

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

**DÉPARTEMENT D'INFORMATIQUE
MAÎTRISE EN INFORMATIQUE DE GESTION**

**PROPOSITION D'ACTIVITÉ DE SYNTHÈSE
PRÉSENTÉE COMME EXIGENCE PARTIELLE DE LA
MAÎTRISE EN INFORMATIQUE DE GESTION**

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

**PAR
RAFAEL VALENCIA**

DIRECTEUR DE RECHERCHE : ALAIN ABRAN

AOÛT 1997

Table des matières

Liste de figures	iv
Liste des tableaux	v
Résumé	vi
Introduction	1
Chapitre I: Problématique	3
Chapitre II: Contexte théorique	5
2.1.-Évaluation de la méthode proposée.....	6
2.1.1 Structure du guide de validation	7
2.1.2 Cycle de vie d'une méthode de mesure	7
2.1.3 Vérification de la méthode proposée et le guide de validation	8
2.2 –Une proposition de mesure de la réutilisation fonctionnelle	11
2.3 La réutilisation des composants logiciels	15
2.3.1 Introduction	15
2.3.2 Création des composants logiciels afin de les réutiliser	16
2.3.3 Réutiliser les composantes déjà existantes	16
2.4 La mesure utilisée dans cette approche	16
2.4.1 Informations requises aux mesures des points de fonction	18

2.4.1.1	Généralités	18
2.4.1.2	Documents nécessaires pour mesurer les points de fonction	18
2.4.2.-	Élaboration de bases de données historiques de points de fonction	19
2.5	Pertinence de la métrique du logiciel	21
2.5.1	Élaboration de modèles de productivité ou d'estimation	22
2.5.2	Bénéfices potentiels d'utilisation de la méthode proposée	22
Chapitre III:	Démarche méthodologique	24
3.1	Description du cadre d'expérimentation de Basili	24
Chapitre 4	Définition du projet de recherche	28
4.1.-	Motivation, objet et propos et le domaine de la recherche	28
4.2	Objectifs et limites de la recherche.	29
4.3	Les utilisateurs.-	30
Chapitre V	Planification du projet de recherche	31
5.1	Les étapes du projet, intrants et livrables	31
5.2	Contenu des livrables	33
5.2.1-	Justification de l'approche d'estimation	33
5.2.2-	Opportunités de l'utilisation de la méthode proposée	33
5.2.3.-	Élaboration de bases historiques de points de fonction	34

5.2.3.1	Préparation
.....	34
5.2.4 Élaboration de scénarios de test de la méthode proposée
35	
5.3.-L'échéancier	prévu
.....	36
Appendice A 38
Bibliographie	
.....	42

Liste des figures

Figure	Page
Figure 2.1 Processus de mesure	7
Figure 2.2 Cycle de vie d'une méthode de mesure	8

Liste des tableaux

Tableau	Page
Tableau 3.1 Cadre de recherche de Basili	24
Tableau 3.2 Activités de la phase 1 du cadre de Basili	26
Tableau 3.3 Activités de la phase 2 du cadre de Basili	27
Tableau 5.1 Les étapes, les intrants et les livrables du projet de recherche	32
Tableau 5.2 Opportunité d'utilisation de la méthode proposée	34
Tableau 5.3 Effort et dates estimées de réalisation du projet	37
Tableau A.1 Distribution moyenne de points de fonction	38
Tableau A.2 Étude de cas. Mesure de la réutilisation fonctionnelle avec et sans réutilisation	40

Résumé

L'article *Measurement of Functional Reuse in Maintenance* (Abran et Desharnais, 1995) identifie et mesure la réutilisation des composants logiciels dans les projets où des fonctionnalités ont été ajoutées aux applications existantes. L'approche proposée est basée sur la mesure de la réutilisation à partir d'une perspective fonctionnelle plutôt que technique.

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 la méthode de points de fonction et de la notion « black box reuse ». Le ratio prédicteur est dérivé à partir de deux concepts. Le premier concept est basé sur la compréhension de la notion « coûts évités si on réutilise au lieu de développer ». Le deuxième concept présente la manière comment la notion de « coûts évités » peut être capturée à partir de bases de données historiques de points de fonction. Ces bases de données ont été collectées dans plusieurs industries et elles sont classifiées selon le domaine d'application informatique.

L'article indique la manière de combiner le ratio prédicteur et l'indicateur de réutilisation, afin de dériver une mesure de taille fonctionnelle alternative qui prend en compte les composants réutilisables, qui ne doivent pas être redéveloppés.

L'article démontre aussi comment ces ratios peuvent être intégrés dans un modèle de productivité afin d'analyser les bénéfices de la réutilisation. Dans la dernière partie de l'article, une étude de cas basé sur un ensemble de données industriels est présentée afin d'illustrer la mesure de la réutilisation fonctionnelle et leur impact dans l'analyse de la productivité dans les différents projets de maintenance.

Dans ce projet de recherche, on propose d'expérimenter et tester ce modèle avec d'autres ensembles de données industrielles. Pour ce faire, on utilisera des données recueillies dans deux organisations financières canadiennes et un organisme gouvernemental du Québec.

Les objectifs de cette recherche sont au nombre de trois:

- Expérimenter le modèle existant à haut niveau
- Vérifier la plausibilité, la complétude, la robustesse et la fiabilité du modèle proposé.
- Identifier les limitations du modèle.

Reliés aux objectifs, les livrables finaux du travail seront:

- Un cadre expérimental et de vérification du modèle.
- Illustration des cas très précis où le modèle ne s'applique pas, s'il y a lieu.

Dans l'élaboration de la présente étude, le cadre expérimental de Basili est utilisé (Basili,1986).

La durée totale prévue du projet est de 700 heures/personne. Sa réalisation sera accomplie dans une période approximative de sept mois, entre juin et décembre 1997.

Mots clés: informatique, mesure, réutilisation, maintenance du logiciel, 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 regardent l'approche de la réutilisation comme une voie de 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 contrôler 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 viabilité de la méthode décrite dans l'article "Measurement of Functionnal Reuse in Maintenance" (Abran et Desharnais, 1995). Les concepts de cette méthode seront analysés et testés avec des bases de données historiques de points de fonction provenant des deux institutions financières canadiennes et d'un organisme gouvernemental de Québec. L'utilisation de ces bases de données permettra d'éclaircir et déterminer les limites du modèle.

Dans ce document, nous présentons d'abord les éléments qui composent la problématique de la présente étude et l'intérêt qu'elle suscite dans la recherche d'une solution au budget très élevé consacré au développement et à la maintenance des applications informatiques.

Le deuxième chapitre porte sur la revue de la littérature. Les références sont regroupées de manière qu'elles soient sous-jacentes à la problématique de cette étude.

Le troisième chapitre présente la démarche méthodologique qui guidera la mise en œuvre de ce projet de recherche. Cette démarche est basée sur le cadre d'analyse des travaux dans le domaine du génie logiciel proposé par Basili et al (1986). Ce cadre propose la réalisation de la recherche en quatre phases. Les deux premières composent ce document de proposition (les

phases de définition et de planification). Les deux autres phases (l'exécution et l'interprétation) seront contenues dans le rapport final de cette activité de synthèse.

