

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

MESURER LA TAILLE FONCTIONNELLE D'UN ÉCHANTILLON DES
APPLICATIONS CENTREX ET ÉTABLIR UN MODÈLE DE PRODUCTIVITÉ POUR
CES ACTIVITÉS DE MAINTENANCE CHEZ ABC Inc.

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

Par

LAURA PRIMERA

AVRIL 2001

REMERCIEMENTS

Je remercie sincèrement tous ceux qui ont contribué à la réalisation de ma thèse dans le cadre de mes études de maîtrise en génie logiciel.

Plus particulièrement, j'ai apprécié beaucoup l'opportunité, la confiance et l'apport octroyé par mon directeur de recherche M. Alain Abran, professeur à l'Université du Québec à Montréal et directeur du Laboratoire de recherche en gestion des logiciels à l'UQAM, pour la réalisation de ce projet.

Mes remerciements également au professeur Robert Dupuis, qui m'a appuyé et orienté significativement tout au long des mes études de maîtrise en génie logiciel.

Je remercie aussi à M. Bertrand Fournier, statisticien consultant professionnel au SCAD (Service de consultation en analyse de données) de l'UQAM pour ses précieux conseils qui me permettaient de valider les données recueillies et faire leur interprétation d'une manière appropriée.

Je remercie tous mes collègues chez ABC Inc. qui ont accepté de participer à la réalisation de ce projet.

Enfin, je remercie Dieu qui éclaire toujours mon horizon et me donne le courage, et la passion pour atteindre mes objectifs. En plus, j'adresse un remerciement plus profond à mon mari, mon petit enfant et à ma famille pour leur confiance, leur appui inconditionnel et leur encouragement à persévérer.

TABLE DES MATIÈRES

REMERCIEMENTS	II
LISTE DES FIGURES	V
LISTE DE TABLEAUX	VI
LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES	VII
RÉSUMÉ	VIII
CHAPITRE I : CONTEXTE DU PROJET	5
1.1 Description sommaire de l'organisation	5
1.2 Situation actuelle de l'entreprise	6
1.3 Besoins à combler pour la maintenance chez ABC Inc.	7
1.4 Besoins du laboratoire de recherche en gestion des logiciels (LRGL)	8
1.5 Situation de l'intervention dans l'entreprise	9
1.6 Définition et Objectifs du mandat	9
1.7 Limites du mandat	10
1.8 Approche de réalisation	11
1.9 Objectifs de l'entreprise et mesures prévues pour leur atteinte	12
1.10 L'apport du projet au génie logiciel	14
CHAPITRE II : ANALYSE DES DONNÉES	15
2.1 Hypothèses pour extrapoler les résultats d'un échantillon de données à la population	16
2.2 Validation de la distribution normale des variables (distribution gaussienne approximative)	16
2.3 Histogrammes et distribution cumulative de fréquences	18
2.4 Critères de fiabilité pour un modèle de productivité	21
2.4.1 Paramètres indiquant la qualité d'ajustement du modèle aux données	22
2.4.2 Paramètres indiquant l'exactitude du modèle	23
CHAPITRE III : DÉTERMINATION D'UN MODÈLE DE PRODUCTIVITÉ	25
3.1 Modèle du coût unitaire moyen	25
3.1.1 Qualité de ce modèle de productivité	27
3.1.2 Modèles avec sous-échantillon	27
3.2 Modèle de régression linéaire	34
3.2.1 Modèle à une seule variable indépendante	34
3.2.2 Identification de facteurs supplémentaires	39
3.2.3 Modèles à plusieurs variables indépendantes	40
3.3 Ajustement des critères utilisés pour classer les données	46
3.3.1 Modèle pour déterminer l'influence de l'ajustement de la complexité fonctionnelle	48
CHAPITRE IV : COMPARAISON DES RÉSULTATS AVEC D'AUTRES ÉTUDES EMPIRIQUES	54
CHAPITRE V : SOMMAIRE DES RÉSULTATS	58

RÉFÉRENCES	62
APPENDICE A – SOMMAIRE DES DONNÉES	64
A1. Données mesurées et complémentaires	64
A2. Validation et transformation des données.....	65
A3. Ajustement des critères utilisés pour classifier les données à l'égard de la complexité fonctionnelle et l'expérience des programmeurs.	66
A4. Modèle du coût unitaire moyen pour l'ensemble de projets.....	67

LISTE DES FIGURES

Figure 1.1 – Phases dans le projet	11
Figure 2.2 – Histogramme de la variable LogFFP.....	19
Figure 2.3 – Distribution cumulative de fréquence de la variable LogFFP	20
Figure 2.4 – Histogramme de la variable LogEffort	20
Figure 2.5 – Distribution cumulative de fréquence de la variable LogEffort.....	21
Figure 3.6 – Erreur relative de l'ensemble de projets.....	26
Figure 3.7 – Erreur relative de chaque projet – Personnel expérimenté	28
Figure 3.8 – Erreur relative de chaque projet – Personnel intermédiaire	30
Figure 3.9 – Erreur relative de chaque projet – Personnel novice.....	31
Figure 3.10 – Erreur relative de chaque sous-échantillon de projets et de l'ensemble de projets	33
Figure 3.11 – Diagramme de dispersion avec la ligne de régression correspondante du modèle de la régression linéaire simple.....	36
Figure 3.12 – Modèle linéaire avec des transformations logarithmiques: $Y = A + B \cdot X$	38
Figure 3.13 – Ajustement des critères utilisés pour classifier les données : Complexité Fonctionnelle et Niveau d'expérience en informatique	47
Figure 3.14 – Représentation graphique de l'équation 3.8 et 3.10	51

LISTE DE TABLEAUX

Tableau 1.1 – Diverses origines des difficultés de la maintenance	6
Tableau 1.2 – Mesures prévues pour atteindre les objectifs de l'entreprise.....	13
Tableau 1.3 – Attributs et métriques pour construire des modèles d'estimation de productivité	13
Tableau 2.4 – Test de distribution normale pour les variables LogFFP, LogEffort	17
Tableau 3.5 – Modèle du coût unitaire moyen par l'ensemble de projets.....	27
Tableau 3.6 – Modèle du coût unitaire moyen par personnel expérimenté	29
Tableau 7 – Modèle du coût unitaire moyen par personnel intermédiaire.....	30
Tableau 3.8 – Modèle du coût unitaire moyen par personnel novice.....	32
Tableau 3.9 – Modèle du coût unitaire moyen par plusieurs sous-échantillons de projets	33
Tableau 3.10 – Régression Simple : Modèles à une seule variable indépendante	34
Tableau 3.11 – Modèle de puissance : $Y = A * X^B$	39
Tableau 3.12 – Facteurs pouvant affecter la productivité des activités de maintenance	39
Tableau 3.13 – Catégories de la variable qualitative Expérience	41
Tableau 3.14 – Régression linéaire multiple :LogFFP, Experience1, Experience2 et LogEffort (Méthode « Stepwise »).....	42
Tableau 3.15 – Régression linéaire multiple :LogFFP, Experience1, Experience2 et LogEffort (Méthode « Direct »)	43
Tableau 3.16 – Catégories de la variable qualitative Complexité.....	45
Tableau 3.17 – Régression linéaire multiple : facteur « complexité fonctionnelle des projets »	45
Tableau 3.18 – Catégories de la variable qualitative Complexité après l'ajustement	49
Tableau 3.19 – Régression linéaire multiple : ajustement du facteur « complexité fonctionnelle »	52
Tableau 3.20 – Modèle linéaire multiple : ajustement du facteur « complexité fonctionnelle »	53

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

Cette section définit les différents acronymes et abréviations utilisés dans ce document.

Acronymes et Abréviations	Description
FFP	Full Function Point
FPA	Function Point Analysis
IFPUG	International Function Point Users Group
LRGL	Laboratoire de recherche en gestion des logiciels
PF	Point de fonction
PFE	Points de fonction étendus
UQAM	Université du Québec à Montréal
COSMIC	The Common Software Measurement International Consortium

RÉSUMÉ

Au cours des dernières années, quelques études dans le domaine du génie logiciel ont tenté de s'attaquer à des problèmes inhérents de la maintenance de logiciels afin de déterminer ou d'identifier des facteurs qui occasionnent l'augmentation des coûts ainsi que des prévisions inconsistantes des efforts et des échéanciers dans cette phase du cycle de vie de logiciel.

Ce projet s'insère dans une démarche chez ABC Inc. et a pour but principal d'établir un modèle d'estimation de productivité de la maintenance. Le moyen utilisé est la mesure de la taille fonctionnelle d'un échantillon des applications CENTREX avec la méthode FFP proposé par le Laboratoire de Recherche en Gestion des Logiciels (LRGL) de l'UQAM. Ce projet en plus de vérifier l'apport de la méthode FFP lors de la mesure d'un logiciel intégrateur (services en ligne avec outils linguistiques).

Afin d'atteindre ce but, on a entrepris certaines étapes telles que: la mesure de l'échantillon des applications CENTREX en utilisant la méthode de mesure des points de fonction étendus (COSMIC-FFP version 2.0 selon la norme ISO/IEC 14143-3 pour mesurer la taille fonctionnelle des projets), ensuite l'analyse statistique des données obtenues au moyen des divers tests statistiques et des critères de qualité et fiabilité de bons modèles de productivité et finalement l'établissement des modèles de productivité considérant la taille fonctionnelle, l'effort et les autres divers facteurs proposés dans la revue de littérature. Deux modèles de productivité ont été retenus : un modèle simple qui peut décrire la relation positive entre la taille fonctionnelle et l'effort (une fonction non linéaire: « Puissance ») et un modèle de régression multiple incluant comme variable supplémentaire la complexité fonctionnelle du projet en fonction de la taille fonctionnelle.

Les résultats obtenus nous ont permis de consolider les résultats des autres études empiriques réalisées récemment par des étudiants du LRGL dans le domaine de la maintenance. De plus, des incertitudes de la mesure dans ce nouveau domaine d'expérimentation (intergiciel) ont été surmontées permettant ainsi d'obtenir des résultats concluants sur l'efficacité de la méthode FFP dans ce domaine, ce qui pourra servir de point de départ pour d'autres recherches.

INTRODUCTION

De plus en plus, diverses compagnies et organisations gouvernementales développent ou maintiennent des logiciels afin de rendre plus efficaces leurs opérations et d'optimiser la mise en marché de leurs produits. Toutefois, des études ont démontré que les coûts annuels de maintenance sont trois à quatre fois ceux du développement d'applications nouvelles [SEL00]. Par conséquent, la maintenance des logiciels est aujourd'hui considérée comme un sujet difficile tout en étant critique dans l'industrie, et essentiel pour la mission de plusieurs organisations.

La maintenance est surtout un des moyens de tirer le meilleur profit du « capital logiciel » investi par les entreprises en prolongeant la durée de vie des applications et en leur permettant d'évoluer et de s'adapter aux besoins des utilisateurs.

Actuellement, des ingénieurs logiciels peuvent exécuter des tâches nécessaires pour concevoir et maintenir des logiciels dans une suite d'étapes logiques facilement identifiables et mesurables [AST96]. La mesure des étapes de développement des produits et des logiciels a une incidence sur le coût, l'échéancier, la qualité et la fonctionnalité d'une livrable. Cette quantification est donc un élément important à considérer pour répondre aux besoins particuliers des différents maillons de production de l'entreprise, alors qu'ils tentent de prédire, contrôler et améliorer la performance de chacune des activités menant à un produit fini. Il s'agit donc d'organiser la maintenance et de la gérer avec des critères de qualité et de mesure très rigoureux, afin d'apporter des améliorations aux processus logiciel, permettant ainsi aux gestionnaires d'atteindre leurs propres objectifs techniques et d'affaires.

