PF
- Durée en jours personnes : 3.25 jours par PF dans les nouvelles applications
2.80 jours par PF dans les projets de maintenance
- Coût pour les nouvelles applications : 5000 PF x 3.25 jours / PF = 16250 jours
- Coût pour les applications de maintenance : 5000 PF x 2.80 jours / PF = 14000 jours
- Application de la méthode de mesure de la réutilisation fonctionnelle

$$\begin{aligned} \text{PF (Indicateur)} &= \text{PF}^E \times \text{PF}_{\text{GDE}}^E(\text{ratio}) \\ &= 5000 \times 15 \% \\ &= 750_{\text{GDE}}^E \end{aligned}$$

$$\begin{aligned} \text{Ratio prédicteur} &= \text{PF}_{\text{GDE}}^N + \text{PF}_{\text{ENT}}^N + \text{PF}_{\text{SOR}}^N + \text{PF}_{\text{INT}}^N / \text{PF}_{\text{GDI}}^N \\ &= 3.48 \end{aligned}$$

$$\begin{aligned} \text{PF}^{\text{ER}} &= \text{PF}^E + \text{Transactions évitées} \\ &= \text{PF}^E + (\text{PF}_{\text{GDE}}^E \times \text{PF}^{\text{R}}(\text{Prédicteur})) \\ &= 5000 + (750 \times 3.48) \\ &= 7610 \text{ PF}_{\text{S}}^{\text{ER}} \end{aligned}$$

Le tableau 5.5 permet d'analyser l'utilisation de la méthode de mesure de la réutilisation fonctionnelle en utilisant les données historiques de points de fonction qui appartiennent à ISBSG (1998). Cette analyse est réalisée par l'entremise d'un modèle de productivité et elle identifie deux points fondamentaux :

- Lorsqu'on ne tient pas compte de la réutilisation, l'addition de fonctions (maintenance) semble coûter moins cher (2.8 jours par PF pour la maintenance au lieu de 3.25 jours par PF pour une nouvelle application), et lorsqu'on tient compte de la mesure de la réutilisation

fonctionnelle, la différence de coût unitaire 1.83 jours par PF pour la maintenance (14000 jours / 7610 jours) est maintenant plus grande et toujours par rapport à 3.25 jours par PF pour une nouvelle application.

- L'alternative de développer une nouvelle application qui n'a pas besoin de rédévelopper les 2610 PF évités dans le projet de maintenance pourrait être d'un total de 7610 PF. Si on prend le coût unitaire de 3.25 points de fonction pour développer une nouvelle application, cette alternative de développer 7610 PF coûterait 24732 jours (7610 PF x 3.25 jours/PF = 24732 jours). À partir de cette perspective l'exemple ci-dessus identifie un avantage économique de 10732 jours personnes (24732 jours – 14000 jours) pour un projet d'ajouts fonctionnels sur son alternative un projet à développer à partir de zéro.

Tableau 5.5

Analyse des résultats

Concept	Type de projet	Coût unitaire	Taille en points de fonction	Coût total (jours)
Sans tenir compte de la mesure de la réutilisation (mesure traditionnelle de la méthode de points de fonction)	Nouvelle application	3.25 jours/P.F.	5000	16250
	Maintenance	2.80 jours/P.F.		14000
En tenant compte de la mesure de la réutilisation (Mesure proposée qui prend en compte les transactions à ne pas développer)	Maintenance	1.83 jours/P.F.	7610	14000
	Nouvelle application	3.25 jours/P.F.		24732

L'appendice B montre les scénarios de tests utilisés sur les fonctionnalités moyennes, tandis que l'appendice C présente les scénarios utilisés sur les fonctionnalités mineures. Les scénarios de tests sur les fonctionnalités petites sont présentés dans l'appendice D.

5.2.2 Impact de la méthode de mesure proposée dans le coût unitaire

L'objectif de cette section est d'examiner les différents résultats expérimentaux de l'impact de la méthode de mesure proposée sur le calcul du coût unitaire dans les projets informatiques où des ajouts fonctionnels ont été réalisés. Pour identifier l'influence de la méthode de mesure proposée sur le calcul du coût unitaire, nous prenons comme hypothèse que le coût unitaire dans un projet logiciel est comme suit :

- Pour le développement à partir de zéro le coût unitaire est de 3.25 jours personnes par point de fonction.
- Pour la maintenance (ajouts fonctionnels) le coût unitaire est de 2.80 jours personnes par point de fonction. Ces données sont tirées d'ISBSG (1998).

Le tableau 5.6 résume les résultats des différentes transformations de la méthode de mesure de réutilisation, ainsi que l'impact du coût unitaire pour les différents scénarios de tests appartenant au regroupement des projets de 5000 points de fonction. Nous avons identifié une variation toujours au-dessous de la constante de 2.80 j-p/PF du coût unitaire illustré dans la colonne impact dans le calcul du coût unitaire (Maintenance). Les différences entre les différents résultats sont dues à la variation des pourcentages des différents types de transactions qui font partie des différentes bases de données historiques de points de fonction utilisées. Ce tableau présente dans un premier lieu les quatre scénarios de test utilisés. Dans la deuxième colonne, on présente le nombre de points de fonction de l'indicateur (nombre de GDE à réutiliser). La troisième colonne illustre le ratio prédicteur, tandis que la quatrième colonne illustre le nombre de transactions (points de fonction), et qui a été calculé à partir du ratio prédicteur. La dernière colonne illustre l'impact de l'utilisation de la méthode de mesure

de la réutilisation fonctionnelle proposée dans le coût unitaire exprimé en terme de jours personne par point de fonction.

Tableau 5.6

Sommaire des résultats des différentes transformations de la méthode de mesure de réutilisation pour les différents scénarios de tests appartenant au regroupement des fonctionnalités majeures

Scénarios Selon la base de données utilisée	PF (Indicateur)	PF (Prédicteur)	Transactions évitées (PF)	Impact dans le calcul du coût unitaire	
				Nouvelle Applicat.	Mainte nance
Institution gouvernementale de Québec	800	4.05	3240	3.25 jp/PF	1.70jp/PF
Institution financière (Valencia et	700	3	2100	3.25 jp/PF	1.97jp/PF

Nguyen-Kim)					
Institution financière (Abran)	1900	2.74	5206	3.25 jp/PF	1.37jp/PF
ISBSG (1998)	750	3.48	2610	3.25 jp/PF	1.83jp/PF

Constantes	
Coût unitaire	
Développement	Maintenance
3.25 j.p. / P.F.	2.80 j.p. / P.F.

Les appendices B, C et D illustrent dans les tableaux B.5, C.5 et D.5 le sommaire des résultats des différentes transformations de la méthode de mesure de réutilisation, ainsi que l'impact du coût unitaire pour les différents scénarios de tests appartenant au regroupement des projets de fonctionnalités moyennes, mineures et petites respectivement.

5.3 CONCLUSION SUR LES SCÉNARIOS DE TEST UTILISÉS

Dans les différents scénarios de test utilisés dans cette étude, nous arrivons aux conclusions suivantes :

- L'utilisation de la mesure de la réutilisation fonctionnelle permet d'identifier les bénéfices tangibles en terme du nombre de jours personnes qui ont été évités ou épargnés. Ceci est vrai pour tous les différents scénarios de tests utilisés dans cette étude.
- Cette mesure de la réutilisation proposée identifie dans tous les scénarios de tests utilisés dans cette étude une variation du calcul du coût unitaire dans les applications où des ajouts fonctionnels ont été réalisés.

- Il existe certains écarts concernant les divers résultats selon les bases de données historiques de point de fonction. Cela est dû aux différents types d'applications qui ont servi à élaborer les base de données historiques.

CHAPITRE VI

INTERPRÉTATION DE RÉSULTATS

6.1 CONTEXTE D'INTERPRÉTATION

Ce chapitre explique la démarche que nous avons entreprise pour évaluer les différentes procédures développées lors de notre test expérimental contrôlé. À cet effet, la section 6.2 expose les principales recommandations de McDonell (1991) qui ont été prises en compte dans l'analyse des résultats. La section 6.3 analyse l'extrapolation des résultats. La section 6.4 nous présente les bénéfices de l'utilisation de la méthode de mesure. Finalement, les conclusions et des suggestions de recherche sont présentées.

6.2 APPROCHE D'ANALYSE EMPIRIQUE

L'approche de cette étude pour l'analyse des résultats de l'application de la méthode de mesure proposée, tient compte des treize principales recommandations des données observationnelles tels que décrits par McDonell (1991), ainsi que sur les recommandations de Ramsey et Basili (1989). Ces recommandations sont les suivantes :

- Analyse de la méthode de mesure proposée

Afin de pouvoir analyser la méthode de mesure proposée, plusieurs comparaisons concernant la mesure de la taille fonctionnelle ont été réalisées. Dans cette étude, on a en outre vérifié la conformité de la méthode proposée par rapport au guide de vérification des méthodes de mesure de taille fonctionnelle (Jacquet et Abran, 1996). Par la suite, on a utilisé la méthode de

points de fonction version 4.0 (IFPUG, 1994), dans la collecte de données servant à l'élaboration de bases historiques de points de fonction. Dans la validation de la collecte de données on a utilisé des procédures de validation postérieures du comptage de points de fonction (Morris et Desharnais, 1993). Pour encadrer les différentes étapes de cette étude on a mis en contribution les recommandations de conception expérimentale de MacDonell (1991), ainsi que les recommandations de Ramsey et Basili (1989). Le cadre de recherche utilisé dans cette étude est le cadre de Basili (Basili, Selby et Hutchens, 1986), qui est décrit sommairement dans le chapitre I.

- Évaluation subjective

Dans cette étude, aucune évaluation subjective concernant des paramètres, des coefficients ou d'autres types de paramètres n'a été effectuée. Le type d'échelle utilisée est celui qui est utilisé dans la méthode de points de fonction (IFPUG version 4.0, 1994).

- La collecte de données

Le processus de collecte de données a été décrit dans le chapitre III. Le personnel responsable de ces mesures a été composé de spécialistes dans le domaine. Les données en plus ont été vérifiées par les chefs de projets, ainsi que par les directeurs de secteurs et de service.

- La taille des applications mesurées

Dans l'élaboration de deux bases de données historiques de points de fonction, les données ont été homogènes. Chaque base de données historique de points de fonction élaborée dans cette étude représente un certain ordre de grandeur similaire pour les différentes applications mesurées. Ceci est un avantage à notre étude, car cela facilite la généralisation de l'extrapolation des résultats.

- La taille de l'échantillon

La taille d'un échantillon doit être suffisamment grande pour permettre des tests qui soient significatifs (Lister, 1982). D'après Henry et Kafura (1981) un test doit posséder un niveau élevé, bien documenté, réel, large et réutilisable. Idéalement, un échantillon fiable et large est nécessaire pour la validation et la vérification dans une recherche. Dans cette étude, la taille de l'échantillon disponible est d'environ 599 observations. Ces observations proviennent de quatre organisations avec des profils différents. Un total de 98 observations appartiennent à deux institutions financières canadiennes. Dans l'organisation gouvernementale du Québec on a 50 observations. Les 451 observations restantes ont été tirées de The Benchmark Release 5 (ISBSG, 1998).

- Les langages d'application et leur domaine d'application

Idéalement, une recherche doit être validée à partir d'une plage large de langages de programmation. Cette étude en particulier dérive ses données à partir des applications du domaine M.I.S.

- Les sujets de l'expérience

Dans cette expérience, les données proviennent des applications réelles du domaine M.I.S. Ces applications ont été développées par du personnel chevronné dans la matière. D'après Halstead (1977) et Gordon (1979), les programmes écrits par du personnel novice (junior) ont tendance à avoir certains « défauts ». Pour cette raison, Shepherd et Lassez (1980) suggèrent que les chercheurs qui utilisent des données créées par du personnel novice doivent prendre des précautions dans la généralisation de leurs conclusions.

- Les facteurs incontrôlables

Davis et LeBlanc (1998) et Woodfield et al. (1981), suggèrent qu'une des étapes initiales dans une recherche est de faire connaître les facteurs incontrôlables dans un projet, afin de limiter la portée du projet et ainsi cimenter sa crédibilité. Les facteurs incontrôlables dans cette étude ont été éliminés dans cette recherche, car on a utilisé des applications issues des environnements homogènes. On a aussi éliminé certaines applications et projets informatiques qui n'étaient

conformes aux divers critères de sélection (énumérés dans ce travail de recherche). La construction des bases de données historiques de points de fonction, ainsi que l'élaboration de différents scénarios de test sont conformes aux critères choisis dans cette étude.