Dans le quatrième chapitre, la définition du projet de recherche est présentée. Nous formulons 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.

Dans le dernier chapitre, nous formulerons la phase de la planification. Nous identifierons les différentes étapes du projet ainsi que leurs intrants et leurs extrants. Une description sommaire des livrables est présentée dans ce chapitre, ainsi que l'échéancier prévu pour la réalisation de ce projet de recherche.

Chapitre I

Problématique

Les ressources dépensées dans le développement ou dans la maintenance des applications informatiques, particulièrement en termes d'effort humain (traduit directement en coût) est une préoccupation majeure dans la gestion de projets informatiques. Étant donné le rôle fondamental que le développement logiciel joue dans la livraison et la maintenance des applications informatiques, les gestionnaires se doivent d'améliorer le processus de développement logiciel. Un des moyens très importants du processus d'amélioration dans une organisation, est le fait de pouvoir mesurer ce même processus. Chidamber et Kemerer (1994) ont souligné la demande croissante de mesures ou de métriques logiciels afin de gérer le processus du développement logiciel. Cette dernière voie de recherche sera l'objet de la présente étude.

Dans la littérature, plusieurs auteurs (Poulin, Caruso et Hancock, 1993), présentent la réutilisation de composants comme la clé pour faire des économies et des bénéfices dans le développement du logiciel. Cependant peu d'études ont analysé comment mesurer cette réutilisation du point de vue fonctionnel. Gaffney et Cruickshank (1992) présentent un modèle économétrique de la réutilisation, modèle dérivé à partir du coût d'intégrer les lignes de code réutilisées. Ce modèle n'adresse pas le fait d'ajouter du nouveau code ou la modification de vieux code dans les vieilles applications. En plus, cette approche ne permet pas d'identifier, d'une manière formelle, les fonctionnalités réutilisées.

Afin de palier à certaines faiblesses du modèle de Gaffney et al., un modèle pour mesurer la réutilisation des composants logiciels du point de vue fonctionnel a été proposé par Abran et Desharnais(1995). Ce modèle permet de tenir compte des ajouts ou modifications de fonctionnalités aux applications existantes. Cette problématique de recherche a été suggérée par Kim et Stohr (1992) et elle se veut une alternative à l'approche proposée par Gaffney et Cruickshank (1992). Ce modèle pourrait permettre de mieux contrôler le budget alloué aux activités où des ajouts ou des modifications fonctionnelles ont été faits. Néanmoins, il n'a pas été validé dans un contexte différent de celui de son design initial.

Est-ce que le modèle de mesurer la réutilisation fonctionnelle tel que proposé par Abran et Desharnais (1995) et qui a été développé avec un seul contexte expérimental peut être généralisé à d'autres contextes ?

La problématique principale dans la présente étude, concerne la vérification du modèle de mesure de la réutilisation fonctionnelle tel que proposé par Abran et Desharnais (1995), avec d'autres ensembles de données expérimentales. Ceci permettra de vérifier le modèle de mesure de composants réutilisables ainsi que les bénéfices de leur utilisation.

La problématique se rattache à l'objectif principal, car il existe déjà un scénario de travail sur lequel s'appuyer. La cohérence est cimentée par l'établissement d'un rapport solide entre le prototype théorique qui se prête bien à être vérifié par l'entremise de bases de données de points de fonction avec plusieurs caractéristiques et complexités diverses.

La présente étude va tester le modèle à partir de deux nouvelles bases historiques de points de fonction. L'utilisation de ces dernières permettra de mieux évaluer le comportement de ce modèle et son adaptation à différents contextes de travail. La contribution principale de la présente étude est la vérification empirique du modèle de mesure de la réutilisation fonctionnelle dans les projets informatiques proposé par Abran et Desharnais (1995).

Chapitre II

Contexte théorique

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

Cette revue de littérature, est composée de quatre sections qui donneront la base fondamentale à la présente étude. En premier lieu, on décrira certaines notions concernant le concept de réutilisation et leur impact sur le développement informatique et la maintenance. Dans la deuxième section, on évaluera la méthode proposée par rapport aux critères de vérification établies 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 sur 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, on présentera une méthode de collecte de données industrielles mesurables, qui nous permettra d'illustrer le comportement du modèle proposé, 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 mesurables nous permettront d'illustrer le comportement du modèle proposé, à partir de contextes industriels. Finalement, on analysera la pertinence du modèle de réutilisation fonctionnelle, leur utilisation dans les analyses de productivité et d'estimation, ainsi que les 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 le modèle devrait être utilisé et dans quelles directions pourraient s'orienter les travaux futurs.

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 mesures logiciels proposées. Dans la section suivante on réalisera une évaluation de la méthode proposée par Abran et al., pour la mesure de réutilisation fonctionnelle, par rapport au guide proposé par Jacquet et Abran (1996).

2.1 Vérification de la méthode proposée

Cette section se veut une vérification de la méthode proposée par Abran et Desharnais, par rapport au guide de vérification des méthodes de mesure de taille fonctionnelle proposé par Jacquet et Abran (1996). Ce guide a pour objet de fournir une méthode de vérification, aussi bien théorique que pratique, des méthodes de mesure de taille fonctionnelle candidates à la norme ISO. 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.

2.1.1 Structure du guide de validation

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 sera ensuite exploité dans différents modèles. La figure 2.1 permet de mieux comprendre la logique à partir de laquelle le processus de mesure est construit

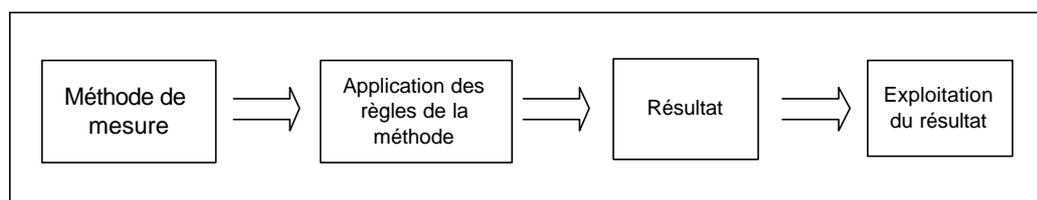


Figure 2.1
Processus de mesure

2.1.2 Cycle de vie d'une méthode de mesure

Le guide proposé par Jacquet et Abran (1997) 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. La figure 2.2 présente les différentes phases qui composent le cycle de vie d'une méthode de mesure.