Depuis la fin des années 1970, le génie logiciel propose des processus et des outils de mesure qui peuvent être utilisés pour favoriser des stratégies d'amélioration continue permettant de mesurer la productivité, d'estimer les coûts des projets, de contrôler des changements de la taille d'un logiciel, de mieux comprendre les étapes

de développement ou de maintenance d'un logiciel. Cependant, aucune de ces méthodes proposées n'a obtenu d'acceptation généralisée de la part de l'industrie à cause de leurs faiblesses inhérentes et de la non-portabilité de leurs résultats dans les divers environnements [LMAGL/SELAM].

Au cours des dernières années, des recherches ont été réalisées dans ce domaine, et une des méthodes de mesure de la taille fonctionnelle des logiciels ayant le meilleur taux de pénétration dans l'industrie est la méthode de l'Analyse des Points de Fonction (FPA). Cette méthode, conçue par Allan Albrecht en 1979 et dont l'évolution est de la responsabilité de « International Function Point Users Group » (IFPUG) à partir de 1984, est devenue un objet d'étalonnage pour mesurer la taille d'un logiciel.

La méthode des points de fonction permet de mesurer adéquatement la taille d'un logiciel, au moins pour les premières phases de son cycle de vie. Elle mesure la taille fonctionnelle des logiciels en considérant une vue fonctionnelle ou logique plutôt qu'une vue physique ou technique des systèmes. Ce sont ces caractéristiques qui permettent de quantifier rigoureusement la taille d'un logiciel en fonction des perspectives de l'utilisateur et de la rendre plus compréhensible pour les gestionnaires [MIK99].

Toutefois, diverses affirmations, souvent contradictoires, témoignent du fait qu'il existe un certain scepticisme dans l'application de la méthode des points de fonction à différents types de logiciels (c.-à-d. télécommunications, des systèmes en temps réel, des systèmes d'exploitation, des systèmes embarqués ou de contrôle, etc.). Une raison, parmi d'autres, qui se dégage de manière plus générale, est qu'un logiciel « technique » ne gère pas seulement une transaction de base d'entrée-sortie effectuée par un utilisateur, mais partage en plus des processus complexes de traitement des données élémentaires aux différents niveaux du logiciel, le problème est donc que ceci n'est pas correctement considéré par des modèles et des règles d'IFPUG [MEL97].

Par conséquent, il existe une nécessité croissante de mettre de l'avant une méthode dans le but de résoudre cette difficulté persistante lors d'une quantification de la taille fonctionnelle; méthode qui diffère donc des méthodes utilisées par une application plus typique de gestion d'information dans la première partie de son cycle de vie.

Une alternative est la nouvelle méthode de mesure de la taille fonctionnelle, proposée en 1997 par le professeur Alain Abran et ses collègues : la méthode de points de fonction étendus (FFP). Cette méthode considère les traitements internes des données à un niveau plus détaillé que ceux de FPA. Les processus transactionnels sont évalués pour les sous-processus, c.-à-d. non seulement les données qui entrent ou sortent d'un processus sont mesurées, mais aussi les processus internes de lecture et d'écriture des groupes de données. Mesurant toute la fonctionnalité qu'un processus requiert pour être considéré terminé plutôt que des aspects externes (à ce processus), ceci permet de compléter une description plus exhaustive de la fonctionnalité du processus pour des comparaisons d'estimation et de productivité [MOR98].

Dans ce projet, on aborde la mesure de la taille fonctionnelle d'un échantillon des applications CENTREX développées chez ABC Inc. en fonction de la méthode FFP (méthode des points de fonctions étendus COSMIC-FFP version 2.0 selon la norme ISO/IEC 14143-3) afin d'établir un modèle d'estimation de productivité de la maintenance (maintenance adaptative : "modifications pour adapter un logiciel aux nouveaux changements de ses exigences et de son environnement" [ABR93]) de ces applications pour des projets futurs. En plus, on vérifie l'apport de la méthode FFP lors de la mesure d'un logiciel intégrateur (services en ligne avec outils linguistiques).

Ce rapport présente le contexte du projet dans le premier chapitre. Ensuite, dans une deuxième chapitre, une description des données obtenues après la mesure est

présentée ainsi que l'évaluation des divers critères de fiabilité et de qualité d'un bon modèle de productivité sur lesquels s'inscrit l'analyse statistique de la banque de données. Dans un troisième chapitre, des modèles de productivité sont établis. Par la suite, des comparaisons avec d'autres études empiriques et des interprétations sont présentées dans un quatrième chapitre et finalement les résultats de cette recherche sont le sujet d'analyses et de recommandations pour de futurs projets.

CHAPITRE I

CONTEXTE DU PROJET

1.1 DESCRIPTION SOMMAIRE DE L'ORGANISATION

ABC Inc. est une organisation fondée à Montréal, Canada en 1994, qui se concentre sur le développement d'une solution novatrice franchissant les limites actuelles des logiciels de linguistique et facilite l'accessibilité aux services linguistiques sur Internet [ROC99].

Cette entreprise développe des technologies innovatrices intégrant les outils et les services linguistiques aux applications de bureau. Ces technologies permettent aux utilisateurs de s'exprimer en plusieurs langues.

ABC Inc. offre présentement son logiciel CENTREX qui est une architecture client-serveur fonctionnant dans un environnement multi-plateforme et permettant aux applications de texte d'utiliser des services fournis par un serveur local ou distant. CENTREX est un intergiciel, c'est-à-dire la couche permettant une communication transparente entre le client et le serveur. Ce logiciel sert de porte d'entrée aux produits de linguistique et aux services en ligne. Les utilisateurs peuvent, à partir d'une application de bureau traitant du texte, naviguer, accéder aux services et commander des produits additionnels disponibles sur Internet. [Plugiciels Windows, Site ABC Inc..com]

Un des principaux objectifs du logiciel CENTREX est d'intégrer parfaitement les moteurs de traduction, les outils de grammaire, les dictionnaires ainsi que les services en ligne dans n'importe quel document actif ou courrier. Ainsi, le logiciel

CENTREX est compatible actuellement à plusieurs applications de texte par l'entremise de « Plug-ins » développés tels que Microsoft Outlook™, Microsoft Word, Microsoft Excel, Microsoft PowerPoint, Adobe FrameMaker, Adobe Illustrator, Corel WordPerfect, Eudora Light , Netscape Communicator, Netscape Navigator et ICQ™. [ROC99]

1.2 SITUATION ACTUELLE DE L'ENTREPRISE

La compagnie ABC Inc. présente une absence de normalisation entre ses projets. La nature ambitieuse des produits impose la nécessité de gérer des critères de qualité et de mesures afin de contrôler et d'améliorer son processus d'estimation de durée de développement, d'évaluation et de maintenance de logiciels. Cette rigueur additionnelle est requise pour faire évoluer ses applications en tenant compte des besoins des utilisateurs et des percées technologiques récentes tout en maximisant la stabilité des exigences dans des propositions précises de leurs projets futurs sans dépasser les prévisions de coûts, d'efforts et d'échéanciers.

L'analyse des difficultés de la maintenance chez ABC Inc., met en évidence diverses origines résultant en les conséquences directes suivantes :

Tableau 1.1 – Diverses origines des difficultés de la maintenance

Diverses origines	Conséquences directes
1.- Absence de ressources standards :	
Il existe une carence de normes, des méthodes et d'outils, tant au moment de l'analyse, de la conception et de l'implémentation que lorsqu'il faut maintenir et faire évoluer les applications.	Défauts et insuffisances de documentation, ce qui complique la continuité et l'évolution des applications et augmente les coûts. Défauts de conception et d'erreurs de programmation qui génèrent des blocages et des dysfonctionnements.

2.- Objectifs faibles lors de la mesure de logiciels :	
<p>Il n'existe pas de données suffisantes qui permettent d'établir avec certitude le bassin d'expertises communiqué dans un projet de développement logiciel.</p> <p>Aucune évidence pour assurer que les données existantes étaient rassemblées selon les règles exactes lors de la définition d'une mesure.</p>	<p>L'absence de la définition formelle d'un processus de développement du logiciel rend la vérification et la mesure presque impossible puisque aucun étalon n'existe.</p> <p>Malgré que la mesure soit considérée comme nécessaire au moins pour évaluer le statut des projets, processus, produits et ressources logiciels chez ABC Inc., il n'existe pas un moyen formel qui permet aux analystes ou chefs de projets de connaître ce qui écarte un projet de ses objectifs.</p> <p>Dans cette situation, les chefs de projets doivent se limiter à un contrôle qualitatif basé sur leurs impressions en utilisant des méthodes plutôt intuitives que formelles.</p>

1.3 BESOINS À COMBLER POUR LA MAINTENANCE CHEZ ABC INC.

La compagnie ABC Inc. a besoin d'appliquer des méthodes et des outils qui lui permettront de mesurer efficacement la taille de ses logiciels pour estimer adéquatement la productivité de la maintenance.

Une évaluation de la taille des logiciels dès la première phase de son cycle de vie, est considérée comme un élément essentiel dans le calcul de son estimation de coûts, d'efforts et d'échéanciers lors de sa réalisation ou de sa maintenance. En vue de cela, ce projet permettra de mesurer la taille fonctionnelle d'un échantillon des applications CENTREX en utilisant la méthode des points de fonction étendus, proposée par le Laboratoire de Recherche en gestion des logiciels afin de construire et d'offrir des modèles de productivité explicatifs pour estimer les modifications futures aux applications CENTREX.

Afin de mettre en place cette méthode, un cadre de référence sera aussi présent. Cette structure référentielle devra identifier des mesures efficaces et objectives de la productivité des projets informatiques précédents dans le but de justifier et de valider l'efficacité et la qualité des modèles de productivité à estimer pour la maintenance des applications CENTREX.

1.4 BESOINS DU LABORATOIRE DE RECHERCHE EN GESTION DES LOGICIELS (LRGL)

Le principal domaine de recherche du LRGL concerne l'élaboration des méthodes et des outils de mesures pour l'industrie en génie logiciel permettant d'améliorer leurs processus décisionnels tout en répondant à leurs objectifs d'affaires.

Plusieurs des projets au LRGL sont aussi orientés vers la mise en pratique des technologies conçues et de leur adaptation aux divers contextes industriels en génie logiciel. Pour cette raison, le laboratoire de recherche en gestion des logiciels s'est intéressé à appliquer la méthode des points de fonction étendus lors de la mesure de la taille fonctionnelle d'un logiciel d'un type différent (intergiciel) de ceux mesurés précédemment. Cet intergiciel a une architecture multi-plateforme client-serveur permettant aux applications de texte d'utiliser des services fournis par un serveur local ou distant.

La mesure de ce logiciel permettra au LRGL de corroborer l'efficacité et l'adaptation de la méthode de points de fonction étendus dans un nouveau domaine. Les résultats permettront également d'augmenter sa gamme de divers contextes industriels et de données de références pour offrir des étalons aux entreprises qui souhaitent réaliser des comparaisons avec d'autres entreprises qui développent et maintiennent des logiciels dans des domaines similaires.

1.5 SITUATION DE L'INTERVENTION DANS L'ENTREPRISE

En vue de la situation présentée précédemment sur l'entreprise, il semble alors important pour les chefs de projets de pouvoir utiliser une structure commune pour mesurer leurs logiciels ainsi que d'utiliser des mesures qui permettent, non seulement d'affiner le positionnement des travaux réalisés par la compagnie ABC Inc., les uns par rapport aux autres, mais aussi, de pouvoir positionner l'entreprise par rapport à la communauté internationale. Ce dernier point exige alors d'utiliser des mesures reconnues au niveau international.