- La validité statistique de la relation analysée

Dans la présente étude, nous n'avons pas utilisé des tests statistiques, car l'atteinte des objectifs de cette recherche peut se faire par l'entremise d'autres procédures.

- L'interprétation des résultats

L'interprétation des résultats respecte les contraintes ou limites de la présente étude.

- La publication de données et les hypothèses

Toutes les données et les hypothèses de travail sont présentées dans cette étude.

6.3 EXTRAPOLATION DES RÉSULTATS

Au début de ce travail, nous nous sommes fixé comme objectif de vérifier la méthode de mesure de la réutilisation fonctionnelle proposée par Abran et Desharnais (1995). Pour cela nous avons proposé une vérification théorique et une autre empirique. La démarche de la vérification théorique permet de comparer la conformité de cette méthode par rapport au processus de mesure et au cycle de vie d'une mesure tel que proposée par Jacquet et Abran (1997). La vérification empirique est faite en comparant les divers résultats de l'application de la méthode de mesure proposée à une série de scénarios de test. Ces scénarios ont été réalisés à partir des contextes réels de travail dans l'industrie.

La vérification théorique de la méthode de mesure de la réutilisation fonctionnelle (Abran et Desharnais, 1995) telle que décrite dans le chapitre III, permet d'atteindre un des objectifs visés dans la démarche de vérification théorique. Cette démarche a fait l'objet d'une comparaison critique qui a permis de vérifier que la méthode de mesure de la réutilisation fonctionnelle proposée adhère et s'adapte facilement aux différentes règles et procédures du processus et du cycle de vie d'une mesure proposée par Jacquet et Abran (1997). La vérification empirique a permis de développer un contexte global cohérent de vérification à partir d'une série de scénarios de tests tels que décrits dans le chapitre V. Cette vérification a permis aussi de développer, de regrouper et d'intégrer d'une manière cohérente les différents scénarios de tests utilisés.

Les résultats de ce type d'analyse permettent de confirmer que, pour ces conditions d'analyse, le modèle de mesure de la réutilisation fonctionnelle, possède un degré de fiabilité. Ceci permet d'affirmer que son utilisation dans des situations similaires produira les mêmes résultats, avec un degré de variabilité négligeable. La mise en garde suivante est pertinente à ce stade ci de ce travail. Le regroupement des divers scénarios de tests a été fait dans l'objectif de synthétiser la vérification du modèle, néanmoins nous pourrions créer des sous regroupements à l'intérieur de chaque regroupement.

Les questions qui ont été soulevées par la démarche suivie dans le regroupement des scénarios de test peuvent faire l'objet de recherches subséquentes. Cependant nous croyons que les écarts dans les résultats quel que soit le scénario de test choisi, permet d'éclaircir et d'apporter des adaptations à l'estimation d'un projet informatique, de contrôler le progrès dans le projet et d'améliorer la prise de décision.

6.4 CONCLUSION

L'objectif de recherche était de vérifier l'hypothèse suivante : Est-ce que la méthode de mesure de la réutilisation fonctionnelle qui a été développée dans un seul contexte expérimental (Abran et Desharnais, 1995) peut être généralisée à d'autres contextes ?

Pour la vérification de cette hypothèse de recherche, trois difficultés majeures ont été rencontrées.

- La première difficulté résidait dans l'accès à des données industrielles. Cette difficulté a été surpassée grâce à l'accès aux données industrielles disponibles dans l'étude d'Abran et Desharnais (1995), et à l'accès aux données industrielles qui ont été collectées en partie par l'auteur de cette recherche et celles tirées d'ISBSG (1998). La base de données disponible dans l'étude d'Abran et Desharnais (1995), contenait des mesures sur 38 projets réalisés dans la période 1986-1990. Une des bases de données collectées en partie par l'auteur de cette étude, contenait des mesures sur 50 applications; cette mesure de ces données a été réalisée entre les mois de septembre et décembre 1995. L'autre base de données collectée en partie aussi par l'auteur de cette recherche contenait des mesures sur 60 projets; cette mesure a été effectuée au cours des périodes 1993-1997. Les bases de données tirées d'ISBSG contenaient des mesures sur 451 projets.
- La deuxième difficulté résidait dans le choix d'un cadre de recherche capable de bien évaluer, comprendre, contrôler et améliorer les différentes itérations du processus de validation de l'hypothèse de travail. L'approche de recherche sélectionnée a été celle de Basili (Basili et al., 1986) qui est présentée sommairement dans le chapitre I de cette recherche.
- La troisième difficulté a été celle de l'élaboration de différents scénarios de tests. Afin de garder un degré élevé de fiabilité, les différents scénarios de tests ont été réalisés à partir de contextes industriels. Cependant, la difficulté s'est posée dans le choix d'un échantillon (pilote) représentatif de diverses complexités fonctionnelles. Nous avons décidé de regrouper les différents scénarios de tests en 4 groupes, selon la taille de la fonctionnalité demandée par l'utilisateur. Néanmoins, ce regroupement pourrait être plus exhaustif, car il faudra le regrouper de manière à représenter les multiples tailles de fonctionnalités demandées. Ce dernier argument se veut aussi une limitation qui peut faire varier les résultats de cette étude.

Malgré ces difficultés, cette étude a expérimenté une mesure alternative de réutilisation fonctionnelle. Cette mesure est basée sur les différents éléments qui font partie de la théorie de

la mesure. En plus elle se veut aussi la contribution des spécialistes dans le domaine. Il faut remarquer aussi la contribution de cette méthode de mesure de la réutilisation fonctionnelle pour le génie logiciel. Le manque de mesures dans l'industrie du logiciel, et surtout dans le contexte de réutilisation, contraint le progrès dans le domaine de l'estimation du coût du logiciel.

On a évalué cette mesure de la réutilisation fonctionnelle par rapport à un ensemble de critères et on peut tirer les conclusions suivantes :

- Elle peut être généralisée à toutes les applications du domaine M.I.S. indépendamment de l'environnement et de la culture de chaque organisation. Dans cette recherche, on a utilisé des données provenant des environnements avec des caractéristiques différentes de travail. En plus cette mesure a été analysée avec des données provenant de plusieurs sites industriels à travers la planète. Toujours est-il, les conditions de recherche utilisées ne nous permettent pas de généraliser cette affirmation à des domaines autre que le domaine M.I.S. avec les résultats obtenus, car il faudra tester cette mesure avec plusieurs ensembles de données et différents scénarios de test.
- En ce qui concerne les scénarios de tests, l'application de cette mesure offre des bénéfices budgétaires (traduit en jours personne) dans tous les scénarios de test utilisés dans cette recherche. Par ailleurs et d'une manière générale les besoins de l'utilisateur dans les entreprises où les collectes de données ont été réalisées se situent au-dessous de 500 points de fonctions.
- Ces scénarios de tests ont identifié une variation toujours au-dessous de la constante de 2.80 j-p/PF (tirée d'ISBSG, 1998) dans le calcul du coût unitaire dans les applications où des ajouts fonctionnels (maintenance) ont été réalisés. Il faut remarquer que les résultats peuvent varier selon le domaine et la grandeur des projets qui ont été mesurés et qui ont servi à l'élaboration des différentes bases de données historiques de points de fonction utilisées.
- Cette méthode de mesure de la réutilisation fonctionnelle est très utile pour identifier et comparer le budget de développement du portefeuille informatique à l'intérieur d'un même site industriel.
- Cette méthode de mesure de la réutilisation fonctionnelle a besoin d'un raffinement dans sa conception qui permettra de mieux adapter les relations entre les différents concepts utilisés. Il est fondamental d'utiliser la méthode « white box reuse » à la place de la méthode « black box reuse » qui est utilisée dans cette mesure proposée. Il est nécessaire de signaler que dans la méthode « black box reuse » une composante doit être réutilisée telle

qu'elle et sans aucune modification. Par contre dans la méthode « white box reuse » une composante peut être adaptée et intégrée dans son environnement hôte (par exemple une composante peut être utilisée pour créer une autre composante) (Barnes et Bollinger, 1991). Ceci nous permettra de mieux identifier la productivité d'un projet logiciel où la réutilisation des composantes a été utilisée.

- La méthode proposée possède un nombre de propriétés désirables dans une mesure, car cette mesure respecte le cycle de vie d'une mesure proposée par Jacquet et Abran (1997).
- Les différentes données utilisées dans cette recherche permettent aussi de montrer la faisabilité des collectes de données utilisée dans l'élaboration des bases de données historiques de points de fonction. Ces dernières servent aussi à l'établissement du portefeuille d'applications informatiques dans une organisation.
- Cette méthode de mesure de la réutilisation fonctionnelle est importante car la réutilisation se veut une stratégie afin d'amoindrir le coût associé au logiciel, ainsi que les risques associés au développement d'une nouvelle application (à partir de zéro).
- Cette recherche a permis de vérifier la démarche utilisée dans l'élaboration de la méthode de réutilisation fonctionnelle proposée. Elle prouve également l'intérêt suscité par cette mesure tant du point de vue théorique que pratique.

6.5 SUGGESTION DE RECHERCHE

Dans l'objectif d'élargir la portée de cette recherche, trois suggestions de recherches potentielles sont présentées :

- L'application de la mesure proposée par Abran et Desharnais à un domaine autre que celui du M.I.S. (temps réel, orienté objet). L'hypothèse à considérer est la suivante : Est-ce que la mesure proposée dans un autre domaine, permettra d'analyser l'impact du changement du domaine, d'envisager des changements requis pour une transposition de contexte, ainsi que la validité de cette mesure.
- Tester la méthode de mesure de la réutilisation fonctionnelle avec une série de scénarios autres que ceux utilisés dans cette étude.

- Il est fondamental d'appliquer la méthode de mesure de la réutilisation fonctionnelle dans la valorisation des projets selon l'approche économique tel que proposé par Molinié (1997) et Jones (1996). Cette approche permettra d'élargir les possibilités d'utilisation de la méthode.

Il est fortement suggéré que le cadre expérimental utilisé dans cette étude soit appliqué à ces quatre suggestions, afin de permettre de mieux comprendre les limites et le potentiel de la méthode proposée par Abran et Desharnais.

Les résultats obtenus ont illustré que la méthode de mesure de la réutilisation fonctionnelle ainsi proposée par Abran et Desharnais (1995) illustre des bénéfices budgétaires (en terme de jours personnes) pour tous les différents scénarios de tests utilisés dans cette étude.

Une mise en garde doit être prise en compte dans cette étude. Cette mise en garde est le fait qu'on n'a pas utilisé des études statistiques (des modèles de régression, etc.) pour soutenir le cadre de travail utilisé dans la vérification de la méthode de mesure proposée. Néanmoins, cela n'empêche pas de conclure sur le bien fondé de la pertinence de la mesure de la réutilisation fonctionnelle proposée.

6.6 AUTRES SUGGESTIONS DE RECHERCHE

Les résultats obtenus ont illustré les forces et les limites de la méthode de mesure proposée. Cette étude a permis de vérifier dans le domaine M.I.S. la démarche utilisée et de montrer l'intérêt tant du point de vue théorique que pratique.

Dans le but d'étendre la portée de ce travail, les suggestions de recherche suivantes sont des voies potentielles afin de mieux explorer le domaine de la mesure de la réutilisation fonctionnelle :

- Utiliser le « feedback » résultant de cette étude afin d'améliorer les travaux futurs dans le domaine de la mesure de réutilisation fonctionnelle du logiciel.
- Un approfondissement des analyses afin de déterminer un cadre de référence qui optimise le présent travail.

Il est finalement fortement suggéré que les méthodes de travail utilisées dans cette étude de recherche puissent être utilisées comme méthodologie pour l'analyse d'autres méthodes de mesures proposées. Pour terminer il faut rappeler qu'à ce stade-ci de la recherche en génie logiciel, l'évaluation de la précision des modèles de mesure devra faire l'objet de recherches supplémentaires, ce qui pose des défis encore plus grands.