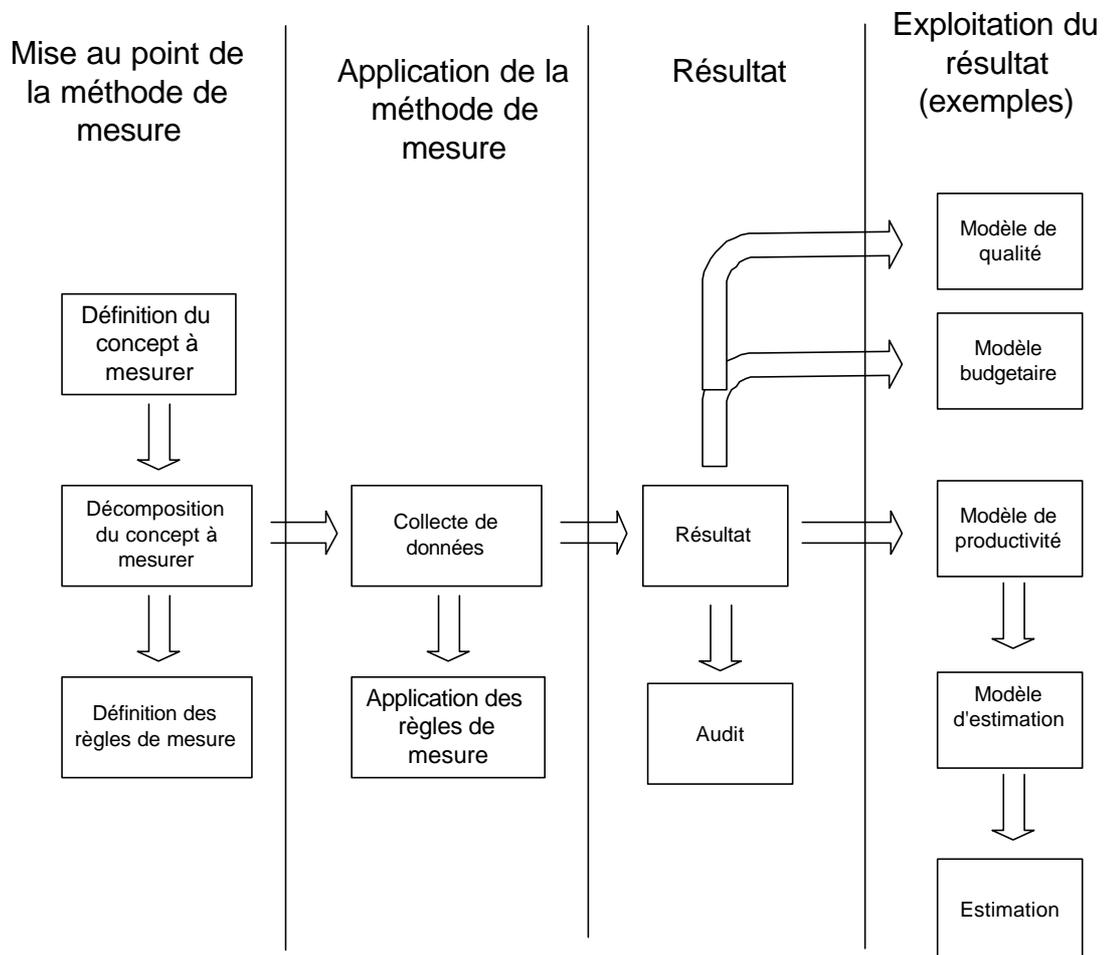


Figure 2.2

Cycle de vie d'une méthode de mesure

2.1.3 Vérification de la méthode proposée et le guide de validation

Dans cette partie on vérifiera la méthode de mesure proposée par Abran et Desharnais(1995). par rapport aux différentes phases du cycle de vie d'une mesure proposé par Jacquet et Abran (1996). Cette vérification sera faite en conformité aux divers points de contrôle qui composent le guide de vérification.

La première phase du cycle de vie d'une méthode de mesure appelé « Mise au point de la méthode de mesure », est composée de trois parties. La première est appelée la définition du concept à mesure. La méthode proposée par Abran et Desharnais (1995) se conforme à cette première partie, car elle est bâtie à partir d'une idée précise du concept à mesurer et du domaine dans lequel s'effectuera la mesure, ainsi que la direction qui 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 décomposition du concept à mesurer doit permettre de décomposer le concept à mesurer en sous-concepts. Ceux-ci doivent à son tour être décomposés jusqu'à ce que la méthode puisse se baser sur des observations directement mesurables. La méthode proposée par Abran et Desharnais, adhère à cette partie, car elle assure que la fonctionnalité (la décomposition du concept à mesurer) sera cohérente par rapport au concept et aussi avec 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 de mesure 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 (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 deux parties : la collecte de données et l'application des règles de mesure. Dans ces deux parties, le guide est focalisé surtout sur la validation de ces deux 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 de 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 quatre 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 simplement 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 proposée par Abran et al., 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 invoque aux questions ci-haut? Premièrement cette méthode renvoie toujours la même valeur quand une même application informatique est mesurée par des personnes de compétences, cultures équivalentes. Deuxièmement, l'application répétée de cette méthode sur une même application renvoie toujours les même valeurs, si elle est mesurée par des personnes de compétences, cultures différentes. Ensuite, elle produit le même résultat quand elle est appliquée soit pour un même contexte, soit dans des contextes différents.

La quatrième phase sera développée à partir de l'analyse des différents résultats de la présente étude.

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), présente une stratégie qui identifie et mesure d'une part les transactions qui ne doivent pas être redéveloppées et d'autre part, elle quantifie les bénéfices dérivés à partir du coût qui est évité pour ne pas redévelopper 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 permettra 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 la méthode de points de fonction 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 concepts. Les deux premiers concepts font partie de la méthode de points de fonction, tandis que le troisième fait partie de la notion « black box reuse ». Le premier concept est l'identification de la frontière du comptage. D'après le guide de comptage de points de fonction, 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. Le deuxième concept est appelé 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. Le troisième concept 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 présente la manière dont la notion « coûts évités si on réutilise au lieu de développer », peut être capturée à partir de bases de données historiques de points de fonction. Ces bases de données ont été collectées dans plusieurs industries et elles sont classifiées selon le domaine d'application informatique. Cette information peut être intégrée dans des modèles de productivité par l'entremise de la combinaison de l'indicateur de réutilisation et du ratio prédictif de la base de données référencée. Ces bases de données sont le résultat du nombre total de points de fonction de plusieurs applications et transformé en pourcentage au total de points de fonction pour chaque composant de la méthode de points de fonction. Cette méthode identifie deux catégories de types de composants relatifs aux données et aux transactions. Les types de composants relatifs aux données représentent les fonctionnalités fournies à l'utilisateur pour lui permettre d'accéder aux données externes et internes. Ils concernent aussi bien des groupes logiques de données internes (GDI) que des groupes logiques de données externes (GDE). Les types de composants relatifs aux transactions représentent les fonctionnalités fournies à l'utilisateur à des fins de traitement des données dans une application. Ils sont définis en termes d'entrées (par exemple un écran de saisie de données), de sorties (par exemple un rapport) et d'interrogations (par exemple un écran de saisie qui est une combinaison d'entrées (requête) et de sorties (extraction)).