La mise en oeuvre de la méthode de points de fonctions étendus chez ABC Inc. permettra de mesurer des logiciels en fonction de leur taille fonctionnelle ainsi que de les analyser par rapport à d'autres variables quantitatives. L'utilisation de données historiques et pertinentes révélera également des procédures plus efficaces qui peuvent être rassemblés et utilisés pour l'amélioration du processus d'estimation de la maintenance chez ABC Inc..

En plus, la mesure et l'enregistrement des caractéristiques des projets satisfaisants et insatisfaisants seront importantes pour l'entreprise. Ceci permettra d'éviter l'exécution des projets sans avoir aucun contrôle sur eux tout en permettant d'analyser les tendances, l'impact des actions correctives et les changements des résultats précédents lors du développement d'un logiciel.

1.6 DÉFINITION ET OBJECTIFS DU MANDAT

Tel que mentionné précédemment, l'objectif du projet est de pouvoir établir des modèles de productivité servant à l'estimation de la maintenance au sein de la compagnie ABC Inc.; ceci en appliquant la méthode de points de fonction étendus à un échantillon des applications CENTREX afin de mesurer leur taille fonctionnelle. Parallèlement, le projet éprouvera la valeur de la méthode de mesure FFP pour quantifier la taille fonctionnelle d'un logiciel avec un niveau de fiabilité acceptable.

En outre, ce projet tentera de soutenir les résultats existants au LRGL sur des recherches qui illustrent l'efficacité de la méthode des points de fonction étendus pour mesurer la relation positive entre l'effort et la taille fonctionnelle dans divers contextes de logiciels, et en particulier dans le domaine d'une architecture multi-plateforme.

De plus, ce projet permettra à l'étudiant de répondre à une des exigences requises à l'UQAM pour l'obtention de son diplôme dans le programme de maîtrise en génie logiciel.

1.7 LIMITES DU MANDAT

L'envergure de ce projet est strictement limitée à la mesure fonctionnelle d'un échantillon des applications CENTREX afin de bâtir un ou des modèles de productivité pour leur maintenance. Bien que les données recueillies ne correspondent qu'à ce groupe d'applications, les résultats pourraient servir de base pour comparer les diverses applications de l'entreprise entre elles, ainsi que dans une activité de recherche plus générale de la méthode de points de fonction étendus.

Bien que la mesure de ces applications sera documentée lors de la réalisation du projet, il est nécessaire de suivre et d'améliorer un programme de mesure, ce qui n'est pas compris dans ce mandat. L'implantation d'un programme de mesure sera important pour que les futures prises de décisions bénéficient des données historiques en comparant la productivité des diverses techniques et technologies mises en place chez ABC Inc. Notons également qu'il faudra mettre en place un processus de vérification permettant de s'assurer que le processus de quantification soit utilisé et maintenu au sein de l'entreprise.

1.8 APPROCHE DE RÉALISATION

L'approche de réalisation à suivre lors de l'exécution de ce mandat est définie par étape. Chacune des étapes sera validée de manière rigoureuse afin de respecter nos engagements et assurer le succès du projet. De plus, l'envergure et le contenu de chaque livrable à la fin de chaque étape seront examinés par les intervenants du projet afin d'en identifier et d'extraire les redondances et les omissions.

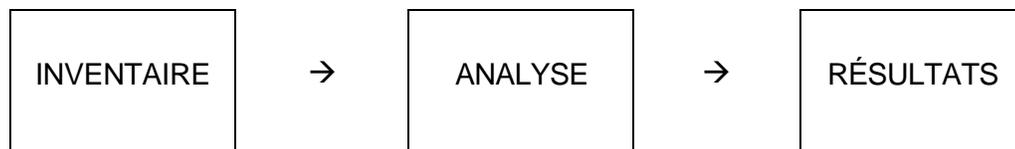


Figure 1.1 – Phases dans le projet

Étape 1 : Phase Inventaire

Rechercher dans la littérature des éléments de référence sur la méthode des points de fonction étendus, inventorier et obtenir toute la documentation disponible au sein de l'entreprise sur les diverses fonctionnalités qui doivent être livrées par l'échantillon des applications CENTREX. Ce travail permettra d'identifier et de mesurer (en appliquant la méthode des points de fonction étendus) les fonctionnalités qui pourraient intervenir lors de l'estimation du processus d'entretien des applications CENTREX. Les extraits de cette phase constituent les intrants principaux de la phase suivante.

La mesure des fonctionnalités de chacune des applications sera faite et évaluée par le mandataire du projet, avec l'aide des responsables des applications.

À la fin, une grille sommaire sera produite pour les fonctionnalités inventoriées de l'échantillon des applications CENTREX.

Étape 2 : Phase Analyse

Étudier la documentation pertinente rassemblée lors de la phase d'inventaire, puis analyser les données recueillies lors de la mesure fonctionnelle des applications.

Ceci permettra d'évaluer les modèles d'estimation et de productivité pour la maintenance des applications CENTREX.

En outre, identifier les principaux aspects intervenant dans les estimations précédentes de productivité et qui peuvent affecter la maintenance en écartant l'entreprise de ses objectifs. Selon certains chercheurs dans le domaine du génie logiciel, plusieurs facteurs peuvent intervenir dans des modèles de productivité en améliorant la qualité de la relation entre la taille fonctionnelle et l'effort [MAY95].

Étape 3 : Présentation des résultats

Proposer un ou des modèles descriptifs de productivité pour la maintenance des applications CENTREX, qui peuvent répondre d'une manière satisfaisante aux besoins de la compagnie.

Des analyses statistiques seront appliquées afin de vérifier la qualité de la relation entre les variables proposées dans les modèles de productivité; variables identifiées lors de l'utilisation de la technique de la régression linéaire à une seule variable indépendante et/ou multiples variables. Ceci nous permettra aussi de corroborer l'influence de certains facteurs de productivité sur la relation entre la taille fonctionnelle et l'effort.

1.9 OBJECTIFS DE L'ENTREPRISE ET MESURES PRÉVUES POUR LEUR ATTEINTE

Dans le cadre de leur initiative d'amélioration du processus d'estimation pour sa maintenance, l'entreprise s'est fixé comme objectifs :

- Améliorer leurs capacités pour l'estimation des logiciels afin d'offrir des propositions précises de projets, d'éviter des dépassements de coûts et d'échéanciers, et de maximiser la stabilité des exigences lors de la réalisation de leurs projets.
- Améliorer leur productivité en réduisant ou stabilisant les divers niveaux du personnel de l'entreprise au moment de l'analyse des divers facteurs intervenant dans la productivité.

Les mesures prévues pour atteindre les objectifs de l'entreprise définis dans ce mandat sont les suivantes:

Tableau 1.2 – Mesures prévues pour atteindre les objectifs de l'entreprise

Mesures Prévues	Intervenant
Sélectionner l'échantillon des applications CENTREX à analyser et mesurer	Chef du projet
Évaluation de chaque fonctionnalité des applications CENTREX sélectionnées.	Mandataire du projet et une ressource responsable de leur développement
Appliquer la méthode des points de fonction étendus pour mesurer la taille fonctionnelle des applications constituant l'échantillon.	Mandataire du projet
Planifier des rencontres afin de recueillir toutes les données historiques pertinentes des diverses applications.	Mandataire du projet et le chef de projet

Étant donné les diverses démarches existantes en génie logiciel pour mesurer les attributs d'un logiciel (par exemple, la taille d'un logiciel peut être mesurée par des lignes du code, des points de fonction, etc.), les mesures à utiliser dans ce mandat seront sélectionnées en fonction de leur utilité pour la construction des modèles d'estimation appropriés de productivité lors de la maintenance des applications. Celles-ci incluent ce qui suit :

Tableau 1.3 – Attributs et métriques pour construire des modèles d'estimation de productivité

Attributs	Mesures
Taille	Taille fonctionnelle FFP
Effort	Person hours (PH)
Complexité	Catégorie du logiciel
Niveau d'expertise du développeur	Junior, senior

1.10 L'APPORT DU PROJET AU GÉNIE LOGICIEL

Dans le domaine du génie logiciel, les mesures de la taille fonctionnelle sont considérées comme essentielles pour quantifier le produit logiciel ainsi que son processus de développement et d'entretien [ABR97]. Par exemple, une mesure de la taille fonctionnelle d'un logiciel, mesurée indépendamment des technologies, est reconnue comme un aspect clé dans l'étalonnage et l'utilisation des modèles de productivité en génie logiciel [OLI99].

Toutefois, l'immatunité constatée par diverses études dans certaines sous-disciplines de ce domaine, exige de les cerner et les classer mieux afin de faire du génie logiciel une discipline plus solide [HIL99] [BOU99]. Ceci a empêché d'établir un standard dans l'industrie logiciel qui réponde adéquatement aux besoins des diverses entreprises lors de la mesure d'un logiciel. Ceci est vrai plus particulièrement pour les petites et moyennes entreprises, souvent beaucoup plus dynamiques mais moins rigoureuses en matière d'élaboration et d'utilisation de mesures quantitatives.

Un apport de ce projet dans le domaine du génie logiciel sera d'évaluer la méthode des points de fonction, proposée par le LRGL dans un domaine différent où elle n'a pas été vérifiée auparavant afin de corroborer son efficacité lors de la mesure de la taille fonctionnelle d'un logiciel. Ceci augmentera l'échantillon de données existant au LRGL sur la méthode afin de permettre des comparaisons plus significatives entre diverses entreprises et de lui amener des améliorations pour la rendre plus révélatrice et solide dans la culture du génie logiciel de l'industrie.

CHAPITRE II

ANALYSE DES DONNÉES

L'analyse statistique de données a comme but d'atteindre une conclusion solide à partir d'un échantillon limité de données. (Il est utile de mentionner que l'échantillon retenu dans cette étude est composé des diverses fonctions mesurées et réalisées par trois analystes programmeurs différents ayant une connaissance variée du domaine des affaires et des applications).

Une conclusion solide dépendra de deux facteurs qu'il est nécessaire de surmonter:

- Des différences importantes peuvent être obscurcies par la variabilité d'une population et l'imprécision expérimentale. Ceci rend difficile de distinguer la vraie différence de la variabilité aléatoire. Le cerveau humain essaie de trouver des modèles, même pour des données aléatoires. Notre inclination naturelle (particulièrement avec nos propres données) est de conclure que les différences sont vraies, et de réduire au minimum la contribution de la variabilité aléatoire.
- Des analyses statistiques sont nécessaires lorsqu'il existe des différences mineures entre l'imprécision expérimentale et la variabilité d'une population. Ceci nous aiderait donc à surmonter les facteurs mentionnés précédemment permettant ainsi de valider statistiquement nos modèles. La plupart des modèles statistiques utilisent les données d'échantillon afin de déterminer la valeur et la signification statistique de leurs coefficients. Il s'agit en effet de représenter une réalité de nature aléatoire au moyen de la confrontation de l'observation et du modèle ainsi que de déduire les lois ou certains éléments des lois qui gouvernent cette réalité [MOT99].

Un des outils statistiques utilisés amplement dans le domaine du génie logiciel pour représenter une relation entre la taille d'un logiciel et l'effort est l'analyse de régression. Celui-ci s'appuie sur une méthode simple pour établir une relation fonctionnelle parmi des variables. Il est cependant essentiel de définir et de caractériser les modèles avant de les exploiter.