APPENDICE A
CADRE DE BASILI
SOMMAIRE DE L'EXPÉRIENCE

Étape 1

1. Définition				
Motivation	Objet	But	Perspective	Portée
<ul style="list-style-type: none"> • Contrôler efficacement le budget informatique. 	<ul style="list-style-type: none"> • Évaluer la méthode de réutilisation fonctionnelle proposée avec des données industrielles. 	<ul style="list-style-type: none"> • Expérimenter la méthode proposée afin de déterminer si elle peut être généralisée à d'autres contextes. • Identification des scénarios de travail qui présentent des avantages budgétaires traduits en termes de jours (personnes). 	<ul style="list-style-type: none"> • Élaboration d'un cadre de travail afin d'évaluer l'objet de cette étude. 	<ul style="list-style-type: none"> • Plusieurs ensembles de données. • Plusieurs scénarios de travail. • Une méthode de la mesure de la réutilisation fonctionnelle.

Étape 2

2. Planification		
Conception expérimentale	Critères	Sélection des mesures
<ul style="list-style-type: none"> • Présentation de la méthode de mesure d'origine. • Identifier le comportement de la méthode. • Vérifier la validité de la méthode. • Cas de test (simulation) faits sur mesure. 	<p>Critères directs</p> <ul style="list-style-type: none"> • Précisions des résultats de la méthode. • L'exactitude pour identifier les éléments réutilisés. • La complexité des cas utilisés. <p>Critères indirects</p> <ul style="list-style-type: none"> • Complexité du processus de construction de la méthode. • Procédures de validation des résultats du modèle. 	<p>Méthode de la mesure</p> <ul style="list-style-type: none"> • Points de fonction version 4.0. <p>Collecte de données</p> <ul style="list-style-type: none"> • Collecte faite manuellement.

Étape 3

3. Exécution		
Préparation	Déroulement	Analyse de données
<ul style="list-style-type: none">• Analyse du modèle de mesure proposé.• Collecte de données• Préparation et sélection de plusieurs ensembles de données de points de fonction.• Élaboration de deux bases de données historiques de points de fonction.• Élaboration des différents scénarios de tests	<ul style="list-style-type: none">• Application de la méthode de mesure proposée aux différents scénarios de tests.	<ul style="list-style-type: none">• Résultats des différents scénarios de tests

Étape 4

4. Interprétation		
Contexte d'interprétation	Extrapolation des résultats	Travaux futurs
<ul style="list-style-type: none"> • Comportement du modèle. • Les limites du modèle. 	<ul style="list-style-type: none"> • Elle peut être généralisée à toutes les applications du domaine M.I.S. • La méthode de mesure de la réutilisation fonctionnelle proposée est conforme au processus de mesure et au cycle de vie d'une mesure proposé par Jacquet et Abran(1996). • La mesure offre des bénéfices budgétaires dans tous les différents scénarios de tests utilisés. • Raffiner la mesure par l'entremise de l'utilisation de la méthode « white box reuse » à la place de la méthode « black box reuse. 	<ul style="list-style-type: none"> • L'application de la mesure proposée par Abran et Desharnais à un domaine autre que celui du M.I.S.(temps réel, orienté objet). • Tester la méthode de mesure de la réutilisation fonctionnelle avec une série de scénarios autres que ceux utilisés dans cette étude. • Raffiner la conception de cette proposition de mesure de la réutilisation fonctionnelle. • Appliquer le modèle de mesure dans une approche économique tel que proposé par Molinié (1997) et Jones (1996).

APPENDICE B

SCÉNARIOS DE TEST DE FONCTIONNALITÉS MOYENNES

SIMULATION DU SCÉNARIO DE TEST DES

FONCTIONNALITÉS MOYENNES

Dans cette section le premier scénario de test concerne les projets informatiques regroupés sous la rubrique de fonctionnalités moyennes. La demande en fonctionnalités pour ce regroupement est entre 200 et 500 points de fonction. Ce scénario de test s'applique à un logiciel de ressources humaines d'une taille approximative de 50000 points de fonction qui demande l'ajout de 500 points de fonction.

**Scénario qui utilise les données de l'institution gouvernementale de Québec
(Valencia et Desharnais)**

Regroupement des projets de 500 points de fonction

- Base de données historique de points de fonction à utiliser : Institution gouvernementale
- Besoin de l'utilisateur : Ajout de 500 PF à une application de 50000 PF
- Durée en jours personnes : 3.25 jours par PF dans les nouvelles applications
2.80 jours par PF dans les projets de maintenance
- Coût pour les nouvelles applications : $500 \text{ PF} \times 3.25 \text{ jours} / \text{PF} = 1625 \text{ jours}$
- Coût pour les applications de maintenance : $500 \text{ PF} \times 2.80 \text{ jours} / \text{PF} = 1400 \text{ jours}$
- Application de la méthode de mesure de la réutilisation fonctionnelle

$$\begin{aligned} \text{PF (Indicateur)} &= \text{PF}^E \times \text{PF}_{\text{GDE}}^E(\text{ratio}) \\ &= 500 \times 16 \% \\ &= 80_{\text{GDE}}^E \end{aligned}$$

$$\begin{aligned} \text{PF}^{\text{R}}(\text{Ratio prédicteur}) &= \text{PF}_{\text{GDE}}^{\text{N}} + \text{PF}_{\text{ENT}}^{\text{N}} + \text{PF}_{\text{SOR}}^{\text{N}} + \text{PF}_{\text{INT}}^{\text{N}} + \text{PF}_{\text{GDI}}^{\text{N}} \\ &= 4.05 \end{aligned}$$

$$\begin{aligned} \text{PF}^{\text{ER}} &= \text{PF}^{\text{E}} + \text{Transactions évitées} \\ &= \text{PF}^{\text{E}} + (\text{PF}_{\text{GDE}}^{\text{E}} \times \text{PF}^{\text{R}}(\text{Prédicteur})) \\ &= 500 + (80 \times 4.05) \\ &= 824 \text{ PF}^{\text{ER}} \end{aligned}$$

Le tableau B.1 permet d'analyser la méthode de mesure de la réutilisation fonctionnelle en utilisant les données historiques de points de fonction qui appartiennent à l'institution gouvernementale de Québec. Cette analyse est réalisée par l'entremise d'un modèle de productivité et elle identifie deux points principaux :

- Lorsqu'on ne tient pas compte de la réutilisation, l'addition de fonctions (maintenance) coûte légèrement moins cher (2.8 jours par PF pour la maintenance au lieu de 3.25 jours par PF pour une nouvelle application), et lorsqu'on tient compte de la mesure de la réutilisation fonctionnelle, la différence de coût unitaire 1.7 jours par PF pour la maintenance (1400 jours / 824 jours) est maintenant beaucoup plus grande et toujours par rapport à 3.25 jours par PF pour une nouvelle application.
- L'alternative de développer une nouvelle application qui n'a pas besoin de ré développer les 324 PF évités dans le projet de maintenance pourrait être d'un total de 824 PF. Si on prend le coût unitaire de 3.25 points de fonction pour développer une nouvelle application, cette alternative de développer 824 PF coûterait 2678 jours (824 PF x 3.25 jours/PF = 2678 jours). À partir de cette perspective l'exemple ci-dessus identifie un avantage économique de 1278 jours personnes (2678 jours – 1400 jours) pour un projet d'ajouts fonctionnels sur son alternative un projet à développer à partir de zéro.

Tableau B.1
Analyse des résultats

Concept	Type de projet	Coût unitaire	Taille en point de fonction	Coût total (jours)
Sans tenir compte de la mesure de la réutilisation (mesure traditionnelle de la méthode de points de fonction)	Nouvelle application	3.25 jours/P.F.	500	1625
	Maintenance	2.80 jours/P.F.		1400
En tenant compte de la mesure de la réutilisation (Mesure proposée qui prend en compte les transactions à ne pas développer)	Maintenance	1.7 Jours/P.F.	824	1400
	Nouvelle application	3.25 jours/P.F.		2678

**Scénario qui utilise les données de l'institution financière canadienne
(Valencia et Nguyen-Kim)**

- Base de données historique de points de fonction à utiliser : Institution financière canadienne (Valencia et Nguyen-Kim)
- Besoin de l'utilisateur : Ajout de 500 PF à une application de 50000 PF
- Durée en jours personnes : 3.25 jours par PF dans les nouvelles applications
2.80 jours par PF dans les projets de maintenance
- Coût pour les nouvelles applications : 500 PF x 3.25 jours / PF = 1625 jours
- Coût pour les applications de maintenance : 500 PF x 2.80 jours / PF = 1400 jours
- Application de la méthode de mesure de la réutilisation fonctionnelle

$$\begin{aligned} \text{PF (Indicateur)} &= \text{PF}^E \times \text{PF}_{\text{GDE}}^E(\text{ratio}) \\ &= 500 \times 14 \% \\ &= 70_{\text{GDE}} \end{aligned}$$

$$\begin{aligned} \text{PF}^R(\text{Ratio prédicteur}) &= \text{PF}_{\text{GDE}}^N + \text{PF}_{\text{ENT}}^N + \text{PF}_{\text{SOR}}^N + \text{PF}_{\text{INT}}^N + / \text{PF}_{\text{GDI}}^N \\ &= 3 \end{aligned}$$

$$\begin{aligned} \text{PF}^{\text{ER}} &= \text{PF}^E + \text{Transactions évitées} \\ &= \text{PF}^E + (\text{PF}_{\text{GDE}}^E \times \text{PF}^R(\text{Prédicteur})) \\ &= 500 + (70 \times 3) \\ &= 710 \text{ PF}^{\text{ER}} \end{aligned}$$

Le tableau B.2 permet d'analyser l'utilisation de la méthode de mesure de la réutilisation fonctionnelle en utilisant les données historiques de points de fonction qui appartiennent à l'institution financière canadienne (Valencia et Nguyen-Kim). Cette analyse est réalisée par l'entremise d'un modèle de productivité et elle identifie deux points principaux :

- Lorsqu'on ne tient pas compte de la réutilisation, l'addition de fonctions (maintenance) coûte légèrement moins cher (2.8 jours par PF pour la maintenance au lieu de 3.25 jours

par PF pour une nouvelle application), et lorsqu'on tient compte de la mesure de la réutilisation fonctionnelle, la différence de coût unitaire 1.94 jours par PF pour la maintenance (1400 / 710) est maintenant beaucoup plus grande et toujours par rapport à 3.25 jours par PF pour nouvelle application.

- L'alternative de développer une nouvelle application qui n'a pas besoin de ré développer les 210 PF évités dans le projet de maintenance pourrait être d'un total de 710 PF. Si on prend le coût unitaire de 3.25 points de fonction pour développer une nouvelle application, cette alternative de développer 710 PF coûterait 2307 jours (710 PF x 3.25 jours/PF = 2307 jours). À partir de cette perspective l'exemple ci-dessus identifie un avantage économique de 907 jours personnes (2307 jours – 1400 jours) pour un projet d'ajouts fonctionnels sur son alternative un projet à développer à partir de zéro.

Tableau B.2

Analyse des résultats

Concept	Type de projet	Coût unitaire	Taille en point de fonction	Coût total (jours)
Sans tenir compte de la mesure de la réutilisation (mesure traditionnelle de la méthode de points de fonction)	Nouvelle application	3.25 jours/P.F.	500	1625
	Maintenance	2.80 jours/P.F.		1400
En tenant compte de la mesure de la réutilisation (Mesure proposée qui prend en compte les transactions à ne pas développer)	Maintenance	1.94 jours/P.F.	710	1400
	Nouvelle application	3.25 jours/P.F.		2307

**Scénario qui utilise les données de l'institution financière canadienne
(Abran)**

- Base de données historique de points de fonction à utiliser : Institution financière canadienne (Abran)
- Besoin de l'utilisateur : Ajout de 500 PF à une application de 50000 PF
- Durée en jours personnes : 3.25 jours par PF dans les nouvelles applications
2.80 jours par PF dans les projets de maintenance
- Coût pour les nouvelles applications : 500 PF x 3.25 jours / PF = 1625 jours
- Coût pour les applications de maintenance : 500 PF x 2.80 jours / PF = 1400 jours
- Application de la méthode de mesure de la réutilisation fonctionnelle

$$PF^{(Indicateur)} = PF^E \times PF_{GDE}^E(\text{ratio})$$

$$= 500 \times 38 \%$$

$$= 190^{E_{GDE}}$$

$$PF^R(\text{Ratio prédicteur}) = PF_{GDE}^N + PF_{ENT}^N + PF_{SOR}^N + PF_{INT}^N + / PF_{GDI}^N$$

$$= 2.74$$

$$PF^{ER} = PF^E + \text{Transactions évitées}$$

$$= PF^E + (PF_{GDE}^E \times PF^R(\text{Prédicteur}))$$

$$= 500 + (190 \times 2.74)$$

$$= 1020 PF^{ER}$$

Le tableau B.3 permet d'analyser l'utilisation de la méthode de mesure de la réutilisation fonctionnelle en utilisant les données historiques de points de fonction qui appartiennent à l'institution financière canadienne (Abran). Cette analyse est réalisée par l'entremise d'un modèle de productivité et elle identifie deux points principaux :

- Lorsqu'on ne tient pas compte de la réutilisation, l'addition de fonctions (maintenance) coûte légèrement moins cher (2.8 jours par PF pour la maintenance au lieu de 3.25 jours

par PF pour une nouvelle application), et lorsqu'on tient compte de la mesure de la réutilisation fonctionnelle, la différence de coût unitaire 1.37 jours par PF pour la maintenance (1400 / 1020) est maintenant beaucoup plus grande et toujours par rapport à 3.25 jours par PF pour une nouvelle application.