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

$$\text{Indice de réutilisation} = \text{FP}(\text{GDE})$$

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

$$\text{Ratio prédicteur de réutilisation} = \frac{\text{PF}(\text{GDE}) + \text{PF}(\text{ENTRÉES}) + \text{PF}(\text{SORTIES}) + \text{PF}(\text{INTERROGAT.})}{\text{PF}(\text{GDI})}$$

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-hauts

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

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

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

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

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

Formule 8

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

À titre d'illustration on présent en annexe A, un cas industriel qui démontre les bénéfices de cette méthode de mesure de la réutilisation fonctionnelle.

2.3 La réutilisation des composants logiciels

2.3.1 Introduction

La réutilisation logicielle consiste d'une manière simple, 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'implémentation, 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'atteint d'une meilleure qualité des logiciels (Tracz, W., 1987). Le point de vue technique présente deux approches de réutilisation. Le premier est l'approche construction de blocs qui consiste à développer des nouvelles applications informatiques à partir de produits existants. Le deuxième approche est le processeur génératif qui sert à réutiliser les processus qui sont encodés dans un processeur génératif, qui réutilise les processus encodés dans un processeur (compilateurs, générateurs de code, etc.). La partie technique concernant la réutilisation, ne fera pas l'objet de la présente étude. On s'attardera surtout sur les bénéfices en terme de productivité que la réutilisation logicielle peut offrir aux divers intervenants dans le développement et la maintenance du logiciel.

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). Un 60% de la conception et du code des applications d'affaire est réutilisable (Lanergan, R.G et Grasso, C.A. 1984). 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 à l'investissement très avantageux.

Dans les sections suivantes 2.3.2 et 2.3.3, 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. Le premier est le développement de produits réutilisables, tandis que le deuxième est le développement avec des composantes réutilisables. Dans les sections 2.3.2 et 2.3.3 , on fera un survol de ces deux approches.

2.3.2 Création des composants logiciels afin de les réutiliser

Concernant l'impact de la réutilisation dans la stratégie de réutilisation dans le processus de génie logiciel, on peut dégager certains critères qui favorisent cette approche:

- Amélioration de l'efficacité des futurs projets
- Élaboration des standards
- Création de composants indépendants des environnements
- Stabiliser les composants à réutiliser
- Faciliter la formation

2.3.3 Réutiliser les composantes déjà existantes

La réutilisation de composants déjà existants dans une application informatique, présente des avantages plus faciles à cerner

- 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

2.4 La mesure utilisée dans cette approche

La métrique utilisée dans le modèle 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). 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 tel que présenté par l'analyse de points de fonction version 4.0. La deuxième approche, utilise les résultats de l'application de cette méthode 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.

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 métrique a été raffinée par Albrecht et Gaffney en 1993 et c'est « l'International Function Point Users Group(IFPUG) » (IFPUG,1994), qui à partir de 1986 établit les normes à suivre.

La taille fonctionnelle dans l'analyse de points de fonction, est obtenue en mesurant le logiciel à partir de deux procédures complémentaires. Premièrement, on assigne un poids pour chaque fonction. Les fonctions sont classifiées en deux catégories ; les données et les transactions. Deux types de fonction composent la partie donnée : le groupe de données internes(GDI) et le groupe de données externes(GDE). Les transactions sont divisées en trois catégories : les entrées, les sorties et les interrogations. 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 appelé 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.

2.4.1 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. La méthode proposée par Abran et al. utilise dans son élaboration un ensemble 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.

2.4.1.1 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 ou les rapports, etc.) Les mesures des points de fonction se basent alors sur les biens livrables aux usagers, afin de les supporter dans leurs affaires. On s'intéresse au « quoi », ou les livrables finaux et non au « comment », ou les moyens utilisés par l'informatique pour arriver aux résultats.

2.4.1.2 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.

2.4.2 Élaboration de base 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. L'élaboration de bases de données historiques de points de fonction utilisées dans cette étude a été faite à partir de deux approches fondamentales: La première approche est d'aller extraire dans la documentation de chaque projet informatique mesuré, les données nécessaires à sa construction(les groupes de données internes, externes, les entrées, les sorties et les interrogations). Par la suite, on trouve 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 sur les changements qui seront apportées ou qui ont été faits. Pour le calcul du pourcentage la procédure est la même. Toutes les données collectées sont emmagasinées soit dans un chiffrier électronique, soit dans un logiciel de base de donnée ou sur un autre support informatique. Ces données sont stockées par projet. Par exemple une application quelconque de 100 points de fonction. Ce total de points de fonction, sont divisés par le nombre de points correspondants à chaque type de fonction.

Pour mieux illustrer aux lecteurs, on pourrait présenter l'exemple suivant de collecte de données.

Numéro du projet : 25

Description du projet : Serveur bancaire

Service ou unité : Service aux entreprises

Chef du projet : No. 1997-1

Nombre total de points de fonction : 200

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

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 suivants:

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%

Le total en pourcentage doit être égal à 100% du nombre total de points de fonction des 50 projets mesurés.

2.5 Potentiel de la méthode proposée par Abran et Desharnais (1995)

Dans le génie logiciel, il existe trois entités à mesurer : le produit logiciel, les ressources impliquées et les différents processus qui interviennent dans le cycle de vie du logiciel. La première entité se réfère aux résultats explicites des différentes activités intervenantes dans le génie logiciel. Ces résultats sont composés par les différents livrables, ainsi que par la documentation des différents processus du développement du logiciel. La deuxième entité concerne les intrants de toutes les activités qui rentrent en compte dans la construction d'un produit logiciel. La troisième entité implique les phases du cycle de vie du logiciel, ainsi que d'autres activités propres au logiciel : la spécification des besoins, la conception, le code, les tests, la maintenance, la réingénierie, la réutilisation et l'assurance qualité. L'objet de la présente étude se situe dans la première entité. Plus spécifiquement à l'aspect concernant la mesure de la réutilisation fonctionnelle dans les applications informatiques.

Le concept de métrique du logiciel, est un des sujets le plus mal compris dans le domaine du génie logiciel. Cependant le rôle critique que joue la mesure dans le génie logiciel est bien reconnu et apprécié ces derniers jours. Tandis que le génie logiciel comprend une collection d'activités concernant le développement et la gestion du logiciel, la métrique du logiciel, concerne les aspects quantifiables dans ces activités. Plus précisément, ces aspects permettent de mesurer et prédire les coûts des projets, la qualité et la complexité du produit logiciel et de mesurer et ainsi pouvoir améliorer la productivité.

La mesure est un moyen quantitatif pour une entreprise de connaître si elles sont en train de rencontrer leurs objectifs techniques et d'affaires. La métrique du logiciel est le moyen par lequel les gestionnaires du génie logiciel peuvent mesurer et prédire certains aspects du processus, des ressources et les produits qui sont pertinents aux activités du génie logiciel.

Le modèle de mesure de la réutilisation fonctionnelle proposée par Abran et Desharnais(1995), peut être utilisée pour l'élaboration de l'estimation d'un projet, afin de contrôler le progrès dans le projet, dans la détermination d'une relative complexité. Elle aide aussi à vérifier certaines pratiques afin d'améliorer la prise de décision. Ce modèle se veut un outil très important pour les divers intervenants du domaine logiciel.