2.1 HYPOTHÈSES POUR EXTRAPOLER LES RÉSULTATS D'UN ÉCHANTILLON DE DONNÉES À LA POPULATION

Afin de s'assurer que la qualité de la relation entre la taille fonctionnelle et l'effort (relation utilisée comme la base d'un ou de plusieurs modèles d'estimation de productivité) fournira des résultats satisfaisants pour la maintenance chez ABC Inc. à partir d'un échantillon de la population (données recueillies). Il y a plusieurs exigences qui doivent être adressées pour soutenir (supporter) cette recherche :

- *Assurer que la population suit une distribution gaussienne approximative.* Cela nous permet d'utiliser des essais statistiques, qui nous permettront de faire des inférences au sujet de la moyenne et d'autres propriétés de la population. Cependant, il existe des désavantages reconnus lorsqu'une approche statistique est utilisée, telle que la possibilité de produire des résultats valides seulement pour cet environnement. Également, le fait d'introduire plusieurs variables additionnelles dans un modèle requiert l'augmentation de la quantité de données.
- *Considérer l'échantillon suffisamment long afin de valider le théorème de la limite centrale,* lequel mentionne que la distribution de la moyenne suivra une distribution gaussienne même si la population n'est pas gaussienne.

Pour dépasser les limitations statistiques, c'est-à-dire pour étendre les inférences plus loin que celle de l'échantillon, il faut utiliser un jugement scientifique et un bon sens commun; car la logique statistique constitue seulement une partie de l'interprétation des données.

2.2 VALIDATION DE LA DISTRIBUTION NORMALE DES VARIABLES (DISTRIBUTION GAUSSIENNE APPROXIMATIVE)

En utilisant l'outil WinSTAT, on a une méthode qui peut être employée pour supporter l'hypothèse que les données sont distribuées normalement. Le test de Kolmogorov Smirnov calcule la distance maximale entre la courbe de fréquence cumulative des données et la courbe qui représente le mieux possible la distribution normale, ainsi que la détermination de la signification de cette distance. Le test de

Kolmogorov-Smirnov est mieux utilisé sur des variables continues, c'est-à-dire des variables qui prennent leur valeur dans des intervalles de l'ensemble des nombres réels.

Tout d'abord, l'étude de la distribution de la variable indépendante, *la taille fonctionnelle* et de la distribution de la variable dépendante, *l'effort*, nous a révélé des distributions de fréquences un peu éloignées d'une distribution normale. On pourrait donc penser que la relation entre l'effort et la taille fonctionnelle n'est pas linéaire, ce qui a été illustré par divers échantillons empiriques, dont la base de données de [BOE81].

On a ensuite décidé de faire une transformation logarithmique de ces variables afin d'obtenir la linéarité dans le modèle de productivité à établir. Ceci respecte une des hypothèses standards de l'analyse de régression linéaire, laquelle présume que le modèle qui décrit les données est linéaire dans les variables. Ainsi, dans notre cas, les variables LogFFP et LogEffort sont distribuées continues, et le test de Kolmogorov-Smirnov rapporte les valeurs suivantes pour le paramètre statistique analysé « p-value » :

Tableau 2.4 – Test de distribution normale pour les variables LogFFP, LogEffort

Test of Normal Distribution			
Kolmogorov-Smirnov test for continuous variables			
	N	D	P
Log(FFP)	15	0,207180649	0,540230863
Log(Effort)	15	0,23011548	0,404948053

Une valeur base de "p" ($p < 0.05$) indiquerait des résultats significatifs, donc on conclue ici que la différence entre les deux courbes n'est pas significative parce que les valeurs des variables Log(FFP) et Log(Effort) sont supérieures à celle de $p < 0.05$. Alors, on peut supposer que les données sont distribuées normalement.

2.3 HISTOGRAMMES ET DISTRIBUTION CUMULATIVE DE FRÉQUENCES

Une image de la distribution d'une variable peut être obtenue en traçant la distribution cumulative de fréquences. Cette méthode évite la rupture ou l'analyse en classes qui étaient nécessaires pour un histogramme.

Une première analyse de la distribution des variables LogFFP et LogEffort est faite à partir des graphiques suivants: histogramme et distribution cumulative de fréquence. Comme on peut observer dans les résultats suivants, la distribution de fréquences des observations ne suit pas exactement une distribution normale. Cependant, selon les références statistiques de l'outil WinSTAT en utilisant le test Kolmogorov-Smirnov, on peut considérer ces distributions comme normales à cause de la valeur obtenue du paramètre statistique "p" dans le Test de la distribution Normale. Ceci nous permet d'assurer que les points correspondants aux données distribuées normalement, seront alors groupés autour de la moyenne selon la courbe gaussienne en forme de cloche. En raison de la nature fixe de la courbe, la probabilité qu'un point des données tombe au-delà d'une distance donnée de la moyenne peut être calculée exactement, et c'est sur ce calcul que les méthodes statistiques utilisées dans notre cas d'étude sont basées.

Frequencies

Log (FFP)	Frequency	Percent	Cumulative Percent
0,75 to 1	1	6,67	6,67
1 to 1,25	0	0,00	6,67
1,25 to 1,5	0	0,00	6,67
1,5 to 1,75	2	13,33	20,00
1,75 to 2	1	6,67	26,67
2 to 2,25	4	26,67	53,33
2,25 to 2,5	5	33,33	86,67
2,5 to 2,75	2	13,33	100,00

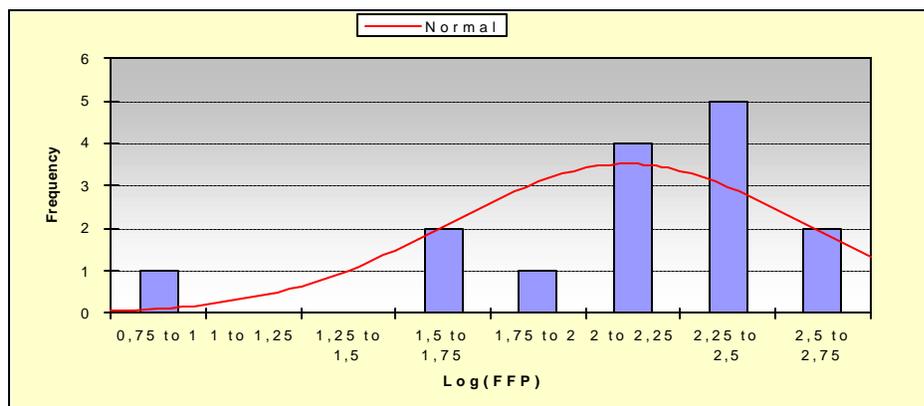


Figure 2.2 – Histogramme de la variable LogFFP

Cumulative Frequencies

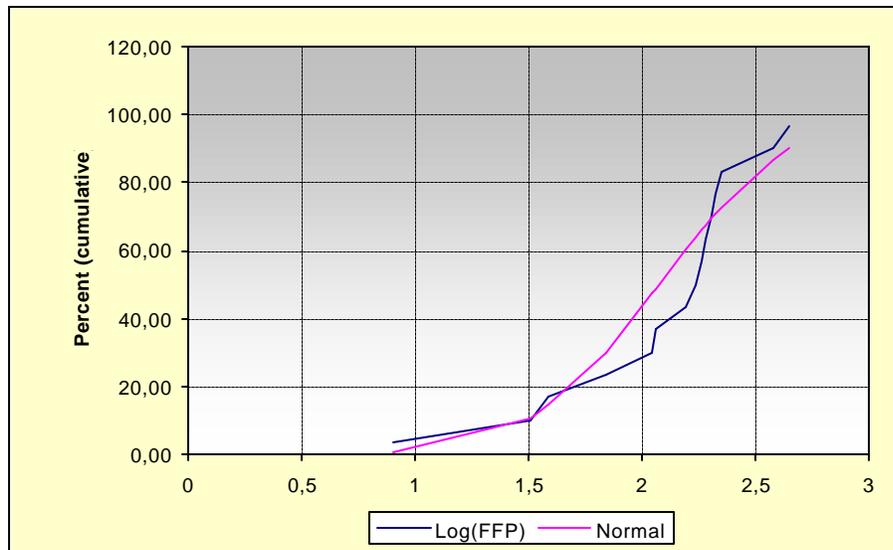


Figure 2.3 – Distribution cumulative de fréquence de la variable LogFFP

Frequencies

Log(Effort)	Frequency	Percent	Cumulative Percent
0,5 to 1	1	6,67	6,67
1 to 1,5	1	6,67	13,33
1,5 to 2	2	13,33	26,67
2 to 2,5	8	53,33	80,00
2,5 to 3	3	20,00	100,00

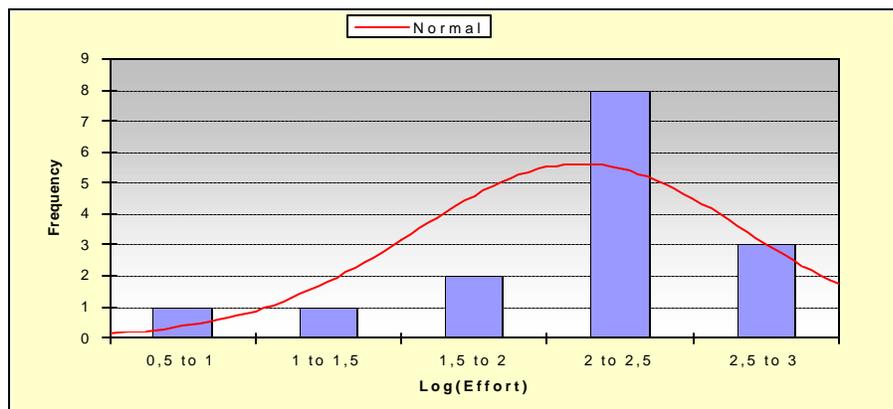


Figure 2.4 – Histogramme de la variable LogEffort

Cumulative Frequencies

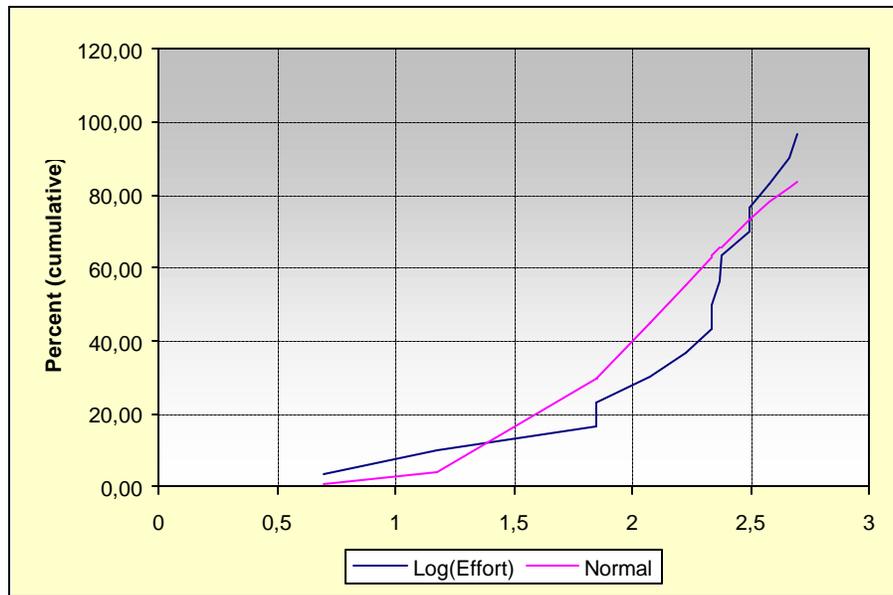


Figure 2.5 – Distribution cumulative de fréquence de la variable LogEffort

2.4 CRITÈRES DE FIABILITÉ POUR UN MODÈLE DE PRODUCTIVITÉ

Des critères statistiques de fiabilité sont amplement employés dans divers domaines afin d'extraire des paramètres pertinents et d'analyser les fluctuations au moyen des outils de modélisation et de traitements mathématiques. Ceci est dû en particulier aux exigences de qualité et de sûreté qui sont liées d'une part aux demandes des clients et à la compétitivité, d'autre part au développement des technologies de plus en plus sophistiquées.