- L'alternative de développer une nouvelle application qui n'a pas besoin de ré développer les 520 PF évités dans le projet de maintenance pourrait être d'un total de 1020 PF. Si on prend le coût unitaire de 3.25 points de fonction pour développer une nouvelle application, cette alternative de développer 1020 PF coûterait 3315 jours (1020 PF x 3.25 jours/PF = 3315 jours). À partir de cette perspective l'exemple ci-dessus identifie un avantage économique de 1915 jours personnes (3315 jours – 1400 jours) pour un projet d'ajouts fonctionnels sur son alternative un projet à développer à partir de zéro.

Tableau B.3

Analyse des résultats

Concept	Type de projet	Coût unitaire	Taille en point de fonction	Coût total (jours)
Sans tenir compte de la mesure de la réutilisation (mesure traditionnelle de la méthode de points de fonction)	Nouvelle application	3.25 jours/P.F.	500	1625
	Maintenance	2.80 jours/P.F.		1400
En tenant compte de la mesure de la réutilisation (Mesure proposée qui prend en compte les transactions à ne pas développer)	Maintenance	1.37 jours/P.F.	1020	1400
	Nouvelle application	3.25 jours/P.F.		3315

Scénario qui utilise les données tirées d'ISBSG (1998)

- Base de données historique de points de fonction à utiliser : ISBSG (1998)r
- Besoin de l'utilisateur : Ajout de 500 PF à une application de 50000 PF
- Durée en jours personnes : 3.25 jours par PF dans les nouvelles applications
2.80 jours par PF dans les projets de maintenance
- Coût pour les nouvelles applications : 500 PF x 3.25 jours / PF = 1625 jours
- Coût pour les applications de maintenance : 500 PF x 2.80 jours / PF = 1400 jours

- Application de la méthode de mesure de la réutilisation fonctionnelle

$$\begin{aligned}
 \text{PF (Indicateur)} &= \text{PF}^E \times \text{PF}_{\text{GDE}}^E(\text{ratio}) \\
 &= 500 \times 15 \\
 &= 75^E_{\text{GDE}}
 \end{aligned}$$

$$\begin{aligned}
 \text{PF}^R(\text{Ratio prédicteur}) &= \text{PF}_{\text{GDE}}^N + \text{PF}_{\text{ENT}}^N + \text{PF}_{\text{SOR}}^N + \text{PF}_{\text{INT}}^N + / \text{PF}_{\text{GDI}}^N \\
 &= 3.48
 \end{aligned}$$

$$\begin{aligned}
 \text{PF}^{\text{ER}} &= \text{PF}^E + \text{Transactions évitées} \\
 &= \text{PF}^E + (\text{PF}_{\text{GDE}}^E \times \text{PF}^R(\text{Prédicteur})) \\
 &= 500 + (75 \times 3.48) \\
 &= 761 \text{ PF}^{\text{ER}}
 \end{aligned}$$

Le tableau B.4 permet d'analyser l'utilisation de la méthode de mesure de la réutilisation fonctionnelle en utilisant les données historiques de points de fonction qui appartiennent à ISBSG (1998). Cette analyse est réalisée par l'entremise d'un modèle de productivité et elle identifie deux points principaux :

- Lorsqu'on ne tient pas compte de la réutilisation, l'addition de fonctions (maintenance) coûte légèrement moins cher (2.8 jours par PF pour la maintenance au lieu de 3.25 jours par PF pour une nouvelle application), et lorsqu'on tient compte de la mesure de la

réutilisation fonctionnelle, la différence de coût unitaire 1.84 jours par PF pour la maintenance (1400 jours / 761 jours) est maintenant beaucoup plus grande et toujours par rapport à 3.25 jours par PF pour une nouvelle application.

- L'alternative de développer une nouvelle application qui n'a pas besoin de ré développer les 261 PF évités dans le projet de maintenance pourrait être d'un total de 761 PF. Si on prend le coût unitaire de 3.25 points de fonction pour développer une nouvelle application, cette alternative de développer 761 PF coûterait 2473 jours (761 PF x 3.25 jours/PF = 2473 jours). À partir de cette perspective l'exemple ci-dessus identifie un avantage économique de 1073 jours personnes (2473 jours – 1400 jours) pour un projet d'ajouts fonctionnels sur son alternative un projet à développer à partir de zéro.

Tableau B.4

Analyse des résultats

Concept	Type de projet	Coût unitaire	Taille en point de fonction	Coût total (jours)
Sans tenir compte de la mesure de la réutilisation (mesure traditionnelle de la méthode de points de fonction)	Nouvelle application	3.25 jours/P.F.	500	1625
	Maintenance	2.80 jours/P.F.		1400
En tenant compte de la mesure de la réutilisation (Mesure proposée qui prend en compte les transactions à ne pas développer)	Maintenance	1.84 jours/P.F.	761	1400
	Nouvelle application	3.25 jours/P.F.		2473

Tableau B.5

Sommaire des résultats des différentes transformations de la méthode de mesure de réutilisation pour les différents scénarios de tests appartenant au regroupement des fonctionnalités moyennes

Scénarios Selon la base de données utilisée	PF (Indicateur)	PF (Prédicteur)	Transactions évitées	Impact dans le calcul du coût unitaire	
				Nouvelle Applicat.	Mainte nance
Institution gouvernementale de Québec	80	4.05	324	3.25 jp/PF	1.70jp/PF
Institution financière (Valencia et Nguyen-Kim)	70	3	210	3.25 jp/PF	1.94jp/PF
Institution financière (Abran)	190	2.74	520	3.25 jp/PF	1.37jp/PF
ISBSG (1998)	75	3.48	261	3.25 jp/PF	1.84jp/PF

Constantes	
Coût unitaire	
Développement	Maintenance
3.25 j.p. / P.F.	2.80 j.p. / P.F.

APPENDICE C

SCÉNARIOS DE TEST DES FONCTIONNALITÉS MINEURES

SIMULATION DU SCÉNARIO DE TEST DES

FONCTIONNALITÉS MINEURES

Dans cette section le premier scénario de test concerne les projets informatiques regroupés sous la rubrique de fonctionnalités mineures. La demande en fonctionnalités pour ce regroupement est entre 75 et 200 points de fonction. Ce scénario de test s'applique à un logiciel de ressources humaines d'une taille approximative de 50000 points de fonction qui demande l'ajout de 200 points de fonction.

**Scénario qui utilise les données de l'institution gouvernementale de Québec
(Valencia et Desharnais)**

- Base de données historique de points de fonction à utiliser : Institution gouvernementale
- Besoin de l'utilisateur : Ajout de 200 PF à une application de 50000 PF
- Durée en jours personnes : 3.25 jours par PF dans les nouvelles applications
2.80 jours par PF dans les projets de maintenance
- Coût pour les nouvelles applications : 200 PF x 3.25 jours / PF = 650 jours
- Coût pour les applications de maintenance : 200 PF x 2.80 jours / PF = 560 jours
- Application de la méthode de mesure de la réutilisation fonctionnelle

$$\begin{aligned} \text{PF (Indicateur)} &= \text{PF}^E \times \text{PF}_{\text{GDE}}^E(\text{ratio}) \\ &= 200 \times 16 \% \\ &= 32^E_{\text{GDE}} \end{aligned}$$

$$\begin{aligned} \text{PF}^R (\text{Ratio prédicteur}) &= \text{PF}_{\text{GDE}}^N + \text{PF}_{\text{ENT}}^N + \text{PF}_{\text{SOR}}^N + \text{PF}_{\text{INT}}^N + / \text{PF}_{\text{GDI}}^N \\ &= 4.05 \end{aligned}$$

$$\begin{aligned} \text{PF}^{\text{ER}} &= \text{PF}^{\text{E}} + \text{Transactions évitées} \\ &= \text{PF}^{\text{E}} + (\text{PF}_{\text{GDE}}^{\text{E}} \times \text{PF}^R (\text{Prédicteur})) \\ &= 200 + (32 \times 4.05) \\ &= 330 \text{ PF}^{\text{ER}} \end{aligned}$$

Le tableau C.1 permet d'analyser la méthode de mesure de la réutilisation fonctionnelle en utilisant les données historiques de points de fonction qui appartiennent à l'institution gouvernementale de Québec. Cette analyse est réalisée par l'entremise d'un modèle de productivité et elle identifie deux points principaux :

- Lorsqu'on ne tient pas compte de la réutilisation, l'addition de fonctions (maintenance) coûte légèrement moins cher (2.9 jours par PF pour la maintenance au lieu de 3.25 jours par PF pour une nouvelle application), et lorsqu'on tient compte de la mesure de la réutilisation fonctionnelle, la différence de coût unitaire 1.70 jours par PF pour la maintenance (560 jours / 330 jours) est maintenant beaucoup plus grande et toujours par rapport à 3.25 jours par PF pour une nouvelle application.
- L'alternative de développer une nouvelle application qui n'a pas besoin de ré développer les 130 PF évités dans le projet de maintenance pourrait être d'un total de 330 PF. Si on prend le coût unitaire de 3.25 jours par point de fonction pour développer une nouvelle application, cette alternative de développer 330 PF coûterait 1072 jours (330 PF x 3.25 jours/PF = 1072 jours). À partir de cette perspective l'exemple ci-dessus identifie un avantage économique de 512 jours personnes (1072 jours – 560 jours) pour un projet d'ajouts fonctionnels sur son alternative un projet à développer à partir de zéro.

Tableau C.1
Analyse des résultats

Concept	Type de projet	Coût unitaire	Taille en point de fonction	Coût total (jours)
Sans tenir compte de la mesure de la réutilisation (mesure traditionnelle de la méthode de points de fonction)	Nouvelle application	3.25 jours/P.F.	200	650
	Maintenance	2.80 jours/P.F.		560
En tenant compte de la mesure de la réutilisation (Mesure proposée qui prend en compte les transactions à ne pas développer)	Maintenance	1.70 Jours/P.F.	330	560
	Nouvelle application	3.25 jours/P.F.		1072

**Scénario qui utilise les données de l'institution financière canadienne
(Valencia et Nguyen-Kim)**

- Base de données historique de points de fonction à utiliser : Institution financière canadienne (Valencia et Nguyen-Kim)
- Besoin de l'utilisateur : Ajout de 200 PF à une application de 50000 PF
- Durée en jours personnes : 3.25 jours par PF dans les nouvelles applications
2.80 jours par PF dans les projets de maintenance
- Coût pour les nouvelles applications : 200 PF x 3.25 jours / PF = 650 jours
- Coût pour les applications de maintenance : 200 PF x 2.80 jours / PF = 560 jours

- Application de la méthode de mesure de la réutilisation fonctionnelle

$$\begin{aligned} \text{PF (Indicateur)} &= \text{PF}^E \times \text{PF}_{\text{GDE}}^E(\text{ratio}) \\ &= 200 \times 14 \% \\ &= 28^E_{\text{GDE}} \end{aligned}$$

$$\begin{aligned} \text{PF}^R(\text{Ratio prédicteur}) &= \text{PF}_{\text{GDE}}^N + \text{PF}_{\text{ENT}}^N + \text{PF}_{\text{SOR}}^N + \text{PF}_{\text{INT}}^N + / \text{PF}_{\text{GDI}}^N \\ &= 3 \end{aligned}$$

$$\begin{aligned} \text{PF}^{\text{ER}} &= \text{PF}^E + \text{Transactions évitées} \\ &= \text{PF}^E + (\text{PF}_{\text{GDE}}^E \times \text{PF}^R(\text{Prédicteur})) \\ &= 200 + (28 \times 3) \\ &= 284 \text{ PF}^{\text{ER}} \end{aligned}$$

Le tableau C.2 permet d'analyser l'utilisation de la méthode de mesure de la réutilisation fonctionnelle en utilisant les données historiques de points de fonction qui appartiennent à

l'institution financière canadienne (Valencia et Nguyen-Kim). Cette analyse est réalisée par l'entremise d'un modèle de productivité et elle identifie deux points principaux :

- Lorsqu'on ne tient pas compte de la réutilisation, l'addition de fonctions (maintenance) coûte légèrement moins cher (2.8 jours par PF pour la maintenance au lieu de 3.25 jours par PF pour une nouvelle application), et lorsqu'on tient compte de la mesure de la réutilisation fonctionnelle, la différence de coût unitaire 1.97 jours par PF pour la maintenance (560 / 284) est maintenant beaucoup plus grande et toujours par rapport à 3.25 jours par PF pour une nouvelle application.
- L'alternative de développer une nouvelle application qui n'a pas besoin de ré développer les 84 PF évités dans le projet de maintenance pourrait être d'un total de 284 PF. Si on prend le coût unitaire de 3.25 points de fonction pour développer une nouvelle application, cette alternative de développer 284 PF coûterait 923 jours (284 PF x 3.25 jours/PF = 923 jours). À partir de cette perspective l'exemple ci-dessus identifie un avantage économique de 363 jours personnes (923 jours – 560 jours) pour un projet d'ajouts fonctionnels sur son alternative un projet à développer à partir de zéro.