2.5.1 Élaboration de modèles de productivité ou d'estimation

Le fait de mesurer la réutilisation de la taille fonctionnelle dans la maintenance du logiciel, n'a aucune importance si ces résultats ne sont pas appliqués à un modèle qui montrera les bénéfices engendrés par cette activité.

Dans la recherche réalisée par Abran et Desharnais (1985) , le résultat du modèle de mesure utilisé sert d'intrants dans un modèle de productivité afin de vérifier dans un scénario particulier l'extrait résultant. Il est très important de remarquer que ce modèle proposé doit être vérifié avec plusieurs bases de données historiques de points de fonction et avec plusieurs scénarios réels d'ajout des fonctionnalités dans les projets informatiques. Ce travail de recherche développera plusieurs jeux d'essais qui seront représentatifs de plusieurs applications informatiques.

2.5.2 Bénéfices potentiels d'utilisation de la méthode proposée

Cette section décrit les bénéfices tangibles pour une organisation qui appliquera cette méthode valide de mesurer la réutilisation fonctionnelle. Les bénéfices seront classifiés selon l'implication du gestionnaire dans le développement et la maintenance des applications informatiques.

Les gestionnaires impliqués dans la maintenance logiciel, seront supportés afin de mieux obtenir les bénéfices de l'utilisation d'une méthode de mesure de réutilisation fonctionnelle. Parmi les bénéfices les plus significatifs on peut nommer:

- Efficacité de l'utilisation des ressources assignées aux différentes activités de maintenance logiciel
- Une meilleure prise de décision
- Réduction du "backlog" dans la maintenance
- Meilleure réponse aux besoins d'affaires
- Rapidité dans l'implantation des changements
- Amélioration de la fiabilité et la stabilité
- Meilleur suivi de l'assignation des tâches

- Meilleure consistance des standards

Pour la haute direction, les bénéfices seront les suivants:

- Estimation plus effective des activités de développement et de maintenance
- Réduction du temps de maintenance et de redéveloppement
- Un alignement des systèmes d'information à la stratégie d'affaires
- Amélioration de la performance des systèmes d'information
- Amélioration du processus d'assurance qualité

Les bénéfices pour les corporations:

- Réponse rapide aux changements d'affaires
- Amélioration du service aux clients
- Réduction des pannes du service d'information

Chapitre III

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)

3.1 Description du cadre d'expérimentation de Basili

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 3.1. Il permettra de réaliser et de documenter ce projet de recherche exploratoire.

Tableau 3.1

Les quatre phases qui composent le 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 de la métrique et définitions		
Phase 3. Réalisation					
Préparation		Déroulement		Analyse des données	
Phase 4. Interprétation					
Contexte d'interprétation			Extrapolation des résultats		

Ce cadre de recherche de Basili est composé de quatre parties principales : la définition, la planification, l'exécution et l'interprétation de 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.

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 de la métrique, 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.

Dans le cadre de cette proposition de l'activité de synthèse, on présente certaines activités contenues dans les deux premières phases de ce cadre méthodologique (Définition et Planification). Le gabarit qui est présenté dans le tableau 3.2 énumère les principales activités qui seront développées dans l'étape première du cadre de Basili (Définition).

Le gabarit présenté dans le tableau 3.3 montre aussi les différentes activités parmi les plus représentatives qui seront réalisées dans la planification de cette étude et qui correspond à la deuxième étape du cadre de Basili.

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 le rapport final de cette étude.

Tableau 3.2

Activités à développer dans la phase 1 du cadre de Basili

Phase 1..Définition					
Motivation	Objet	But	Perspective	Domaine	Portée
.Évaluation de la mesure de réutilisation fonctionnelle	.Modèle de réutilisation fonctionnelle (Modèle de productivité)	.Expérimenter et vérifier la mesure de réutilisation fonctionnelle et le modèle de productivité avec un ensemble de données recueillies dans l'industrie	.Utilisateurs .Gestionnaires de projets	.Type de données qui provient de l'industrie ?	.Plusieurs contextes (scénarios) de travail .Plusieurs ensembles de données

Dans le tableau ci-haut on identifie les 5 sous-phases qui définissent la motivation de la présente étude. L'objet, le but, la perspective, ainsi que le domaine et la portée sur lesquels la présente étude sera élaborée.

Le tableau ci-dessous permet de dégager les activités qui seront nécessaires pour encadrer la démarche de recherche de la présente étude. On identifiera les activités concernant la conception expérimentale. Les critères qui détermineront la vérification de la recherche seront aussi présentés et finalement on définira la métrique utilisée ainsi que la manière comme les données industrielles qui seront utilisées dans cette étude ont été collectées.

Tableau 3.3

Activités à développer dans la phase 2 du cadre de Basili

Phase 2. Planification		
Conception expérimentale	Critères	Sélection de la métrique et définitions
<ul style="list-style-type: none"> . Choix des industries . Recueillir les données industrielles . Analyse de la méthode de mesure (Points de fonction) . Plusieurs scénarios de simulation faits sur mesure . Écarts de fiabilité des résultats . Identifier le comportement du modèle 	<ol style="list-style-type: none"> 1. Critères directs <ul style="list-style-type: none"> . Cohérence du modèle . Existence d'un point d'inflexion 2. Critères indirects <ul style="list-style-type: none"> . Procédures de vérification des résultats du modèle 	<ol style="list-style-type: none"> 1. Métrique <ul style="list-style-type: none"> . Points de fonction . Élaboration de bases de données historiques de points de fonction . Analyse de diverses bases de points de fonctions utilisées 2. Collecte de données <ul style="list-style-type: none"> . Plusieurs ensembles de bases de données de points de fonction . Collecte faite manuellement

Chapitre IV

Définition du projet de recherche

Dans ce chapitre, nous présenterons 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.

4.1 Motivation, objet, propos et le domaine de la recherche

Dans la plupart d'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 composantes 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 composantes informatiques, Abran et Desharnais(1995) ont élaboré un modèle de mesure de la réutilisation fonctionnelle. Toutefois, ce modèle n'a été développé que avec un seul contexte expérimental.

L'objet de cette étude est d'évaluer le modèle de réutilisation fonctionnelle avec une série de cas concrets qui ont été élaborés à partir de certaines données industrielles. Dans cette partie, nous évaluerons une mesure de réutilisation fonctionnelle, ainsi que le modèle de productivité contenue dans ce modèle. Ce dernier sert à montrer les gains potentiels pour le budget informatique, si on utilise le modèle proposé.

Le propos du travail concerne l'élaboration d'un cadre de travail qui permettra d'évaluer l'objet de cette étude, soit le modèle de mesurer la réutilisation fonctionnelle, tel que décrit.

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 créés.

4.2 Objectifs et limites de la recherche

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

- Expérimenter la mesure de la réutilisation fonctionnelle, ainsi que le modèle de productivité proposé, afin de vérifier leur consistance.
- L'identification des scénarios de travail dans lesquels le modèle s'applique et ne s'applique pas.