On peut distinguer dans cette étude deux grandes approches lors de l'utilisation de critères de fiabilité pour un modèle de productivité: l'analyse du retour d'expérience qui permet de déterminer les paramètres caractéristiques de sûreté du produit étudié (le modèle reflète les données), et l'analyse prévisionnelle qui donne des indications sur le comportement du produit à partir des caractéristiques de ses composants et

de leurs interactions (le modèle prédit les données) [<http://www-math.univ-mlv.fr/math/DESS.html> DESS de Mathématiques Appliquées Méthodes Statistiques et Numériques].

Toutes les remarques que l'on fait plus bas sur la validation des critères de fiabilité utilisés sont considérés comme valables dans la littérature, mais il faudra cependant être très prudent dans l'interprétation de chacun d'eux lors de l'étude.

La démarche statistique suivie dans cette étude considère les critères suivants pour déterminer la qualité des modèles de productivité à proposer pour les activités de maintenance chez ABC Inc. : la qualité d'ajustement du modèle aux données, indiquée par les paramètres R^2 , $\text{Adj-}R^2$ et l'écart-type (standard error) ainsi que l'indication de l'exactitude du modèle par le paramètre $\text{Pred}(X)$ [SIL00].

2.4.1 Paramètres indiquant la qualité d'ajustement du modèle aux données

Coefficient de détermination (R^2) : C'est une mesure globale du pourcentage d'explication du modèle qui est liée au gain de variance sur la variable prédictive utilisée dans le modèle de régression linéaire. Autrement dit, le coefficient de détermination (R^2) est une mesure globale de la qualité d'un modèle donné. La valeur de ce coefficient varie entre 0 et 1; donc un R^2 de près de 1 indique que presque toute la variabilité de la réponse de la variable prédictive est expliquée par le modèle, c'est-à-dire qu'il existe une forte relation entre la variable indépendante et la variable dépendante.

R^2 Ajusté ($\text{Adj-}R^2$) mesure la diminution de variance obtenue grâce au modèle de régression. C'est un indicateur plus réaliste de la qualité d'ajustement du modèle car celui-ci est ajusté par le nombre de paramètres dans le modèle. R^2 ajusté est toujours moins que R^2 [WEI85]. $\text{Adj-}R^2$ est représenté par l'équation 2.1: n est la quantité des variables et $n - k - 1$ est le degré de liberté, c'est-à-dire le nombre d'observations moins le nombre de variables indépendantes moins un :

$$\text{Adj-R}^2 = 1 - \frac{n-1}{n-k-1} * (1- R^2) \text{ (équation 2.1)}$$

En outre, les coefficients des variables indépendantes dans les modèles de productivité sont évalués en utilisant le paramètre statistique *P value*. Cette valeur nous indique que le résultat obtenu a une probabilité de p% d'être observé s'il n'existe pas de liaison entre la variable indépendante étudiée et la variable prédictive. L'influence significative des variables indépendantes dans les modèles de productivité est déterminée lorsqu'elles sont inférieures au seuil habituel de 0.05 (5%) [WEI85].

2.4.2 Paramètres indiquant l'exactitude du modèle

On a choisi d'introduire simultanément les autres indicateurs suivants :

L'erreur relative moyenne (ERM), qui signale l'importance de l'écart en pourcentage, entre les valeurs estimées par un modèle et les données réelles. Des valeurs estimées parfaites donneraient une ERM de zéro. Cependant, selon Conte et al.(1986) une ERM égale ou plus petite que 25% serait acceptable [MAY95] :

$$ERM = \frac{1}{n} \sum_{i=1}^n \left| \frac{\text{Valeur}_{\text{estimee}i} - \text{Valeur}_{\text{reelle}i}}{\text{Valeur}_{\text{reelle}i}} \right| \text{ (équation 2.2)}$$

Qualité de la prédiction du modèle : Selon l'approche classique du génie logiciel, un modèle de productivité est considéré bon s'il est capable de rencontrer le critère de l'erreur relative moyenne (Magnitude Relative Error) de +/- 25% pour 75% des observations, ceci est aussi connu comme la fiabilité de la prédiction [CON86], [VER81]. Ces chercheurs emploient cette notion pour définir une mesure de qualité de la prédiction du modèle en définissant qu'une évaluation technique peut être acceptable si PRED (0.25) est au moins 0.75.

En plus, d'autres auteurs (Kemerer 1987, Miyazaki 1985, Gulezian 1991, Helm 1992) considèrent comme acceptable une erreur relative moyenne de +/- 20% dans 70% des observations [ABR93-2].

CHAPITRE III

DÉTERMINATION D'UN MODÈLE DE PRODUCTIVITÉ

En premier lieu, on cherche à expliquer la relation simple entre la variabilité de la variable effort et la variable taille fonctionnelle des projets dans cette étude. Ces variables sont exprimées en personnes-heures nécessaires pour le développement d'un projet (variable effort), et par leur nombre de points de fonctions étendus pour la variable taille fonctionnelle.

On écrit cette relation sous la forme des deux types de modèles utilisés : le modèle du *coût unitaire moyen*, qui est décrit comme le rapport entre l'effort et le nombre des points de fonction ainsi que le *modèle statistique de la régression linéaire* avec les points de fonction étendus (régression simple) et les divers facteurs qui peuvent influencer la productivité des activités de maintenance (régression multiple) [SIL00].

3.1 MODÈLE DU COÛT UNITAIRE MOYEN

Le coût unitaire moyen est défini sous la forme suivante :

$$\text{Coût unitaire moyen} = \frac{1}{n} \sum_{i=1}^n \frac{\text{Effort}_i}{\text{FFPi}} \quad (\text{équation 3.1})$$

Ce modèle correspond au cas où nous analyserions les variables FFPi et Efforti afin de déterminer l'effort moyen nécessaire pour produire un point de fonction étendu, autrement dit, le nombre moyen d'heures par FFP dans notre cas.

Un modèle de productivité construit à partir du coût unitaire moyen a donc la forme suivante :

$$\text{Effort} = \text{FFP} * \text{coût unitaire moyen (équation 3.2)}$$

Après avoir effectué le calcul pour notre échantillon (15 projets), le coût unitaire moyen était de 1.47 heures par point de fonction étendu (FFP). Le modèle de productivité proposé à partir de ce coût unitaire moyen a donné le résultat suivant :

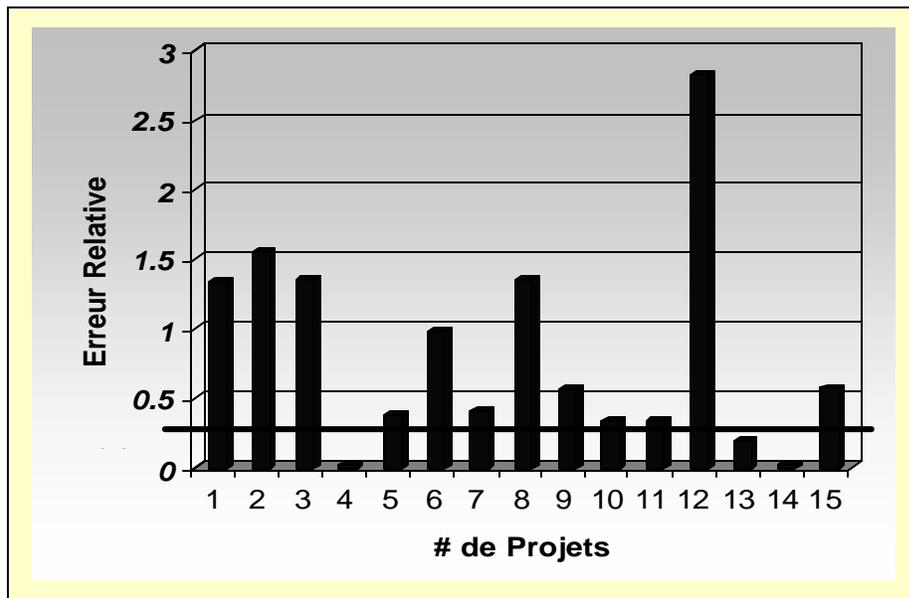


Figure 3.6 – Erreur relative de l'ensemble de projets

Seulement les projets 4 (Service), 13 (Gestionnaire CENTREX) et 14 (Find service) ont une erreur relative inférieure à 25%

Tableau 3.5 – Modèle du coût unitaire moyen par l'ensemble de projets

Équation	ERM(%)	PRED(+/- 25%)	
		# projets à l'intérieur	%
Effort = 1.47 * FFP	81.28	3	20

3.1.1 Qualité de ce modèle de productivité

Dans l'analyse des résultats obtenus, il faut regarder en premier lieu les divers critères utilisés pour déterminer la qualité de ce modèle. On observe que l'ERM (erreur relative moyenne) de 81.28% est supérieure à celle de 25% proposée par Conte et al.(1986) dans la littérature. Cette erreur obtenue est très élevée par rapport au critère acceptable défini par ces auteurs, lesquels suggèrent une ERM égale ou plus petite que 25%. En plus, seulement 3 projets (20% de l'échantillon) sont à l'intérieur de l'écart de fiabilité des estimations de +/- 25% par rapport aux efforts réels. En conséquence, cette probabilité est faible, donc ce modèle pour notre échantillon d'observations ne donne pas des résultats intéressants et utiles pour un modèle d'estimation des efforts basé sur la seule variable de la taille fonctionnelle.

3.1.2 Modèles avec sous-échantillon

Selon la littérature, une autre possibilité est offerte, celle de séparer et d'observer l'échantillon disponible par sous-groupes de projets qui ont des caractéristiques similaires, ceci afin de voir s'il est possible d'obtenir des modèles de productivité plus fiables et satisfaisants [MAY95]. Le critère sélectionné pour diviser l'échantillon dans notre étude concernait le personnel informatique impliqué dans les projets : expérience en informatique et avec le logiciel.

3.1.2.1 Sous-échantillon de projets développés par du personnel expérimenté

Les applications impliquées dans cette section sont : Gestionnaire CENTREX, Find Service et DDE (3 projets). Elles ont été développées dans le même langage de programmation C++ avec un niveau de complexité fonctionnelle haute par un programmeur ayant une expérience informatique avancée. Bien que le programmeur ait la même expérience en langage de programmation et dans le domaine fonctionnel, il ne maîtrisait pas parfaitement la méthode FFP pour mesurer la taille fonctionnelle de chacune des applications. Donc, toutes les fonctions des applications ont été mesurées par l'étudiante mandataire (avec une connaissance de la méthode FFP mais ayant une connaissance superficielle du domaine du logiciel) avec l'aide du programmeur de l'application mesurée qui apportait des précisions sur les spécifications fonctionnelles.