Tableau C.2

Analyse des résultats

Concept	Type de projet	Coût unitaire	Taille en point de fonction	Coût total (jours)
Sans tenir compte de la mesure de la réutilisation (mesure traditionnelle de la méthode de points de fonction)	Nouvelle application	3.25 jours/P.F.	200	650
	Maintenance	2.80 jours/P.F.		560
En tenant compte de la mesure de la réutilisation (Mesure proposée qui prend en compte les transactions à ne pas développer)	Maintenance	1.97 jours/P.F.	284	560

	Nouvelle application	3.25 jours/P.F.	923
--	----------------------	-----------------	-----

Scénario qui utilise les données de l'institution financière canadienne (Abran)

- Base de données historique de points de fonction à utiliser : Institution financière canadienne (Abran)
- Besoin de l'utilisateur : Ajout de 200 PF à une application de 50000 PF
- Durée en jours personnes : 3.25 jours par PF dans les nouvelles applications
2.80 jours par PF dans les projets de maintenance
- Coût pour les nouvelles applications : 200 PF x 3.25 jours / PF = 650 jours
- Coût pour les applications de maintenance : 200 PF x 2 jours / PF = 560 jours
- Application de la méthode de mesure de la réutilisation fonctionnelle

$$PF^E(\text{Indicateur}) = PF^E \times PF_{GDE}^E(\text{ratio})$$

$$= 200 \times 38 \%$$

$$= 76^E_{GDE}$$

$$PF^R(\text{Ratio prédicteur}) = PF_{GDE}^N + PF_{ENT}^N + PF_{SOR}^N + PF_{INT}^N + / PF_{GDI}^N$$

$$= 2.74$$

$$PF^{ER} = PF^E + \text{Transactions évitées}$$

$$= PF^E + (PF_{GDE}^E \times PF^R(\text{Prédicteur}))$$

$$= 200 + (76 \times 2.74)$$

$$= 408 PF^{ER}$$

Le tableau C.3 permet d'analyser l'utilisation de la méthode de mesure de la réutilisation fonctionnelle en utilisant les données historiques de points de fonction qui appartiennent à l'institution financière canadienne (Abran). Cette analyse est réalisée par l'entremise d'un modèle de productivité et elle identifie deux points principaux :

- Lorsqu'on ne tient pas compte de la réutilisation, l'addition de fonctions (maintenance) coûte légèrement moins cher (2.8 jours par PF pour la maintenance au lieu de 3.25 jours par PF pour une nouvelle application), et lorsqu'on tient compte de la mesure de la réutilisation fonctionnelle, la différence de coût unitaire 1.37 jours par PF pour la maintenance (560 / 408) est maintenant beaucoup plus grande et toujours par rapport à 3.25 jours par PF pour une nouvelle application.
- L'alternative de développer une nouvelle application qui n'a pas besoin de ré développer les 208 PF évités dans le projet de maintenance pourrait être d'un total de 408 PF. Si on prend le coût unitaire de 3.25 points de fonction pour développer une nouvelle application, cette alternative de développer 408 PF coûterait 1326 jours (408 PF x 3.25 jours/PF = 1326 jours). À partir de cette perspective l'exemple ci-dessus identifie un avantage économique de 766 jours personnes (1326 jours – 560 jours) pour un projet d'ajouts fonctionnels sur son alternative un projet à développer à partir de zéro.

Tableau C.3

Analyse des résultats

Concept	Type de projet	Coût unitaire	Taille en point de fonction	Coût total (jours)
Sans tenir compte de la mesure de la réutilisation (mesure traditionnelle de la méthode de points de fonction)	Nouvelle application	3.25 jours/P.F.	200	650
	Maintenance	2.80 jours/P.F.		560
En tenant compte de la mesure de la réutilisation (Mesure proposée qui prend en compte les transactions à ne pas développer)	Maintenance	1.37 jours/P.F.	408	560

	Nouvelle application	3.25 jours/P.F.	1326
--	----------------------	-----------------	------

Scénario qui utilise les données tirées d'ISBSG (1998)

- Base de données historique de points de fonction à utiliser : ISBSG (1998)
- Besoin de l'utilisateur : Ajout de 200 PF à une application de 50000 PF
- Durée en jours personnes : 3.25 jours par PF dans les nouvelles applications
2.80 jours par PF dans les projets de maintenance
- Coût pour les nouvelles applications : 200 PF x 3.25 jours / PF = 650 jours
- Coût pour les applications de maintenance : 200 PF x 2.80 jours / PF = 560 jours
- Application de la méthode de mesure de la réutilisation fonctionnelle

$$\begin{aligned}
 \text{PF}^{\text{E}}(\text{Indicateur}) &= \text{PF}^{\text{E}} \times \text{PF}_{\text{GDE}}^{\text{E}}(\text{ratio}) \\
 &= 200 \times 15 \\
 &= 30^{\text{E}}_{\text{GDE}}
 \end{aligned}$$

$$\begin{aligned}
 \text{PF}^{\text{R}}(\text{Ratio prédicteur}) &= \text{PF}_{\text{GDE}}^{\text{N}} + \text{PF}_{\text{ENT}}^{\text{N}} + \text{PF}_{\text{SOR}}^{\text{N}} + \text{PF}_{\text{INT}}^{\text{N}} / \text{PF}_{\text{GDI}}^{\text{N}} \\
 &= 3.48
 \end{aligned}$$

$$\begin{aligned}
 \text{PF}^{\text{ER}} &= \text{PF}^{\text{E}} + \text{Transactions évitées} \\
 &= \text{PF}^{\text{E}} + (\text{PF}_{\text{GDE}}^{\text{E}} \times \text{PF}^{\text{R}}(\text{Prédicteur})) \\
 &= 200 + (30 \times 3.48) \\
 &= 304 \text{ PF}^{\text{ER}}
 \end{aligned}$$

Le tableau C.4 permet d'analyser l'utilisation de la méthode de mesure de la réutilisation fonctionnelle en utilisant les données historiques de points de fonction qui appartiennent à

ISBSG (1998). Cette analyse est réalisée par l'entremise d'un modèle de productivité et elle identifie deux points principaux :

- Lorsqu'on ne tient pas compte de la réutilisation, l'addition de fonctions (maintenance) coûte légèrement moins cher (2.8 jours par PF pour la maintenance au lieu de 3.25 jours par PF pour une nouvelle application), et lorsqu'on tient compte de la mesure de la réutilisation fonctionnelle, la différence de coût unitaire 1.84 jours par PF pour la maintenance (560 jours / 304) est maintenant beaucoup plus grande et toujours par rapport à 3.25 jours par PF pour une nouvelle application.
- L'alternative de développer une nouvelle application qui n'a pas besoin de ré développer les 104 PF évités dans le projet de maintenance pourrait être d'un total de 304 PF. Si on prend le coût unitaire de 3.25 points de fonction pour développer une nouvelle application, cette alternative de développer 304 PF coûterait 988 jours (304 PF x 3.25 jours/PF = 988 jours). À partir de cette perspective l'exemple ci-dessus identifie un avantage économique de 428 jours personnes (988 jours – 560 jours) pour un projet d'ajouts fonctionnels sur son alternative un projet à développer à partir de zéro.

Tableau C.4

Analyse des résultats

Concept	Type de projet	Coût unitaire	Taille en point de fonction	Coût total (jours)
Sans tenir compte de la mesure de la réutilisation (mesure traditionnelle de la méthode de points de fonction)	Nouvelle application	3.25 jours/P.F.	200	650
	Maintenance	2.80 jours/P.F.		560
En tenant compte de la mesure de la réutilisation (Mesure proposée qui prend en compte les transactions à ne pas développer)	Maintenance	1.84 jours/P.F.	304	560

	Nouvelle application	3.25 jours/P.F.		988
--	----------------------	-----------------	--	-----

Tableau C.5

Sommaire des résultats des différentes transformations de la méthode de mesure de réutilisation pour les différents scénarios de tests appartenant au regroupement des fonctionnalités mineures

Scénarios Selon la base de données utilisée	PF (Indicateur)	PF (Prédicteur)	Transactions évitées	Impact dans le calcul du coût unitaire	
				Nouvelle Applicat.	Mainte Nance
Institution gouvernementale de Québec	32	4.05	129	3.25 jp/PF	1.70jp/PF
Institution financière (Valencia et Nguyen-Kim)	28	3	84	3.25 jp/PF	1.97jp/PF
Institution financière (Abran)	76	2.74	208	3.25 jp/PF	1.37jp/PF
ISBSG (1998)	30	3.48	104	3.25 jp/PF	1.84jp/PF

Constantes	
Coût unitaire	
Développement	Maintenance
3.25 j.p. / P.F.	2.80 j.p. / P.F.

APPENDICE D

SCÉNARIOS DE TEST DES FONCTIONNALITÉS PETITES

SIMULATION DU SCÉNARIO DE TEST DES

FONCTIONNALITÉS PETITES

Dans cette section le premier scénario de test concerne les projets informatiques regroupés sous la rubrique de fonctionnalités majeures. La demande en fonctionnalités pour ce regroupement est entre 20 et 75 points de fonction. Ce scénario de test s'applique à un logiciel de ressources humaines d'une taille approximative de 50000 points de fonction qui demande l'ajout de 75 points de fonction.

**Scénario qui utilise les données de l'institution gouvernementale de Québec
(Valencia et Desharnais)**

Dans ce scénario de test, nous voulons expérimenter le comportement de la méthode de mesure de la réutilisation fonctionnelle avec la base de données historique de points de fonction construite à partir des projets informatiques de l'institution gouvernementale de Québec. Dans ce scénario de test, l'utilisateur demande des améliorations de 75 points de fonction à une application existante de 50000 points de fonction.