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

- Cette étude se base seulement sur les concepts élaborés dans l'article "Measurement of Functional Reuse in Maintenance"(Abran et Desharnais, 1995)
- Ce projet est élaboré à partir de quatre bases de points de fonction qui proviennent de deux institutions financières canadiennes, d'un organisme gouvernemental du Québec et une autre extraite de la littérature.
- Les différents scénarios de travail, sont élaborés à partir de situations réelles qui se sont présentés dans plusieurs entreprises. Néanmoins, l'auteur de la présente étude ne décrit les situations d'une manière très détaillée afin de garder la confidentialité de ces entreprises.
- Les scénarios de travail de la présente étude, sont limités à des situations où des applications MIS ont été développées.

4.3 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, 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 développement et maintenance du logiciel, et en modèles de productivité, qui profiteront des éléments contenus dans cette étude et qui trouveront de nouvelles pistes pour des recherches futures.

Chapitre V

Planification du projet de recherche

Ce chapitre correspond à la phase deux du cadre méthodologique de recherche de Basilic. Cette partie présente les différentes étapes qui composent le projet, leurs intrants et livrables associés. Une brève description du contenu prévu de chaque livrable sera aussi énoncée. L'échéancier prévu pour ce projet est présenté dans la dernière partie de ce chapitre.

5.1 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 composent les 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 de livrables intermédiaires ou finaux par rapport aux résultats du projet. Les livrables de chaque étape sont assemblés entre eux afin de constituer un livrable finaux qui sera le résultat de ce projet de recherche. L'ensemble des livrables finals 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 5.1.

Tableau 5.1

Les étapes les intrants et les livrables du projet de recherche

Étapes	Intrants	Livrables
Étape 1 . Positionnement de la méthode proposée dans le développement et la maintenance logiciel	. Revue de la littérature . Cas concrets extraits des sites industriels, concernant la mesure de productivité dans les projets informatiques à l'aide des points de fonction . Utilisation des bases historiques de points de fonction pour la vérification de la méthode proposée	. Justification de l'emploi de la méthode (Livrable final du projet)
Étape 2 . Élaboration de bases historiques de points de fonction	. Données concernant la mesure des projets provenant de plusieurs projets mesurés avec 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)

5.2 Contenu des livrables

5.2.1 Justification de l'approche d'estimation

La planification d'un programme pour mesurer le logiciel et l'établissement des priorités basées sur des objectifs est une tâche qui concerne plusieurs niveaux d'organisation dans une firme. Les objectifs pour les programmes de métriques doivent respecter les caractéristiques d'ordre stratégiques, mesurables, réalisables, réalistes et bien ciblés. (Dekkers, 1997).

Avec ces objectifs, on détermine l'audience cible et les questions à répondre, afin d'atteindre les objectifs visés. L'analyse de l'audience et les groupes ciblés sont deux véhicules pour obtenir l'information nécessaire pour les gestionnaires dans la prise des décisions pertinentes. Dans cette activité de synthèse, on propose une approche de prototypage qui sera très utile aux gestionnaires. Avec le raffinement du modèle, d'autres bases de données historiques et des scénarios différents de ceux utilisés dans cette recherche, permettront l'amélioration de la méthode.

5.2.2 Opportunités de l'utilisation de la méthode proposée

Une fois l'étape de la planification et la construction d'un programme de mesure passée, la prochaine étape est l'identification d'opportunité d'utilisation de la méthode proposée.

Le tableau 5.2 permet de dégager certaines activités de gestion dans lesquelles la méthode proposée sera fort utile dans la planification des différentes activités de développement informatique et compris la mesure de la réutilisation fonctionnelle dans la maintenance du logiciel.

Tableau 5.2
Opportunité d'utilisation de la méthode proposée

Objectives	Méthode proposée
Amélioration des estimés	oui
Amélioration de la productivité	oui
Amélioration de la qualité	non
Évaluation, construction ou achat du logiciel	oui
Amélioration des activités de tests du logiciel	non

5.2.3 Élaboration de bases historiques de points de fonction

5.2.3.1 Préparation

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. L'élaboration des bases de données historiques de points de fonction qui seront utilisées dans cette recherche, a été réalisée dans des institutions de calibre industriel. La première collecte a été effectuée et testée dans une institution financière canadienne à partir de 1990 jusqu'à 1994 (Abran, 1994), (Abran et Nguyenkim, 1993) et (Abran et Nguyenkim, 1991). La deuxième collecte a été réalisée dans une institution gouvernementale du Québec, pendant l'automne 1995. La troisième est réalisée depuis 1993 et elle se fait dans une institution financière canadienne. La deuxième et la troisième collectes de données ont été réalisées en partie par l'auteur de cette recherche. Une dernière base de données de points de fonction est tirée de la littérature.

En plus de ces deux bases de données de moyenne de points de fonction, cette étude fera l'utilisation de deux autres bases de données de points de fonction collectées dans une institution financière canadienne (Abran, 1994, Abran et Nguyenkim, 1993 et Nguyen-kim, 1991). On

utilisera aussi une autre base de moyenne de points de fonction puisée dans la littérature (Dreger, 1989)

5.2.4 Élaboration de scénarios de test de la méthode proposée

L'hypothèse fondamentale de cette conception expérimentale est de vérifier le comportement du modèle proposé et ses effets dans des situations autres que celles du contexte initial. Le résultat de cette hypothèse est d'assurer que la méthode de mesure de la réutilisation fonctionnelle dans la maintenance, soit décrite et construite correctement et que leur résultat soit "cohérent et consistant". Dans le cas contraire identifier les situations qui font varier le comportement du modèle initial proposé.

Le critère fondamental de cette conception expérimentale est d'illustrer le potentiel d'utilisation de ce modèle avec un ensemble de données industrielles. Pour ce faire, on utilisera des données fiables et représentatives de la plupart d'applications informatiques dans le domaine MIS. Deux critères ont été pris en compte dans la collecte de données. Le premier est le calibre industriel des entreprises, car on voulait des données représentatives et diversifiées. Le deuxième critère est l'utilisation des techniques d'analyses structurées connues afin de recueillir les données nécessaires à la technique de points de fonction. Cela a permis de dégager une cohérence précise concernant les données collectées et les bases de données historiques de points de fonction qui ont été élaborées à partir de ces données.

En ce qui concerne les différents scénarios de travail, ils seront élaborés à partir de situations réelles qui se sont présentés dans plusieurs entreprises. Cependant afin de garder la confidentialité des ces institutions, on croit sage de ne pas les citer. Néanmoins ces données font partie de la vie quotidienne de ces institutions et elles sont analogues à la plupart d'institutions qui œuvrent dans le même créneau.

5.3 L'échéancier prévu

L'effort du présent projet de recherche est estimé en 700 heures/personnes. De ce total 270 ont été reliées à l'élaboration et à la présentation de cette proposition d'activité de synthèse. Le nombre de 420 heures/personnes, concernent l'exécution du projet, à l'interprétation des résultats jusqu'à la finalisation du rapport final. Cet effort comprend le travail réalisé à partir de la dernière semaine du mois de juillet jusqu'à la première semaine du mois de décembre (1997). Cet échéancier permettra l'accomplissement des différentes étapes prévues dans le projet.