Le modèle du coût unitaire moyen (projets développés par du personnel expérimenté) était de 2.24 heures par point de fonction étendu (FFP) et peut être représenté de la manière suivante :

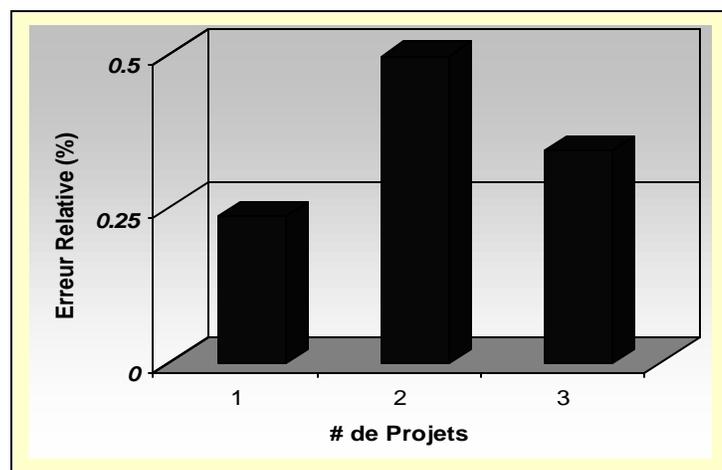


Figure 3.7 – Erreur relative de chaque projet – Personnel expérimenté

Ce modèle de productivité présente une ERM égale à 35.57%, ce qui est supérieur à 25% et la capacité de prédiction pour l'intervalle de +/-25% pour rapport aux efforts réels est 33.3%. L'imprécision de ce modèle est probablement dû à l'échantillon analysé qui comprend très peu de projets (seulement trois projets ayant les caractéristiques précédentes), ce qui ne permet pas de donner des résultats plus concluants.

Tableau 3.6 – Modèle du coût unitaire moyen par personnel expérimenté

Équation	ERM(%)	PRED(+/- 25%)	
		# projets à l'intérieur	%
Effort = 2.24 * FFP	35.57	1	33.3

3.1.2.2 Sous-échantillon de projets développés par du personnel intermédiaire

Les applications impliquées dans ce sous-échantillon sont : Classe, Resource, Constant, Service, Localisation, Targets (six projets) avec une complexité fonctionnelle variée : *Haute, Moyenne et Basse*. Elles ont été aussi développées dans le même langage de programmation C++ par un programmeur ayant une expérience informatique intermédiaire. Cependant, le programmeur maîtrisait favorablement la méthode FFP pour mesurer les points de fonctions de chacune des applications ayant des complexités fonctionnelles différentes. Cela lui a permis d'appliquer ses connaissances plus précises lors de la mesure des spécifications fonctionnelles des applications avec l'étudiante mandataire (connaissant la méthode FFP mais ayant une connaissance superficielle du domaine du logiciel). Les résultats des mesures de la taille fonctionnelle et du temps dépensé lors de la réalisation de chaque application étaient plus consistants. Cependant, cela ne garantit pas que l'effort requis pour faire une modification au niveau fonctionnelle dans les applications soit semblable.

Le modèle du coût unitaire moyen dans le sous-échantillon de projets développés par du personnel intermédiaire peut être représenté par la forme suivante :

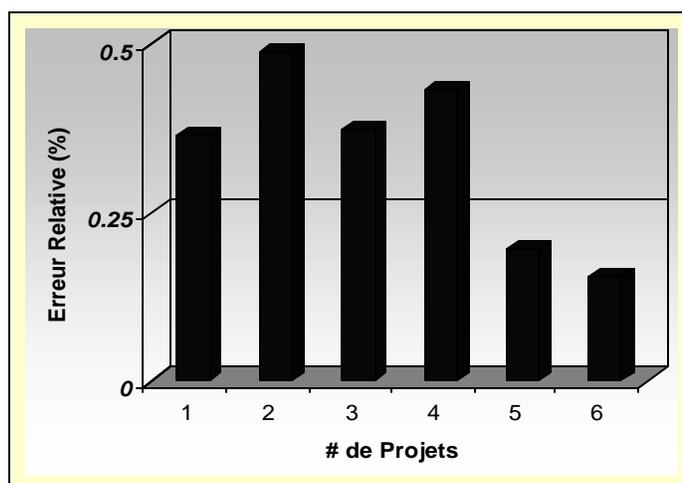


Figure 3.8 – Erreur relative de chaque projet – Personnel intermédiaire

Ce modèle de productivité présente une ERM égale à 33.45%, ce qui est légèrement supérieur à celle de 25% . Ce résultat est plus près de cette limite de 25% si on le compare avec le modèle précédent. Ce résultat est probablement dû au fait que les projets impliqués dans cette section ont été développés par un même analyste-programmeur avec plus de consistance dans ses procédures de développement. Mais seulement 33% des efforts estimés sont à l'intérieur d'un intervalle de +/-25% par rapport aux efforts réels. Ce modèle ne donne pas dans l'ensemble des résultats satisfaisants.

Tableau 7 – Modèle du coût unitaire moyen par personnel intermédiaire

Équation	ERM(%)	PRED(+/- 25%)	
		# projets à l'intérieur	%
Effort = 0.86 * FFP	33.45	2	33.33

3.1.2.3 Sous-échantillon de projets développés par du personnel novice

Les applications impliquées dans ce sous-échantillon sont : Plug-in Outlook, AutoSetup Outlook, Plug-in FrameMaker, AutoSetup FrameMaker, Plug-in WordPerfect, AutoSetup WordPerfect (six projets). Elles ont aussi été développées dans le même langage de programmation C++ par un programmeur ayant une expérience informatique novice.

Le projet concernant le Plug-in WordPerfect demandait au programmeur plus de temps et de connaissance du domaine de l'application lors de sa réalisation. Ceci a occasionné moins de consistance des mesures qui représentaient la relation d'effort et de la taille fonctionnelle pour cette application.

Le programmeur novice également ne maîtrisait pas parfaitement la méthode FFP pour mesurer les points de fonction de chacune des applications. Mais toutes les fonctions des applications ont été mesurées par l'étudiante mandataire (connaissance de la méthode FFP mais ayant une connaissance superficielle du domaine) avec l'aide du programmeur.

Le modèle du coût unitaire moyen par personnel novice peut être représenté sous la forme suivante :

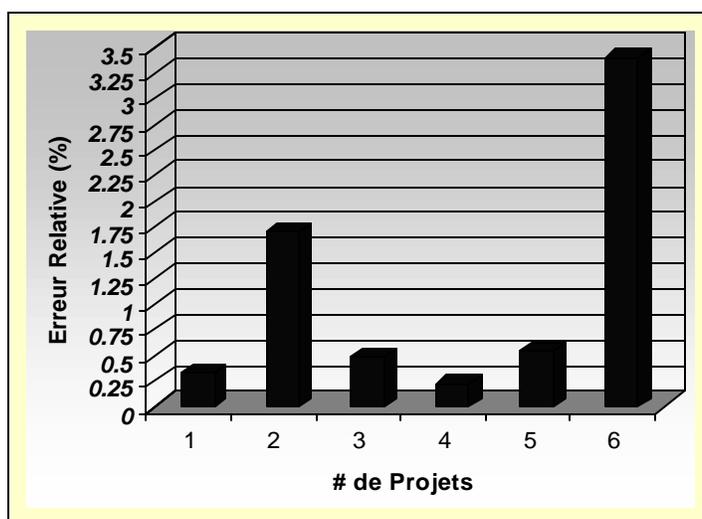


Figure 3.9 – Erreur relative de chaque projet – Personnel novice

Ce modèle de productivité présente une ERM très élevée égale à 111.64% et seulement 16.66% des efforts estimés sont à l'intérieur d'un intervalle de +/-25% pour rapport aux efforts réels. Ce modèle ne donne pas de résultats satisfaisants.

Tableau 3.8 – Modèle du coût unitaire moyen par personnel novice

Équation	ERM(%)	PRED(+/- 25%)	
		# projets à l'intérieur	%
Effort = 1.69 * FFP	111.64	1	16.66

Dans cette étude, on peut conclure que les modèles construits à partir du coût unitaire moyen n'atteignent pas les critères de bons modèles de productivité. Cependant, ils pourraient être susceptibles d'améliorations avec une augmentation de la quantité d'observations pour chaque type de sous-échantillons.

3.1.2.4 Comparaison des sous-échantillons d'observations et l'ensemble de projets

Si on analyse les critères de fiabilité du modèle du coût unitaire moyen de l'ensemble de projets par rapport aux critères de fiabilité des modèles de coûts unitaires moyen des applications développées par un personnel type (tableau 3.5), on remarque que les résultats sont très semblables : des ERM très élevés et la capacité de prédiction de chacun des modèles est très basse. D'autre part, un des sous-échantillon qui comportaient plus d'observations (six applications développées par personnel novice) a signalé que le niveau d'expérience en informatique est un facteur important et qui peut occasionner que l'effort requis pour changer un point de fonction soit différent dans chacune des sous-échantillon.

Le tableau suivant résume de façon succincte les critères de fiabilité obtenus des différents modèles du coût unitaire moyen pour applications d'un personnel type et pour l'ensemble de projets.

Tableau 3.9 – Modèle du coût unitaire moyen par plusieurs sous-échantillons de projets

Applications	Équation	ERM(%)	PRED(+/- 25%)	
			# projets à l'intérieur	%
Ensemble de projets	Effort - 1.47*FFP	81.28	3	20
Personnel expérimenté	Effort = 2.24*FFP	35.57	1	33.33
Personnel intermédiaire	Effort = 0.86*FFP	33.45	2	33.33
Personnel novice	Effort=1.69*FFP	111.64	1	16.66

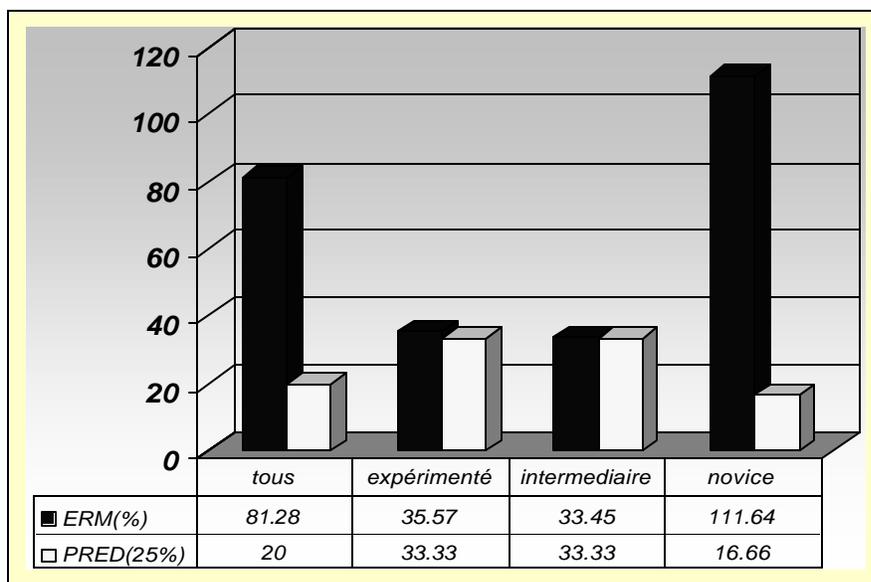


Figure 3.10 – Erreur relative de chaque sous-échantillon de projets et de l'ensemble de projets

3.2 MODÈLE DE RÉGRESSION LINÉAIRE

Le modèle statistique le plus simple à utiliser et à comprendre après le coût unitaire est le modèle de régression linéaire simple. C'est le modèle qui correspond au cas où nous avons qu'une seule variable dépendante quantitative permettant ainsi de mesurer l'influence de cette variable sur une autre variable quantitative indépendante. La première sous-section présente les modèles construits avec les points de fonction étendus comme la seule variable indépendante (en considérant les recommandations statistiques qui sont de travailler d'abord avec le modèle le plus simple possible [NET83]). Ensuite, la deuxième sous-section présente les modèles de régression multiple construits à partir de divers critères qui pourraient affecter la productivité des activités de maintenance.