- Base de données historique de points de fonction à utiliser : Institution gouvernementale
- Besoin de l'utilisateur : Ajout de 75 PF à une application de 50000 PF
- Durée en jours personnes : 3.25 jours par PF dans les nouvelles applications
2.80 jours par PF dans les projets de maintenance
- Coût pour les nouvelles applications : $75 \text{ PF} \times 3.25 \text{ jours} / \text{PF} = 244 \text{ jours}$
- Coût pour les applications de maintenance : $75 \text{ PF} \times 2.80 \text{ jours} / \text{PF} = 210 \text{ jours}$
- Application de la méthode de mesure de la réutilisation fonctionnelle

$$\begin{aligned}
 \text{PF (Indicateur)} &= \text{PF}^E \times \text{PF}_{\text{GDE}}^E(\text{ratio}) \\
 &= 75 \times 16 \% \\
 &= 12^E_{\text{GDE}}
 \end{aligned}$$

$$\begin{aligned}
 \text{PF}^R(\text{Ratio prédicteur}) &= \text{PF}_{\text{GDE}}^N + \text{PF}_{\text{ENT}}^N + \text{PF}_{\text{SOR}}^N + \text{PF}_{\text{INT}}^N + / \text{PF}_{\text{GDI}}^N \\
 &= 4.05
 \end{aligned}$$

$$\begin{aligned}
 \text{PF}^{\text{ER}} &= \text{PF}^E + \text{Transactions évitées} \\
 &= \text{PF}^E + (\text{PF}_{\text{GDE}}^E \times \text{PF}^R(\text{Prédicteur})) \\
 &= 75 + (12 \times 3.48) \\
 &= 116 \text{ PF}^{\text{ER}}
 \end{aligned}$$

Le tableau D.1 permet d'analyser la méthode de mesure de la réutilisation fonctionnelle en utilisant les données historiques de points de fonction qui appartiennent à l'institution gouvernementale de Québec. Cette analyse est réalisée par l'entremise d'un modèle de productivité et elle identifie deux points principaux :

- Lorsqu'on ne tient pas compte de la réutilisation, l'addition de fonctions (maintenance) coûte légèrement moins cher (2.8 jours par PF pour la maintenance au lieu de 3.25 jours par PF pour une nouvelle application), et lorsqu'on tient compte de la mesure de la réutilisation fonctionnelle, la différence de coût unitaire 1.81 jours par PF pour la maintenance (210 jours / 116 jours) est maintenant beaucoup plus grande et toujours par rapport à 3.25 jours par PF pour une nouvelle application.
- L'alternative de développer une nouvelle application qui n'a pas besoin de ré développer les 41 PF évités dans le projet de maintenance pourrait être d'un total de 116 PF. Si on prend le coût unitaire de 3.25 points de fonction pour développer une nouvelle application, cette alternative de développer 116 PF coûterait 377 jours (116 PF x 3.25 jours/PF = 377 jours). À partir de cette perspective l'exemple ci-dessus identifie un avantage économique de 167 jours personnes (377 jours – 210 jours) pour un projet d'ajouts fonctionnels sur son alternative un projet à développer à partir de zéro.

Tableau D.1
Analyse des résultats

Concept	Type de projet	Coût unitaire	Taille en point de fonction	Coût total (jours)
Sans tenir compte de la mesure de la réutilisation (mesure traditionnelle de la méthode de points de fonction)	Nouvelle application	3.25 jours/P.F.	75	244
	Maintenance	2.80 jours/P.F.		210
En tenant compte de la mesure de la réutilisation (Mesure proposée qui prend en compte les transactions à ne pas développer)	Maintenance	1.81 Jours/P.F.	116	210
	Nouvelle application	3.25 jours/P.F.		377

Scénario qui utilise les données de l'institution financière canadienne

(Valencia et Nguyen-Kim)

- Base de données historique de points de fonction à utiliser : Institution financière canadienne (Valencia et Nguyen-Kim)
- Besoin de l'utilisateur : Ajout de 75 PF à une application de 50000 PF
- Durée en jours personnes : 3.25 jours par PF dans les nouvelles applications
2.80 jours par PF dans les projets de maintenance
- Coût pour les nouvelles applications : 75 PF x 3.25 jours / PF = 244 jours
- Coût pour les applications de maintenance : 75 PF x 2.80 jours / PF = 210 jours

- Application de la méthode de mesure de la réutilisation fonctionnelle

$$PF^E (\text{Indicateur}) = PF^E \times PF_{GDE}^E (\text{ratio})$$

$$= 75 \times 14 \%$$

$$= 10_{GDE}^E$$

$$PF^R (\text{Ratio prédicteur}) = PF_{GDE}^N + PF_{ENT}^N + PF_{SOR}^N + PF_{INT}^N + / PF_{GDI}^N$$

$$= 3$$

$$PF^{ER} = PF^E + \text{Transactions évitées}$$

$$= PF^E + (PF_{GDE}^E \times PF^R (\text{Prédicteur}))$$

$$= 75 + (10 \times 3)$$

$$= 105 PF^{ER}$$

Le tableau D.2 permet d'analyser l'utilisation de la méthode de mesure de la réutilisation fonctionnelle en utilisant les données historiques de points de fonction qui appartiennent à l'institution financière canadienne (Valencia et Nguyen-Kim). Cette analyse est réalisée par l'entremise d'un modèle de productivité et elle identifie deux points principaux :

- Lorsqu'on ne tient pas compte de la réutilisation, l'addition de fonctions (maintenance) coûte légèrement moins cher (2.8 jours par PF pour la maintenance au lieu de 3.25 jours par PF pour une nouvelle application), et lorsqu'on tient compte de la mesure de la

réutilisation fonctionnelle, la différence de coût unitaire 2 jours par PF pour la maintenance (210 / 105) est maintenant beaucoup plus grande et toujours par rapport à 3.25 jours par PF pour une nouvelle application.

- L'alternative de développer une nouvelle application qui n'a pas besoin de ré développer les 30 PF évités dans le projet de maintenance pourrait être d'un total de 105 PF. Si on prend le coût unitaire de 3.25 points de fonction pour développer une nouvelle application, cette alternative de développer 105 PF coûterait 341 jours (105 PF x 3.25 jours/PF = 341 jours). À partir de cette perspective l'exemple ci-dessus identifie un avantage économique de 131 jours personnes (341 jours – 210 jours) pour un projet d'ajouts fonctionnels sur son alternative un projet à développer à partir de zéro.

Tableau D.2

Analyse des résultats

Concept	Type de projet	Coût unitaire	Taille en point de fonction	Coût total (jours)
Sans tenir compte de la mesure de la réutilisation (mesure traditionnelle de la méthode de points de fonction)	Nouvelle application	3.25 jours/P.F.	75	244
	Maintenance	2.80 jours/P.F.		210
En tenant compte de la mesure de la réutilisation (Mesure proposée qui prend en compte les transactions à ne pas développer)	Maintenance	2 jours/P.F.	105	210
	Nouvelle application	3.25 jours/P.F.		341

Scénario qui utilise les données de l'institution financière canadienne (Abran)

- Base de données historique de points de fonction à utiliser : Institution financière canadienne (Abran)
- Besoin de l'utilisateur : Ajout de 75 PF à une application de 50000 PF
- Durée en jours personnes : 3.25 jours par PF dans les nouvelles applications
2.80 jours par PF dans les projets de maintenance
- Coût pour les nouvelles applications : $75 \text{ PF} \times 3.25 \text{ jours} / \text{PF} = 244 \text{ jours}$
- Coût pour les applications de maintenance : $75 \text{ PF} \times 2 \text{ jours} / \text{PF} = 150 \text{ jours}$
- Application de la méthode de mesure de la réutilisation fonctionnelle

$$\begin{aligned} \text{PF (Indicateur)} &= \text{PF}^E \times \text{PF}_{\text{GDE}}^E(\text{ratio}) \\ &= 75 \times 38 \% \\ &= 28 \text{ }^E_{\text{GDE}} \end{aligned}$$

$$\begin{aligned} \text{PF}^R(\text{Ratio prédicteur}) &= \text{PF}_{\text{GDE}}^N + \text{PF}_{\text{ENT}}^N + \text{PF}_{\text{SOR}}^N + \text{PF}_{\text{INT}}^N + / \text{PF}_{\text{GDI}}^N \\ &= 2.74 \end{aligned}$$

$$\begin{aligned} \text{PF}^{\text{ER}} &= \text{PF}^E + \text{Transactions évitées} \\ &= \text{PF}^E + (\text{PF}_{\text{GDE}}^E \times \text{PF}^R(\text{Prédicteur})) \\ &= 75 + (28 \times 2.74) \\ &= 151 \text{ PF}^{\text{ER}} \end{aligned}$$

Le tableau D.3 permet d'analyser l'utilisation de la méthode de mesure de la réutilisation fonctionnelle en utilisant les données historiques de points de fonction qui appartiennent à l'institution financière canadienne (Abran). Cette analyse est réalisée par l'entremise d'un modèle de productivité et elle identifie deux points principaux :

- Lorsqu'on ne tient pas compte de la réutilisation, l'addition de fonctions (maintenance) coûte légèrement moins cher (2.8 jours par PF pour la maintenance au lieu de 3.25 jours par PF pour une nouvelle application), et lorsqu'on tient compte de la mesure de la

réutilisation fonctionnelle, la différence de coût unitaire 1.39 jours par PF pour la maintenance (210 / 151) est maintenant beaucoup plus grande et toujours par rapport à 3.25 jours par PF pour une nouvelle application.

- L'alternative de développer une nouvelle application qui n'a pas besoin de ré développer les 76 PF évités dans le projet de maintenance pourrait être d'un total de 151 PF. Si on prend le coût unitaire de 3.25 points de fonction pour développer une nouvelle application, cette alternative de développer 151 PF coûterait 490 jours (151 PF x 3.25 jours/PF = 490 jours). À partir de cette perspective l'exemple ci-dessus identifie un avantage économique de 280 jours personnes (490 jours – 210 jours) pour un projet d'ajouts fonctionnels sur son alternative un projet à développer à partir de zéro.

Tableau D.3

Analyse des résultats

Concept	Type de projet	Coût unitaire	Taille en point de fonction	Coût total (jours)
Sans tenir compte de la mesure de la réutilisation (mesure traditionnelle de la méthode de points de fonction)	Nouvelle application	3.25 jours/P.F.	75	244
	Maintenance	2.80 jours/P.F.		210
En tenant compte de la mesure de la réutilisation (Mesure proposée qui prend en compte les transactions à ne pas développer)	Maintenance	1.39 jours/P.F.	151	210
	Nouvelle application	3.25 jours/P.F.		490

Scénario qui utilise les données tirées d'ISBSG (1998)

- Base de données historique de points de fonction à utiliser : ISBSG (1998)
- Besoin de l'utilisateur : Ajout de 75 PF à une application de 50000 PF
- Durée en jours personnes : 3.25 jours par PF dans les nouvelles applications
2.80 jours par PF dans les projets de maintenance
- Coût pour les nouvelles applications : $75 \text{ PF} \times 3.25 \text{ jours} / \text{PF} = 244 \text{ jours}$
- Coût pour les applications de maintenance : $75 \text{ PF} \times 2.80 \text{ jours} / \text{PF} = 210 \text{ jours}$
- Application de la méthode de mesure de la réutilisation fonctionnelle

$$\begin{aligned} \text{PF}^{\text{E}} (\text{Indicateur}) &= \text{PF}^{\text{E}} \times \text{PF}_{\text{GDE}}^{\text{E}} (\text{ratio}) \\ &= 75 \times 15 \\ &= 11 \text{ PF}_{\text{GDE}}^{\text{E}} \end{aligned}$$

$$\begin{aligned} \text{PF}^{\text{R}} (\text{Ratio prédicteur}) &= \text{PF}_{\text{GDE}}^{\text{N}} + \text{PF}_{\text{ENT}}^{\text{N}} + \text{PF}_{\text{SOR}}^{\text{N}} + \text{PF}_{\text{INT}}^{\text{N}} + / \text{PF}_{\text{GDI}}^{\text{N}} \\ &= 3.48 \end{aligned}$$

$$\begin{aligned} \text{PF}^{\text{ER}} &= \text{PF}^{\text{E}} + \text{Transactions évitées} \\ &= \text{PF}^{\text{E}} + (\text{PF}_{\text{GDE}}^{\text{E}} \times \text{PF}^{\text{R}} (\text{Prédicteur})) \\ &= 75 + (11 \times 3.48) \\ &= 113 \text{ PF}^{\text{ER}} \end{aligned}$$

Le tableau D.4 permet d'analyser l'utilisation de la méthode de mesure de la réutilisation fonctionnelle en utilisant les données historiques de points de fonction qui appartiennent à ISBSG (1998). Cette analyse est réalisée par l'entremise d'un modèle de productivité et elle identifie deux points principaux :

- Lorsqu'on ne tient pas compte de la réutilisation, l'addition de fonctions (maintenance) coûte légèrement moins cher (2.8 jours par PF pour la maintenance au lieu de 3.25 jours par PF pour une nouvelle application), et lorsqu'on tient compte de la mesure de la réutilisation fonctionnelle, la différence de coût unitaire 1.85 jours par PF pour la

maintenance (210 jours / 113) est maintenant beaucoup plus grande et toujours par rapport à 3.25 jours par PF pour une nouvelle application.

- L'alternative de développer une nouvelle application qui n'a pas besoin de ré développer les 38 PF évités dans le projet de maintenance pourrait être d'un total de 113 PF. Si on prend le coût unitaire de 3.25 points de fonction pour développer une nouvelle application, cette alternative de développer 113 PF coûterait 367 jours ($113 \text{ PF} \times 3.25 \text{ jours/PF} = 367 \text{ jours}$). À partir de cette perspective l'exemple ci-dessus identifie un avantage économique de 157 jours personnes ($367 \text{ jours} - 210 \text{ jours}$) pour un projet d'ajouts fonctionnels sur son alternative un projet à développer à partir de zéro.