L'effort estimé des tâches principales, ainsi que leurs dates de réalisation, sont présentées dans le tableau 5.3

Tableau 5.3
Effort et dates estimées de réalisation du projet

Tâches principales	Effort (heures/personne)	Début	Fin
Proposition de l'activité de synthèse	270	2 juin 97	29 août 97
Ajustement du projet	40	29 août 97	5 septembre 97
Réalisation des étapes du projet	150	8 septembre 97	15 septembre 97
Interprétation des résultats	60	16 septembre 97	31 octobre 97
Finalisation du projet	125	3 novembre 97	28 novembre 97
Gestion du projet et imprévus	50	1 décembre 97	4 décembre 97
Présentation de l'activité de synthèse	5	5 décembre 97	5 décembre 97
Total du projet	700	2 juin 97	5 décembre 97

Cette proposition d'activité de synthèse comprend les phases de définition et planification du projet, aussi bien que la présentation de cette proposition.

La phase d'exécution du projet comprend trois étapes. La première celle de la préparation. En ce qui concerne cette activité de synthèse, elle consistera à la sélection des industries qui seront mis à contribution et à l'analyse du modèle proposé. La deuxième étape du déroulement concerne les étapes de collecte de données, ainsi que leur validation. La troisième étape est celle de l'analyse de données. Dans cette étape on fera une analyse préliminaire du résultat des différents scénarios de travail testés.

La phase de l'interprétation est composée de deux parties : le contexte d'interprétation et l'extrapolation des résultats. Dans la première partie, on construira plusieurs scénarios de travail qui détermineront le comportement du modèle étudié. Dans la deuxième partie, on identifiera le potentiel de cette étude concernant des travaux futurs dans le domaine concerné.

Appendice A

Exemple d'un cas industriel qui démontre les bénéfices de la méthode de mesure de la réutilisation fonctionnelle

Un utilisateur demande l'addition de 200 points de fonction dans une application existante composée de 3000 points de fonction.

En moyenne, dans le développement d'une application à partir du « scratch » chaque point de fonction coûte deux jours-personne.

En moyenne, dans une activité de maintenance (ajout de fonctionnalités) chaque point de fonction coûte trois jours-personne.

Afin de mieux présenter les diverses opérations à effectuer, on présente dans le tableau A.1, une distribution moyenne de points de fonction à partir d'un portefeuille d'applications informatiques d'une institution financière canadienne. (Abran, 1995) et un autre tiré de la littérature (Dreger, 1989). Les données contenues dans ce tableau serviront à l'application de la méthode proposée par Abran et Desharnais..

Tableau A.1 Distribution moyenne de points de fonction

Types de fonction	Nouvelles applications (Abran 94)	Ajout de fonctionnalités majeurs (Abran 94)	Nouvelles applications (Dreger 89)
Groupe de données internes (GDI)	27%	13%	36%
Groupe de données externes (GDE)	14%	38%	1%
Entrées	22%	9%	31%
Sorties	21%	31%	27%
Interrogations	17%	9%	9%

Le coût de développement pour cette application de 200 points de fonction, devra être de 600

jours-personne. Si on utilise la méthode proposée par Abran et Desharnais, l'indicateur de réutilisation est obtenu en appliquant le ratio de GDE du tableau A.1, correspondant à la section ajout de fonctionnalités majeurs (38%). On utilise la formule 2 afin de trouver le nombre de points de fonctions qui correspondent aux GDE réutilisables.

$$\begin{aligned} \text{PF}(\text{indicateur}) &= \text{PF} \times \text{PF (GDE) (ratio)} \\ &= 200 \text{ PF} \times 38\% \\ &= 76 \text{ PF GDE} \end{aligned}$$

L'indicateur de réutilisation représente 76 points de fonction de GDE qui existent en dehors de la frontière de l'application mesurée. Ces fonctions ne doivent pas être développées, car elles existent déjà et doivent être réutilisées sans modification.

Afin de connaître les différentes transactions qui sont nécessaires afin de créer les fonctions réutilisées, le concept de ratio prédicteur doit être utilisé. On utilise la formule 3. En moyenne chaque point d'un fichier interne est prédicteur de 2.7 points (1 point pour un GDI(27%) et 73% pour les autres types de fonction. Ce résultat est le ratio prédicteur nécessaire afin de maintenir les fichiers créés. Le ratio prédicteur de 2.7 appliqué au 76 PF de l'indicateur de réutilisation donne 205 transactions qui ne doivent pas être développées.

Étape 5. La taille de la mesure alternative peut être calculée à partir de la formule 8 en ajoutant le nombre de points de fonction des transactions évitées à la taille fonctionnelle du projet.

$$\begin{aligned} \text{PF} &= \text{PF}(\text{demandés}) + \text{transactions évitées} \\ \text{PF} &= 200 \text{ PF} + 205 \text{ PF réutilisés} \\ \text{PF} &= 405 \text{ PF réutilisés} \end{aligned}$$

Du point de vue de la réutilisation la taille alternative du projet est de 405 PF réutilisés) au lieu de 200 PF quand les transactions évitées et non redéveloppées sont prises en compte.

Dans la dernière partie de l'article une analyse de productivité est réalisée afin de vérifier le modèle proposé. Le projet original (ajout de 200 points de fonctions) sera de 600 jours-personne

Cependant, quand les transactions évitées sont prises en compte, le coût pour chaque point de fonction décroît à 1.5 jours-personne (600 jours/405jours) au lieu de 3 jours-personne. Ceci est un indice de l'avantage d'utiliser le modèle de mesure de la réutilisation proposé. Dans le tableau A.2 on présente les résultats du cas pris comme exemple par les auteurs de la méthode proposée.

Tableau A.2

Étude de cas : Mesure de la réutilisation fonctionnelle avec et sans réutilisation

Concept réutilisation	Type de projet	Coût unitaire	Taille en PF	Coût total
SANS RÉUTILISATION (mesure avec des PF traditionnels)	Nouvelle application	2 jours-personne par PF	200	400
	Ajout de fonctionnalités	3 jours personne par PF		600
AVEC RÉUTILISATION (inclut les transactions évitées)	Ajout de fonctionnalités	1.5 jours- personne par PF	405	600
	Alternative(si construit comme nouvelle application)	2 jours-personne par PF		810

Ce tableau présente deux sections. La première est la partie dans laquelle les projets ont été développés sans réutilisation et la deuxième partie représente les projets qui ont été développés avec réutilisation. La nouvelle mesure alternative a besoin du redéveloppement de 205 points de fonction (qui doivent être des transactions évitées) possède une taille totale de 405 points de fonction. Basé dans le coût de 2 jours-personne par PF pour les nouvelles applications, le projet à développer aura le coût de 810 jours-personne ($405 \text{ PF} \times 2 \text{ jours-personne} = 810 \text{ jours-personne}$). Cet exemple nous donne un avantage économique de 210 jours-personne ($810 - 600 \text{ jours-personne}$) pour un projet où des fonctionnalités majeures ont été ajoutées.