3.2.1 Modèle à une seule variable indépendante

Dans un premier temps, le modèle de régression linéaire simple a été construit avec le total de points de fonction étendus de chaque projet comme variable indépendante, et l'effort comme variable dépendante ou prédictive. Ceci permet de vérifier l'hypothèse déjà établie dans le domaine du génie logiciel sur l'absence de linéarité de la relation entre l'effort et la taille fonctionnelle [ISBSG98]. Selon les résultats obtenus, en tenant compte des 15 observations de l'échantillon, le modèle de la régression linéaire simple est représenté par la droite :

$$Y = A + B \cdot X \rightarrow Y = 91.1612 + 0.76814 \cdot X \quad (R^2 = 0.378 ; n = 15) \quad (\text{équation 3.3})$$

Tableau 3.10 – Régression Simple : Modèles à une seule variable indépendante

Simple Regression					
X-Variable: FFP					
Y-Variable: Efforts Total					
	N	A	B	R	R-Square
Y = A + B*X	15	91.16119016	0.768135438	0.614757384	0.377926642
Y = A * X^B	15	0.948681703	1.046212274	0.846452617	0.716482032
Y = A * e^(B*X)	15	45.95098263	0.006603543	0.622388428	0.387367355
Y = A + B*ln(X)	15	-241.2013933	96.7927853	0.664978001	0.442195742
Y = A + B/X	15	265.160595	-2570.562883	0.524080073	0.274659923
Y = 1 / (A + B*X)	15	0.057006895	-0.000202571	0.477501139	0.228007338
Y = X / (A + B*X)	15	1.614288644	-0.004863173	0.969351355	0.93964205

Le modèle fourni par l'équation 3.3 est représenté à la figure 3.1 par son diagramme de dispersion avec la ligne de régression correspondante. Ceci n'est évidemment pas le résultat d'une représentation linéaire fortement positive lorsqu'on analyse la qualité du modèle donné par son coefficient de détermination $R^2 = 0.3779$. Seulement 38% environ de la variabilité totale de l'effort dépensé dans les projets est expliquée par la variation dans leur nombre de points de fonction étendus.

Simple Regression

X-Variable: FFP
Y-Variable: Efforts Total

	N	A	B	R	R-Square
Y = A + B*X	15	91,16119016	0,768135438	0,614757384	0,377926642

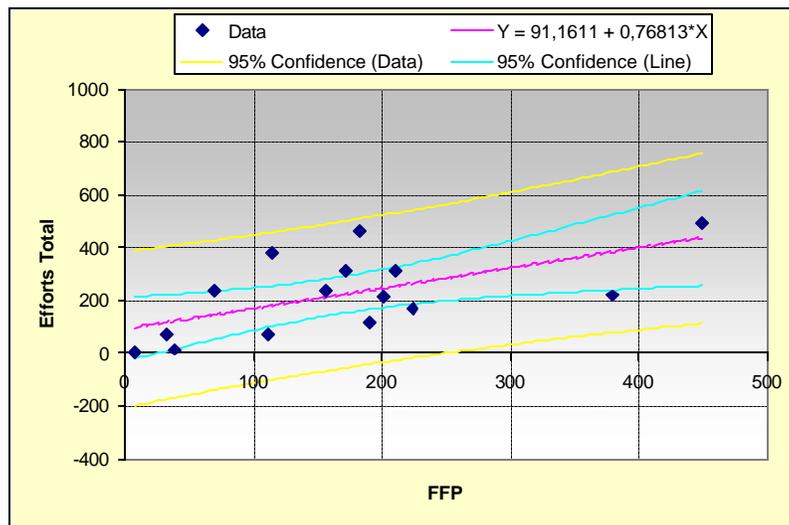


Figure 3.11 – Diagramme de dispersion avec la ligne de régression correspondante du modèle de la régression linéaire simple

D'autre part, il était indispensable de vérifier l'hypothèse standard faite à priori sur les termes du modèle à partir de l'analyse des résidus. Les résidus sont des estimations des termes d'erreur, l'hypothèse essentielle concernant ces termes est celle de leur distribution statistique, (termes indépendants de distribution normale de moyenne nulle et de variance constante) [WEI85].

L'analyse des résidus (standardized residuals) a permis de conclure qu'il n'existait pas une dispersion des résidus comprise entre -2 et $+2$ ni aléatoirement distribuée autour de zéro. Ceci devait être corroboré afin d'assurer qu'approximativement 95%

de l'échantillon des variables (distribution normale et d'écart type 1) serait compris dans ces valeurs.

On pourrait penser qu'il n'existe pas de relation linéaire complète entre les deux variables analysées : le modèle linéaire est donc insuffisant et doit être ajusté.

On a ensuite construit d'autres modèles de régression non linéaires; les résultats obtenus sont présentés dans le tableau 3.5. On peut ensuite comparer ces résultats pour déterminer le type de modèle non linéaire qui représente mieux cette relation non linéaire entre ces deux variables (FFP et l'effort). On constate que l'hypothèse de distribution Log-Normal pour ces variables [ISBSG98], laquelle est représentée par le modèle de Puissance : $Y = A * X^B$, donne des résultats satisfaisants.

À cet effet, des transformations logarithmiques ont été réalisées aux données afin d'atteindre la linéarité supposée dans cette analyse de régression simple. Ce modèle non linéaire est reconnu dans la littérature comme une fonction « linearizable » avec ses transformations appropriées.

Simple Regression

X-Variable: Log(FFP)
Y-Variable: Log(Effort)

	N	A	B	R	R-Square
Y = A + B*X	15	-0,022879475	1,046212274	0,846452617	0,716482032

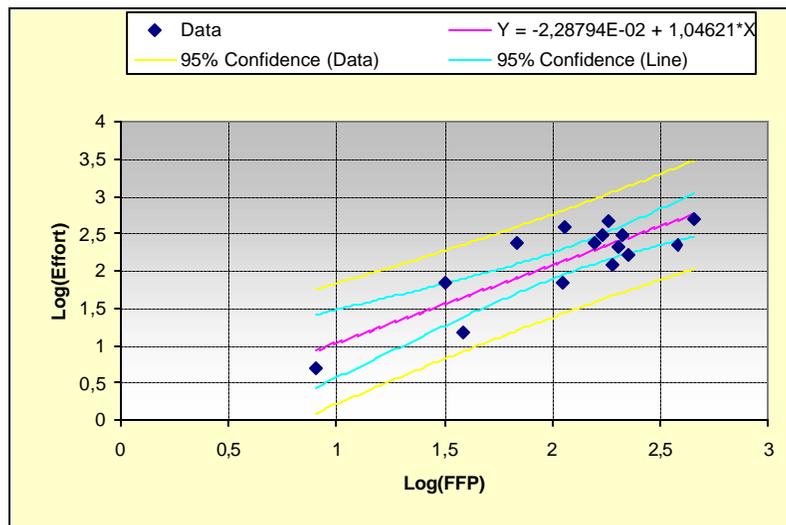


Figure 3.12 – Modèle linéaire avec des transformations logarithmiques: $Y = A + B \cdot X$

Pour cette régression, l'équation est la suivante:

$$Y = 1.04621X - 2.28794E-02 \text{ (équation 3.4)}$$

Donc

$$\text{Log}Y = 1.04621 \text{Log} X - 2.28794E-02$$

$R^2 = 0.72$; Standard erreur : 0.31; N = 15 observations

En appliquant l'antilogarithme à la formule de régression linéaire, on obtient la fonction $Y = A \cdot X^B$, laquelle permet d'estimer l'effort par rapport à la taille de la forme suivante :

$$\text{Effort} = \text{Taille}^{1.0462} * 0.9358 \quad (\text{équation 3.5})$$

Tableau 3.11 – Modèle de puissance : $Y = A * X^B$

Équation	R ²	ERM(%)	PRED (+/- 25%)		PRED (+/- 30%)		PRED (+/- 35%)	
			# Projets	%	# Projets	%	# Projets	%
3.5	0.72	26.66	4	26.6	4	26.6	5	33.3

En conclusion de cette première étape d'analyse, on peut suggérer que ce modèle de productivité simple représenté par l'équation (2), à une seule variable indépendante, et en tenant compte de l'échantillon des 15 projets, donne un modèle pour expliquer l'effort nécessaire pour produire un projet chez ABC Inc. Cependant ce modèle comporte une certaine marge d'imprécision au niveau de la prédiction de chacun des projets pris individuellement.

3.2.2 Identification de facteurs supplémentaires

On a donc décidé de segmenter de nouveau l'échantillon des observations afin d'identifier l'impact de facteurs pouvant affecter la productivité des activités de maintenance, ce qui pourraient améliorer le modèle de productivité à une seule variable. Les facteurs utilisés ont été pris selon la classification de Banker et al., de facteurs qui affectent la productivité des activités de maintenance selon des différents auteurs [MAY95]. Cependant, le niveau de chaque facteur dans chacun des projets a été établi par des jugements d'experts, donc de nature subjective.

Tableau 3.12 – Facteurs pouvant affecter la productivité des activités de maintenance

Facteurs	Type	Niveau
Concernant le personnel informatique	Expérience en informatique	Novice Intermédiaire Avancé
Concernant l'environnement technique	Complexité fonctionnelle et technique	Basse Moyenne Haute
Taille	Taille du logiciel	Fonctionnelle

3.2.3 Modèles à plusieurs variables indépendantes

Pour déterminer l'influence des divers facteurs, on a construit des modèles de régression à plusieurs variables indépendantes considérant les FFP et les niveaux mentionnés précédemment. L'objectif est de voir s'il y a une amélioration de la qualité de la relation entre la taille fonctionnelle et l'effort avec cette information, et par conséquent voir si ces facteurs ont un effet positif dans la relation de productivité.

3.2.3.1 Modèle pour déterminer l'influence de l'expérience des programmeurs en informatique

Dans la littérature, un des facteurs considérés comme ayant une influence forte dans la productivité des projets de développement et de maintenance est l'expérience des programmeurs. Il s'agit donc de construire un modèle de régression qui tient compte de cette variable qualitative indépendante «Expérience» ainsi que de la variable quantitative indépendante FFP. Des variables qualitatives peuvent être représentées par indicateurs ou variables appelées «dummy» qui décrivent leurs diverses catégories ou leurs valeurs. Les variables «dummy» prennent seulement deux valeurs, 0 et 1, et le nombre requis de ces variables dans le modèle est égal au nombre de catégories de la variable qualitative moins 1 [CHA91].

Dans notre étude, il existe 3 niveaux qui décrivent la variable qualitative « Expérience » : Novice, Intermédiaire et Avancée. Ainsi 2 variables « dummy » ont été définies de la forme suivante :

Expérience1 = 1 si l'expérience du programmeur est Avancée
= 0 autrement

Expérience2 = 1 si l'expérience du programmeur est Intermédiaire
= 0 autrement

La valeur de la catégorie « Novice » est représentée par des valeurs 0 dans les variables précédentes.

Tableau 3.13 – Catégories de la variable qualitative Expérience

Catégories/ Indicateur Variable	Expérience1	Expérience2
Avancée	1	0
Intermédiaire	0	1
Novice	0	0

Il est important d'observer que les deux dernières catégories (intermédiaire et novice) représentent 80 % des observations.

La régression linéaire multiple a été réalisée au moyen du logiciel *Winstat*, lequel donne pour chacune des variables dans l'équation leur coefficient estimé ainsi que l'intervalle de confiance pour leur coefficient. On peut assurer à 95% que la valeur réelle du paramètre en étude se trouve à l'intérieur des limites indiquées par l'intervalle de confiance lorsque l'échantillon est sélectionné à l'intérieur d'une population qui suit une distribution normale. P indique la signification du terme donné dans l'équation. Les coefficients de régression calculés sont des coefficients de régression partielle; ce terme signifie qu'ils mesurent l'effet de chaque variable une fois que les autres sont introduites dans la régression.