Tableau D.4

Analyse des résultats

Concept	Type de projet	Coût unitaire	Taille en point de fonction	Coût total (jours)
Sans tenir compte de la mesure de la réutilisation (mesure traditionnelle de la méthode de points de fonction)	Nouvelle application	3.25 jours/P.F.	75	244
	Maintenance	2.80 jours/P.F.		210
En tenant compte de la mesure de la réutilisation (Mesure proposée qui prend en compte les transactions à ne pas développer)	Maintenance	1.85 jours/P.F.	113	210
	Nouvelle application	3.25 jours/P.F.		367

Tableau D.5

Sommaire des résultats des différentes transformations de la méthode de mesure de réutilisation pour les différents scénarios de tests appartenant au regroupement des fonctionnalités petites

Scénarios Selon la base de données utilisée	PF (Indicateur)	PF (Prédicteur)	Transactions évitées	Impact dans le calcul du coût unitaire	
				Nouvelle Applicat.	Mainte Nance
Institution gouvernementale de Québec	12	4.05	42	3.25 jp/PF	1.81jp/PF
Institution financière (Valencia et Nguyen-Kim)	10	3	30	3.25 jp/PF	2 jp/PF
Institution financière (Abran)	28	2.74	77	3.25 jp/PF	1.39jp/PF
ISBSG (1998)	11	3.48	38	3.25 jp/PF	1.85jp/PF

Constantes	
Coût0 unitaire	
Développement	Maintenance
3.25 j.p. / P.F.	2.80 j.p. / P.F.

APPENDICE E
L'ANALYSE DE POINTS DE FONCTION
DESCRIPTION SOMMAIRE DE L'ANALYSE
DE POINTS DE FONCTION

L'analyse de points de fonction a été introduite par Albrecht (Albrecht, 1979). Elle se veut une mesure de la taille fonctionnelle d'un système d'information qui n'est pas influencée par les méthodes de développement ni par les techniques utilisées. La taille fonctionnelle d'une application informatique se veut une mesure pour décrire la taille des fonctionnalités demandées par l'utilisateur. Cette méthode de mesure a été publiée par Albrecht et Gaffney en 1983 et c'est « l'International Function Point Users Group (IFPUG) » (IFPUG, 1994) qui à partir de 1986 établit les normes à suivre.

Le principe de cette méthode de mesure repose sur l'identification, en nombre et en complexité, de cinq types de composantes du processus de traitement de l'information. Ces types de composantes sont les entrées, les sorties, les interrogations, les fichiers logiques internes et les fichiers logiques externes. Ces types de composantes forment la taille fonctionnelle d'une application. La taille fonctionnelle dans l'analyse de points de fonction est obtenue en mesurant le logiciel à partir de deux procédures complémentaires.

Les fonctions sont classifiées en deux catégories : les données et les transactions. Deux types de fonction composent la partie données : les fichiers logiques internes ou groupe de données internes (GDI) et les fichiers logiques externes ou groupe de données externes (GDE). Les transactions sont divisées en trois catégories : les entrées, les sorties et les interrogations. Ensuite, on assigne un poids pour chaque fonction.

La deuxième procédure constitue la valeur du facteur d'ajustement. Elle est basée sur la pondération de quatorze caractéristiques du logiciel, qui représentent les fonctionnalités générales de l'application mesurée. Parmi ces caractéristiques, on peut nommer la communication de données, le taux de transactions, la facilité d'installation. Chaque caractéristique générale du système, également appelée facteur d'envergure, possède une description associée qui aide à la détermination de son degré d'influence. Les degrés d'influence fluctuent sur une échelle de zéro à cinq et d'une influence nulle à une influence importante. Les 14 caractéristiques du système sont prises en compte dans le calcul du facteur d'ajustement. Lorsque ce dernier est appliqué, il ajuste le nombre de points de fonction brut de $\pm 35\%$ pour donner le nombre final de points de fonction (IFPUG, 1994). Le résultat du processus de mesure donne alors un nombre total de points de fonction qui est une mesure des fonctionnalités livrées à l'utilisateur.

L'analyse de points de fonction comprend donc, trois phases : le calcul des points de fonction non ajustés, la détermination du facteur d'ajustement par l'entremise des caractéristiques générales du système (tableau E.1) et finalement le calcul des points de fonction ajustés.

D'une manière plus détaillée, le premier pas pour le calcul des points de fonction est l'identification de toutes les composantes : les entrées, les sorties, les interrogations, les fichiers logiques internes et les fichiers logiques externes de l'application informatique candidate à être mesurée. Par la suite, il faut déterminer le niveau de complexité (bas, moyen ou élevé) pour chacune des composantes.

Le niveau de complexité est déterminé de la façon suivante :

- Si on doit déterminer le niveau de complexité pour une entrée, une sortie ou une interrogation, on doit compter le nombre de champs d'entrée ou de sortie de la composante. Ces champs sont appelés éléments de données (ED). En suite on compte le nombre de fichiers logiques internes et externes référencés par cette composante appelée fichiers référencés (FR). La figure E.1 illustre la façon de déterminer cette complexité. Il

faut remarquer qu'une interrogation est formée par une partie entrée et une autre appelée sortie. Alors, pour déterminer la complexité des composantes d'une interrogation, on utilise la matrice des entrées pour la partie entrée et la matrice des sorties pour la partie sortie.

	1 à 4 ED	5 à 15 ED	16 et + ED		1 à 4 ED	5 à 15 ED	16 et + ED
0 ou 1 FR	B	B	M	0 ou 1 FR	B	B	M
2 FR	B	M	E	2 ou 3 FR	B	M	E
3 et + FR	M	E	E	4 et + FR	M	E	E

Entrées **Sorties**

B= Bas M = Moyen E = Élevé ED= Éléments de données FR=Fichiers référencés TE = Types d'enregistrements

Figure E.1

Matrices de complexité pour les entrées et les sorties.

- Pour déterminer la complexité des fichiers logiques internes ou externes, on compte le nombre de champs du fichier, appelés dans le jargon de l'analyse de points de fonction éléments de données (ED) et le nombre d'enregistrements logiques du fichier, appelés types d'enregistrements (TE). La figure E.2 illustre la matrice de complexité pour les fichiers internes et externes.

	1 à 19 ED	20 à 50 ED	51 et + ED		1 à 19 ED	20 à 50 ED	51 et + ED
1 TE	B	B	M	0 ou 1 FR	B	B	M
2 à 5 TE	B	M	E	2 ou 3 FR	B	M	E
6 et + TE	M	E	E	4 et + FR	M	E	E
Fichiers logiques internes				Fichiers logiques externes			

B= Bas M = Moyen E = Élevé ED= Éléments de données FR=Fichiers référencés TE = Types d'enregistrements

Figure E.2

Matrices de complexité pour les fichiers logiques internes et externes.

- Le troisième pas consiste à attribuer un poids pour les différentes composantes et selon leurs niveaux de complexité. Ensuite on fait l'addition de ces poids en utilisant la grille présentée dans le tableau E.1. De cette façon, on obtient le total de points de fonction non ajustés (PfnA).

Tableau E.1

Pondération des composantes

Type de composante	Niveau de complexité			Total
	Bas	Moyen	Élevé	
Entrées	_ * x 3	__ x 4	___ x 6	___
Sorties	__ x 4	___ x 5	___ x 7	___
Interrogations	__ x 3	___ x 4	___ x 6	___
Fichiers logiques internes	__ x 7	___ x 10	___ x 15	___

Fichiers logiques externes	__ x 5	__ x 7	__ x 10	__
Total de points de fonction non ajustés (PfnA)				__

* Nombre de composantes

- Le facteur d'ajustement

Le facteur d'ajustement est le fait d'évaluer l'impact que les 14 caractéristiques générales d'un système déterminées dans l'analyse de points de fonction peuvent avoir sur la taille fonctionnelle de l'application mesurée. Chacune de ces caractéristiques est notée de 0 à 5 en fonction de son degré d'influence. Le tableau E.2 illustre les 14 caractéristiques générales du système.

Le facteur d'ajustement (FA) se calcule en appliquant la formule suivante :

$$FA = 0.65 + (TDI \times 0.01)$$

Où TDI = Total des degrés d'influence.

Ce facteur d'ajustement peut varier dans la plage de 0.65 à 1.35, c'est-à-dire qu'il peut avoir un impact de $\pm 35\%$ sur le comptage final de l'application..

- La dernière étape du processus de mesure d'une application consiste à multiplier le nombre de points de fonction non ajustés par le facteur d'ajustement. De cette manière on obtient le total de points de fonction ajustés (PFA) ou total de l'application mesurée. L'équation suivante illustre ce total.

$$PFA = PFnA \times FA$$

Où :

PFA = Points de fonction ajustés.

PFnA = Points de fonction non ajustés

FA = Facteur d'ajustement

Tableau E.2

Les 14 caractéristiques générales du système tel que déterminé
par l'analyse de points de fonction

Caractéristiques générales	Degré d'influence (DI)
1. Télécommunications	* _____
2. Distribution des traitements	_____
3. Performance	_____
4. Charge sur la configuration	_____
5. Taux de transactions	_____
6. Saisi interactif des données	_____
7. Convivialité	_____
8. Mise à jour en temps réel	_____
9. Complexité interne du traitement	_____
10. Réutilisation de l'application	_____
11. Facilité d'installation	_____
12. Facilité d'exploitation	_____
13. Plusieurs sites	_____
14. Facilité de modification	_____
Total des degrés d'influence (TDI)	_____

*** Pondération des
degrés d'influence :**

0 - Pas d'influence
1 - Insignifiante
2 - Modérée
3 - Moyenne
4 - Forte
5 - Très forte

L'analyse de points de fonction a évolué à partir du modèle présenté par Albrecht en 1979 jusqu'à date. À partir de l'automne 1984 c'est l'International Function Point Users Group (IFPUG) qui a pour mandat de diffuser la technique des points de fonction, d'en coordonner l'évolution, de suivre l'implantation dans l'industrie ainsi que de promouvoir l'homogénéité de l'application des règles. Le tableau E.3 présente d'une façon générale le modèle de l'analyse de points de fonction.

Tableau E.3

Modèle de mesure de l'analyse de points de fonction

Composantes	Complexité		
	Bas	Moyen	Élevé
Entrées	3	4	6
Sorties	4	5	7
Interrogations	3	4	6
Fichiers internes	7	10	15
Fichiers externes	5	7	10

 \times

Facteur d'ajustement
14 caractéristiques générales du système

 $=$

Total de points de fonction (PF ajustés)
$\pm 35 \%$

APPENDICE F

**ARTICLE « MEASUREMENT OF FUNCTIONAL REUSE IN
MAINTENANCE »**

(ABRAN ET DESHARNAIS, 1995)

BIBLIOGRAPHIE

- Abran, A. 1994. « Analyse du processus de mesure des points de fonction ». Thèse de doctorat. École Polytechnique, Université de Montréal.
- Abran, A. et J.-M. Desharnais. 1995. « Measurement of Functional Reuse in Maintenance ». *Software Maintenance : Research and Practice*, Vol. 7, p. 263-277.
- Albrecht, A.J. et J.E. Gaffney. 1983. « Software Function, Source Lines of Code and Development Effort Prediction : A Software Science Validation ». *IEEE Transactions on Software Engineering*, SE-9(6) Nov. p. 639-648.
- Abran, A. et H. Nguyen-kim. 1993. « Measurement of the Maintenance Process from a Demand-based Perspective ». *Software Maintenance : Research and Practice*, Vol. 5, p. 63-69.
- Abran, A. et H. Nguyen-kim. 1991. « Analysis of Maintenance Work Categories Through Measurements ». *Proceeding of the IEEE Conference of Software Maintenance*, October, p. 104-113.
- Abran, A et P.N. Robillard. 1990. « Software management based on software deliverables ». *In Proceedings CIPS/CATA Congress 90, Canadian Information Processing Society, Ottawa, Ontario, Canada*, p. 237-245.
- Abran, A. et Z. Taboubi. 1994. « Validation de données pour le modèle de points de fonction spécifique à la maintenance adaptative ». *Technical report, Université du Québec à Montréal*, Novembre.
- Albrecht, A.J. 1979. « Measuring Application Development Productivity ». *Proceeding of the Joint SHARE, GUIDE and IBM Application Development Symposium*, Monterey, CA, October, p. 83-92.
- Arthur, L.J. 1988. « *Software Evolution – The Software Maintenance Challenge* ». John Wiley & Sons.
- Balda, D.M. et D.A. Gustafson. 1990. « Cost-estimation models for the reuse and prototype software development ». *ACM SIGSOFT*, July, p. 42-50.
- Barnes, B.H. et T.B. Bollinger. 1991. « Making reuse cost-effective ». *IEEE Software*, Vol. 8, no 1, Jan., p. 13-24.