Bibliographie

Abran, Alain. 1994. Analyse du processus de mesure des points de fonction. PhD thesis, École Polytechnique , Université de Montréal

Albrecht, A.J. 1979. « Measuring Application Development Productivity ». In proceeding of the joint SHARE GUIDE and IBM Application Development Symposium, pp 83-92, Monterey, CA., Oct.

Abran, Alain et Desharnais, Jean-Marc. 1995. « Measurement of Functional Reuse in Maintenance, Software Maintenance:Research and Practice ». vol.7, pp263-277

Albrecht, A.J. et Gaffney, J.E. 1983. « Software Function, Source Lines of Code and Development Effort Prediction: A Software Science Validation ». IEEE Transactions on Software Engineering, SE-9(6):639-648, Nov.

Abran, A. et Nguyenkim,H. 1993 « Measurement of the Maintenance Process from a Demand-based Perspective ». Software Maintenance:Research and Practice, 5:63-69

Abran, A. et Nguyenkim, H. 1991. « Analysis of Maintenance Work Categories Through Measurements ». In proceeding of the IEEE Conference on Software Maintenance, pp 104-113, Octobre

Abran A. et Taboubi, Z. 1994. « Validation de données pour le modèle de points de fonction spécifique à la maintenance adaptative ». Technical report, Université du Québec à Montréal, Novembre

Basili, Victor R. 1986. « Experimentation in Software Engineering ». IEEE Transactions on Software Engineering, vol. SE-12, no.7, July

Basili, V.R. et Musa, J. D. 1991. « The future Engineering of Software : A Management Perspective ». IEEE Computer, Sept, vol. 24(9), pp. 90-96

Biggerstaff T.J. et Perlis, A.J. 1984. « Reusability Framework, Assessment, and Directions ». IEEE Software, July, pp. 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. pp 44 86

Chidamber, S.R. et Kemerer, C.F. 1991. « Towards a metrics suite for object oriented design ». In proc. 6th ACM Conf. Object Oriented Programming Syst., Lang. And applicat. (OOPSLA), Phoenix, AZ, 1991, pp.197-211

Dreger, J. Brian. 1989. « Function Points Analysis ». Prentice Hall, Englewood Cliffs, N.J.,

Dunham, J.R., Kruesi, E. 1983. « The Measurement Task Area ». IEEE Computer, November

Freeman, P. 1983. « Reusable software engineering : Concepts and research directions ». In Workshop on Reusability in Programming, Sept.

Gaffney, John E. et Cruickshank R.D. 1992. « A general economics model of software reuse ». Proceedings of 14th International Conference on Software Engineering, ACM Press, New York, pp. 327-337

Grady, Robert B. 1990. « Work-Product Analysis ». The Philosopher's Stone of Software IEEE Software, March

Grady, Robert B. 1987. « Measuring and Managing Software Maintenance ». IEEE Software, vol.4, no 9, September, pp 35-45

Grady, Robert B. et Caswell, Deborah, L. 1987. « Software Metrics: Establishing a Company ». Wide Program, Prentice Hall

Harrison, Richard. 1987. « Maintenance giant sleeps undisturbed in Federal data centers ». Computerworld, 21(10), 81,86

IFPUG International Function Points Users Group. 1994. IFPUG Function Point Counting Practice Manual Release 4.0, IFPUG, Westerville, OH 293 pages

Jacquet, J. P. et Abran, A. 1996. « ISO Verification of a functional size measurement method (Part A) » DRAFT

Jacquet, J.P. et Abran, A. 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 Abran, A. 1996. « Conception d'un guide de validation des méthodes de mesure de taille fonctionnelle » Neuvièmes journées internationales, Paris

Kearney, J. Sedlmeyer, L, Thompson, W.B., Gray, M.A. et Adler, M.A. 1986. « Software Complexity Measurements ». Communication of the ACM, November, vol 29, no 11

Kim, Yongbeom et Stohr, Edward A. 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, pp. 612-623

Kitchenham, Barbara. 1992. « Procedures and Mechanisms for Measurement Experimentation »

Krueger, C.W. 1992. « Software Reuse ». ACM Computing Surveys, vol. 24, no 2, June, pp. 131-183

Lanergan, R.G. et Grasso, C.A. 1984. « Software Engineering with Reusable Design and Code ». IEEE Transactions on Software Engineering, vol. Se-10, no 5 Sept, pp. 498-501

Lim, W.C. 1994. « Effects of Reuse on Quality, Productivity and Economics ». IEEE Software, vol. 11, no 5, Sept., pp. 23-30

Lientz, Bennet P. et Swanson, Burton E. 1980. « Software Maintenance Management ». Addison-Wesley, Reading, Ma. 214 pages

Oskarsson, Östen et Glass, Robert L. 1996. « An ISO 9000 Approach to Building Quality ». Software, Prentice Hall PTR, New Jersey, USA

Page-Jones, Meilir. 1988. « The Practical Guide to Structured Systems ». Second Edition, Yourdon Press

Poulin, Jeffrey S. 1994. « Measuring Software reusability ~~td~~ ». Software Conference, Rio 94, IEEE

Poulin, J.S., Caruso, J.M. et Hancock, D.R. 1993. « The business case for software reuse, ». IBM Systems Journal, vol. 32, no 4 pp. 567-594

Prater, R.E. 1984. « An axiomatic theory of software complexity measures ». Computer Journal, vol. 27, pp. 340-346

Pressman, R.S. et Herron, S.R. 1991. « Software Shock ». Dorset House, New York

Roberts, F.S. et Franke C.H. 1976. « On the Theory of Uniqueness in Measurement ». Journal of Mathematical Psychologie, 14 pp 211-218

Rubin, h.A. 1987. « Critical Success Factors for Measurement Programs ». Proc. Spring Conference of IFPUG, International Function Point Users Group, Scottsdale, Arizona

Schach, Stephen R. 1993. « Software Engineering ». Second Edition, IRWIN, Burr Ridge, Illinois

Schneidewind, Norman 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 pp 95-98

Shepperd, M. et Ince, D. 1993. « Derivation and validation of software metrics ». Oxford Science Publications

Suppes, P. et Zinnes, J.L. 1963. « Basic Measurement Theory, Handbook of Mathematical Psychology ». Edited by D.D. Luce, R.R. Bush, E. Galanter, Volume I, Wiley, New York 1963, pp 1-76

Tracz, Will. 1987. « Software Reuse: Motivators and Inhibitors ». In Proceedings of COMPCONS'87, pp. 358-363, IEEE Press

Weyuker, E. 1988. « Evaluating software complexity measures ». IEEE Transaction Software Eng., vol. 14, pp. 1357-1365

Zuse, Horst. 1996. « A Framework of Software Measurement ». Berlin (It's not yet public)

Zuse, Horst. 1994. « Foundations of validation, prediction and software measure ». Proceeding of the AOSW94 (Annual Oregon Software Metric Workshop), Portland, April 20-22.