- Barnes, B., T. Durek, J. Gaffney et A. Pyster. 1987. « A framework and economic foundation for software reuse ». *Proceeding Workshop Software Reusability and Maintainability*.
- Basili, V.R. et al. 1992. « The software engineering laboratory-an operational software experience ». *Proceeding of the 14th International Conference Software Engineering*, Melbourne, Australia, May 11-15, p. 370-381.
- Basili, V.R. et J.D. Musa. 1991. « The future Engineering of Software : A Management Perspective ». *IEEE Computer*, Sept, Vol. 24, no 9, p. 90-96.
- Basili, V.R. et Phillips, T.Y. 1981. « Evaluating and Comparing Software Metrics in the Software Engineering Laboratory ». *ACM SIGMetrics*, Vol. 10, p. 95-106.
- Basili, V.R., R.W. Selby et D.H. Hutchens. 1986. « Experimentation in Software Engineering ». *IEEE Transactions on Software Engineering*, Vol. SE-12, no 7, July.
- Biggerstaff, T.J. et A.J. Perlis. 1984. « Reusability Framework, Assessment, and Directions ». *IEEE Software*, July, p. 474-476.
- Boehm, B.W. 1989. « Software Engineering : R and D Trends and Defense Needs. Research Directions in Software Technology ». P. Wegner (Ed) The MIT Press, Cambridge, MA. p 44-86.
- Boehm, B.W. 1987. « Improving Software Productivity ». *IEEE Computer*, September, p. 43-57.
- Boehm, B.W. 1988. « Understanding and Controlling Software Cost ». *IEEE Transactions on Software Engineering*, Vol. 14, no 10, October, p. 1462-1477.
- Boehm, B.W. 1981. *Software Engineering Economics*. Prentice-Hall Inc., Englewood Cliffs, N.J.
- Chidamber, S.R. et C.F. Kemerer. 1971. « Towards a metrics suite for object oriented design ». In *Proceedings of 6th ACM Conference Object Oriented Programming Systems, Languages And applications (OOPSLA)*, Phoenix, AZ, p.197-211.
- Davis J.S. et R.J. LeBlanc. « A Study of the Applicability of Complexity Measures ». *IEEE Transactions Software Engineering*, Vol. 14, p. 1366-1372.
- Dreger, J.B. 1989. *Function Points Analysis*. Prentice Hall, Englewood Cliffs, N.J.
- Dunham, J.R., E. Kruesi. 1983. « The Measurement Task Area ». *IEEE Computer*, November.

- Fetcke, T., A. Abran et T.H. Nguyen. 1996. « Mapping the OO-Jacobson Approach into Function Point Analysis ». *From Tools USA'97 (Technology of Object-Oriented Languages and Systems)*.
- Freeman, P. 1983. « Reusable software engineering : Concepts and research directions ». *In Workshop on Reusability in Programming*, September.
- Gaffney, J.E. et R.D. Cruickshank. 1992. « A general economics model of software economics model of software reuse ». *Proceedings of 14th International Conference on Software Engineering, ACM Press, New York*, p. 327-337.
- Grady, R.B. 1990. « Work-Product Analysis. The Philosopher's Stone of Software ». *IEEE Software*, March.
- Grady, R. B. 1987. « Measuring and Managing Software Maintenance ». *IEEE Software*, Vol. 4, no 9, September, p. 35-45.
- Grady, R. et D.L. Caswell. 1987. *Software Metrics : Establishing a Company*. Wide Program, Prentice Hall.
- Gordon, R.D. 1979. Measuring Improvements in Program Clarity. *IEEE Transactions Software Engineering*, Vol. 5, p. 79-90.
- Halstead, M.H. 1977. *Elements of Software Science*, Elsevier North-Holland, New York.
- Han, W.T., Y.C. Choe et Park, Y.J. 1987 « Software Metrics Using Operand Type ». *In Proceeding, TENCON'87, Seoul*, p. 1212-1215.
- Harrison, R. 1987. « Maintenance giant sleeps undisturbed in Federal data centers ». *ComputerWorld*, Vol. 21, no 10, p.81-86.
- Henry S. et D. Kafura. 1981. « Software Structure Metrics Based on Information Flow ». *IEEE Transaction Software Engineering*, Vol. 7, p. 510-518.
- Horowitz E. et J.B. Munson. 1984. « An Expansive View of Reusable Software ». *IEEE Transactions on Software Engineering*, Vol. 10, no 5, p. 477-487.
- IFPUG. 1994. « IFPUG Function Point Counting Practice Manual Release 4.0 », International Function Points Users Group, Westerville, OH, 293 p.
- ISBSG. 1998. « Worldwide Software Development. The Benchmark Release 5 ». International Software Benchmarking Standards Group Ltd , March.

- Jacquet, J. P. et A. Abran. 1996. « ISO Verification of a functional size measurement method (Part A) » DRAFT.
- Jacquet, J.P. et A. Abran. 1997. « A Structured Analysis of the Proposed ISO Standard on : Functional Size Measurement – Definition of Concepts (ISO/IEC 14143-1) » DRAFT, April.
- Jacquet, J.P. et A. Abran. 1997 « From Software Metrics to Software Measurement Methods : A Process Model ». *Third International Symposium and Forum on Software Engineering Standards ISESS'97*, Walnut Creek (CA), June.
- Jones, C. 1996. « Conflict and Litigation Between Clients and Developers ». Software Productivity Research Inc.
- Jones, C. 1986. *Programming productivity*. McGraw-Hill Inc., New York, NY.
- Kearney, J.L. Sedlmeyer, W.B. Thompson, M.A. Gray et M.A. Adler. 1986. « Software Complexity Measurements ». *Communication of the ACM*, Vol. 29, no 11, November.
- Kemerer, C.F. 1993. « Reliability of Function Points Measurement – A Field Experiment ». *Communications of the ACM*, Vol. 36, no 2, February, p. 85-97.
- Kim, Y. et E.A. Stohrrd. 1992. « Software reuse : issues and research directions ». *Proceeding of the 25th Hawaii International Conference on Systems Sciences*, Vol. 3, IEEE Computer Society Press, Los Alamitos, CA, p. 612-623.
- Krueger, C.W. 1992. « Software Reuse ». *ACM Computing Surveys*, Vol. 24, no 2, June, p. 131-183.
- Lanergan, R.G. et C.A. Grasso. 1984. « Software Engineering with Reusable Design and Code ». *IEEE Transactions on Software Engineering*, Vol. Se-10, no 5, Sept, p. 498-501.
- Lim, W.C. 1994. « Effects of Reuse on Quality, Productivity and Economics ». *IEEE Software*, Vol. 11, no 5, Sept., p. 23-30.
- Lientz, B.P. et B.E. Swanson. 1980. *Software Maintenance Management*. Addison-Wesley, Reading, MA, 214 p.
- Lister, A.M. 1982. « Software Sciences-The Emperor's New Clothes? », *Australian Computer Journal*, Vol. 14, p. 66-70.
- McCabe, T.I. 1976. « A Complexity Measure ». *IEEE Transactions on software Engineering*, Vol. SE-2, December, p. 308-320.

- McDonell, S.G. 1991. « Rigor in Software Complexity Measurement Experimentation ». *Journal of Systems Software*, Vol. 16, no 2, October.
- Molinié, L. 1997. « Une approche de gestion économique de la maintenance des logiciels dans un environnement d'impartition : une étude exploratoire », *Activité de Synthèse M.Sc.* Montréal, Université du Québec à Montréal.
- Morris, P.M et J.-M. Desharnais. 1996. « Function Point Analysis. Validating the Result ». *IFPUG Spring Conference*, Atlanta, April.
- Navlaka, J. 1986. « Software Productivity Metrics : Some candidates and their evaluation ». *Proceeding of National Computer Conference*, p. 69-75.
- Page-Jones, M. 1988. *The Practical Guide to Structured Systems*. Second Edition, Yourdon Press.
- Poulin, J.S. 1994. « Measuring Software reusability /rd ». *Software Conference*, Rio, IEEE.
- Poulin, J.S., J.M. Caruso et D.R. Hancock. 1993. « The business case for software reuse ». *IBM Systems Journal*, Vol. 32, no 4, p. 567-594.
- Prater, R.E. 1984. « An axiomatic theory of software complexity measures ». *Computer Journal*, Vol. 27, p. 340-346.
- Pressman, R.S. et S.R. Herron. 1991. *Software Shock*. Dorset House, New York.
- Ramsey, C.L. et V.R. Basili. 1989. « An Evaluation of Expert Systems for Software Engineering Management ». *IEEE Transaction on Software Engineering*, Vol. SE-15, no 6, Juin, p. 747-759.
- Rault, J.C. 1979. « An Approach Towards Reliable Software ». *In Proceedings 4th International Conference on Software Engineering*, Munich, p. 220-230.
- Roberts, F.S. et C.H. Franke. 1976. « On the Theory of Uniqueness in Measurement ». *Journal of Mathematical Psychology*, Vol.14, p 211-218.
- Rubin, H.A. 1987. « Critical Success Factors for Measurement Programs ». *Proceeding Spring Conference of IFPUG, International Function Point Users Group*, Scottsdale, Arizona.
- St-Pierre et al. 1997. « Adapting Function Points to Real-Time Software ». *IFPUG 1997 Fall Conference, International Function Point Users Group*, Scottsdale, Arizona.
- Schach, S.R. 1993. *Software Engineering*. Second Edition, IRWIN, Burr Ridge, Illinois.

- Schneidewind, N.F. 1983. « A Complexity Metric which Integrates Structural and Textual Metrics ». *Proceedings of the Second Annual Phoenix Conference on Computers and Communications*, Phoenix, AZ, New York, IEEE, March, p. 95-98.
- Shepperd, M. et D. Ince. 1993. « Derivation and validation of software metrics ». Oxford Science Publications.
- Shepperd, M. et J.L. Lassez. 1980. « Opposing Views on the Use of Software Science Measures for Automatic Assessment of Student Programs ». *Australian Computer Science Communications*, Vol. 2, p. 205-215.
- Suppes, P. et J.L. Zinnes. 1963. « Basic Measurement Theory, Handbook of Mathematical Psychology ». Edited by D.D. Luce, R.R. Bush, E. Galanter, Vol. I, Wiley, New York, p.1-76.
- Tracz, W. 1987. « Software Reuse : Motivators and Inhibitors ». *In Proceedings of COMPCONS'87*, IEEE Press, p. 358-363.
- Verner, J.M. et G. Tate. 1985. « A Software Size Model ». *IEEE Transactions on Software Engineering*, Vol. 18, no 4, April, p. 265-278.
- Verner, J.M. et G. Tate. 1985. « Software Sizing and Costing ». *In Proceeding, 9th New Zealand Computer Conference*, Auckland, N.Z., p. 220-230.
- Weiss, D.M. et Basili, V.R. 1985. « Evaluating software development by analysis of Changes : Some Data from the Software Engineering Laboratory ». *IEEE Transactions on Software Engineering*, Vol. SE-11, no 2, February, p. 157-168.
- Weyuker, E. 1988. « Evaluating software complexity measures ». *IEEE Transaction Software Engineering*, Vol. 14, p. 1357-1365.
- Woodfield, S.N., V.Y. Shen et H.E. Dunsmore. 1981. « A study of Several Metrics for Programming Effort ». *Journal of Systems and Software*, Vol. 2, p. 97-103.
- Yau, S.S. et Collofello. 1980. « Some Stability Measures for Software Maintenance ». *IEEE Transactions on Software Engineering*, Vol. 6, p 545-552.
- Zuse, H. 1996. « A Framework of Software Measurement ». Berlin, October.
- Zuse, Horst. 1994. « Foundations of validation, prediction and software measure ». *Proceeding of the AOSW94 (Annual Oregon Software Metric Workshop)*, Portland, April, p. 20-22.