La méthode « Stepwise » offerte par l'outil Winstat, analyse chacun des coefficients pas à pas. Seulement ceux qui ont un apport significatif d'augmentation ou de diminution (coefficient positif ou négatif) dans la variation de la variable dépendante Y quand la variable indépendante augmente d'une unité sont introduits dans le modèle. Cependant, l'autre méthode « Direct », introduit toutes les variables dans le modèle laissant à l'utilisateur la possibilité de les interpréter selon les critères statistiques établis lors de l'analyse du paramètre p.

Tableau 3.14 – Régression linéaire multiple :LogFFP, Experience1, Experience2 et LogEffort (Méthode « Stepwise »)

Multiple Regression					
X-variable: Log(FFP), Expérience1, Expérience2					
Y-Variable: Log(Effort)					
Method: Stepwise					
Steps	P	R-Square	Corrected		
Log(FFP)(+)	6.9172E-05	0.716482032	0.694672958		
Summary					
	N	R	R-Square	Std.Error	
normal	15	0.846452617	0.716482032	0.309985735	
corrected		0.83347043	0.694672958		
Equation					
	Coefficient	95% Conf. (±)	Std.Error	T	P
Constant	-0.022879475	0.83636852	0.387133152	-0.059099757	0.953771484
Log(FFP)	1.046212274	0.394342051	0.18253064	5.731707699	6.9172E-05
Analysis of variance					
	Sum of Squares	Degrees of Freedom	Mean Square	F	P
Regression	3.156832117	1	3.156832117	32.85247315	6.9172E-05
Residue	1.249185026	13	0.096091156		
Total	4.406017142	14	0.31471551		

Selon les résultats obtenus après avoir appliqué les diverses méthodes (« Stepwise » et « Direct » dans la régression linéaire multiple, avec les variables indépendantes LogFFP, Experience1, Experience2 et la variable dépendante l'effort), la seule variable indépendante qui a une valeur statistique significative dans le modèle par rapport au paramètre « P value » est la variable LogFFP.

Tableau 3.15 – Régression linéaire multiple :LogFFP, Experience1, Experience2 et LogEffort (Méthode « Direct »)

Multiple Regression					
X-variable: Log(FFP), Expérience1, Expérience2					
Y-Variable: Log(Effort)					
Method: Direct					
Summary					
	N	R	R-Square	Std.Error	
normal	15	0.85487314	0.730808086	0.328365676	
corrected		0.810797206	0.657392109		
Equation					
	Coefficient	95% Conf. (±)	Std.Error	T	P
Constant	-0.053580252	1.151471296	0.523157287	-0.102417099	0.920268804
Log(FFP)	1.026514876	0.466987957	0.212170424	4.838161976	0.000520522
Expérience1	0.176098729	0.552580241	0.25105826	0.701425755	0.497608734
Expérience2	0.04615342	0.522259966	0.237282603	0.194508235	0.849323673
Analysis of variance					
	Sum of Squares	Degrees of Freedom	Mean Square	F	P
Regression	3.219952953	3	1.073317651	9.954346713	0.001818313
Residue	1.186064189	11	0.107824017		
Total	4.406017142	14	0.31471551		

Cette influence a déjà été corroborée lorsqu'on avait fait l'analyse de régression à une seule variable.

Toutefois, dans cette section, l'objectif est de constater l'influence de l'expérience du programmeur dans la relation entre la taille fonctionnelle et l'effort. Malgré la bonne qualité du modèle donnée par son coefficient de détermination, $R^2 = 0.73$, le résultat lié au facteur « expérience du programmeur » à trois niveaux ($p > 0.05$) n'explique pas l'effort nécessaire pour produire un projet lorsque la relation entre la taille fonctionnelle et l'effort est décrite par cette régression linéaire multiple.

Ce modèle ne présente aucune avantage par rapport au modèle à une seule variable.

3.2.3.2 Modèle pour déterminer l'influence de la complexité fonctionnelle

On a construit un deuxième modèle à plusieurs variables indépendantes qui considère la complexité fonctionnelle des projets selon les trois niveaux définis et les FFP comme variables indépendantes et l'effort comme variable dépendante.

De même que pour le facteur « expérience du programmeur », on a défini des variables « dummy » pour les diverses catégories du facteur « complexité fonctionnelle des projets »

Complexité1 = 1 si la complexité fonctionnelle du projet est Haute
= 0 autrement

Complexité2 = 1 si la complexité fonctionnelle du projet est Moyenne
= 0 autrement

La valeur de la catégorie « Basse » est représentée par des valeurs 0 dans les variables précédentes.

Tableau 3.16 – Catégories de la variable qualitative Complexité

Catégories/ Indicateur Variable	Complexité1	Complexité2
Haute	1	0
Moyenne	0	1
Basse	0	0

Résultats obtenus de la régression linéaire multiple selon l'outil Winstat :

Tableau 3.17 – Régression linéaire multiple : facteur « complexité fonctionnelle des projets »

Multiple Regression					
X-variable: Complex1, Complex 2, Log(FFP)					
Y-Variable: Log(Effort)					
Method: Direct					
Summary					
	N	R	R-Square	Std.Error	
normal	15	0.957171682	0.916177628	0.183234278	
corrected		0.945154475	0.893316981		
Equation					
	Coefficient	95% Conf. (±)	Std.Error	T	P
Constant	-0.09767274	0.50769048	0.230663131	-0.423443226	0.680132793
Complex 1	0.544294312	0.243804963	0.110769885	4.913739079	0.000461428
Complex 2	0.178027991	0.310490466	0.14106765	1.262004372	0.233049276
Log(FFP)	0.942691239	0.251803797	0.11440406	8.240015584	4.92624E-06
Analysis of variance					
	Sum of Squares	Degrees of Freedom	Mean Square	F	P
Regression	4.036694334	3	1.345564778	40.07662735	3.25886E-06
Residue	0.369322808	11	0.033574801		
Total	4.406017142	14	0.31471551		

Le modèle obtenu pour le facteur « complexité fonctionnelle des projets » présente un coefficient de détermination $R^2 = 0.76$, légèrement supérieur à celui du modèle du

facteur « expérience du programmeur ». Cependant, les coefficients des variables qualitatives liés à la « complexité fonctionnelle des projets » sont peu significatifs statistiquement, p value >0.05 , ce qui occasionne que le modèle ne donne pas de résultats concluants.

Les modèles de productivité construits précédemment avec les divers facteurs considérés n'améliorent pas la qualité de la relation entre la taille fonctionnelle et l'effort. L'apport dû à chaque facteur analysé n'est donc pas significatif car certaines des hypothèses faites sur les termes de ces modèles ne sont pas valables.

3.3 AJUSTEMENT DES CRITÈRES UTILISÉS POUR CLASSIFIER LES DONNÉES

Le processus d'introduction de variables indépendantes dans le modèle à plusieurs variables peut continuer. Cependant, le fait d'avoir un échantillon avec une taille trop faible pour chaque catégorie des variables qualitatives indépendantes ne nous permet pas de déceler des valeurs suspectes. On se trouve donc avec des modèles de régression dans lesquels le nombre de variables indépendantes est élevé par rapport à la taille de l'échantillon disponible.

Il est donc raisonnable d'examiner un ajustement dans les divers critères utilisés pour classifier les données [CHA91].

Cet ajustement peut être obtenu en se donnant une classification du facteur de « complexité fonctionnelle » à seulement deux niveaux à l'égard aussi du niveau d'expérience des programmeurs. Ceci nous permet de réduire une variable qualitative dans le modèle de régression à plusieurs variables et d'obtenir une interprétation plus facile de la validité des résultats. Pour en comprendre la raison, il suffit de regarder la décomposition à chaque palier et d'observer que la combinaison des deux critères nous amène à un effort requis qui est mesuré en nombre d'heures utilisées pour produire un projet avec ces caractéristiques.

La complexité fonctionnelle peut alors être représentée par une zone grise : Complexité Haute et une autre zone plus foncée : Complexité Basse. Toutefois, lorsque la complexité fonctionnelle est moyenne et l'expérience du programmeur est intermédiaire, cette relation peut probablement être comprise dans les deux catégories. Il faudra donc bien de regarder les données disponibles afin de construire un ajustement adéquat.

C1 : Complexité Basse; C2 : Complexité Moyenne; C3 : Complexité Haute.

E1 : Expérience Novice; E2 : Expérience Intermédiaire; E3 : Expérience Avancée

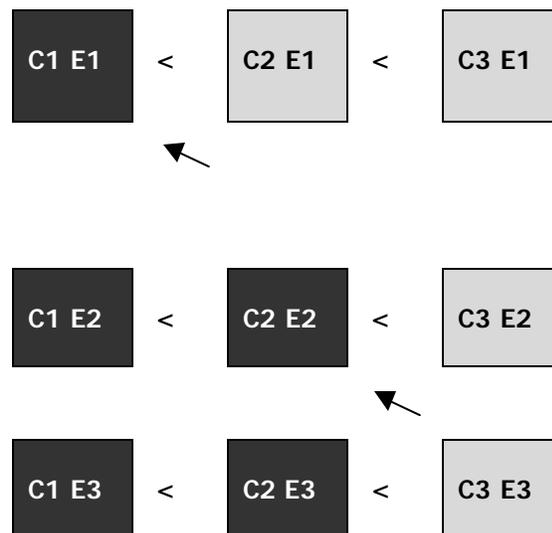


Figure 3.13 – Ajustement des critères utilisés pour classifier les données : Complexité Fonctionnelle et Niveau d'expérience en informatique

L'effort dépensé pour les caractéristiques C1E1 (complexité Basse et expérience Novice) est assigné comme T11 et considéré comme minimum. Il est inférieur à celui de T21 et de T31 correspondants aux applications qui ont des caractéristiques liées à la complexité moyenne et haute avec un niveau d'expérience du programmeur égal à novice : C2E1 et C3E1 respectivement.

Au fur et à mesure que le niveau de complexité augmente mais que le niveau d'expérience du programmeur reste égal, l'effort requis devient donc supérieur. Il existe une tendance vers la complexité Haute.

Par contre, au fur et à mesure que le niveau d'expérience du programmeur augmente, mais que le niveau de complexité fonctionnelle des applications reste égal, l'effort requis devient donc inférieur. Il existe une tendance vers la complexité Basse.

De plus, pour les applications qui ont un niveau équilibré d'expérience du programmeur par rapport à la complexité fonctionnelle de l'application (C1E1, C2E2, C3E3) la tendance dans le domaine en informatique est d'obtenir un effort minimum qui s'oriente vers la complexité Basse lors de la réalisation du projet. Toutefois, on remarque dans l'étude en cours que les projets ayant une complexité fonctionnelle Haute et un niveau d'expérience du programmeur égal à Avance ont tendances d'obtenir un effort très élevé. Ceci était occasionné par les caractéristiques inhérentes des projets qui comportent entre autres des programmes de protocoles de communications permettant de lier intelligemment les logiciels aux connaissances contenues sur un intranet, dans Internet ou chez un fournisseur de services applicatifs.

3.3.1 Modèle pour déterminer l'influence de l'ajustement de la complexité fonctionnelle.

On est revenu à la régression linéaire multiple, dans le cas de l'ajustement de la complexité fonctionnelle, pour corroborer l'influence qui selon la littérature peut exercer ce facteur qualitatif dans la productivité des projets de développement et de maintenance.

Une première possibilité consiste à construire un modèle à plusieurs variables indépendantes qui considère *l'effet d'interaction* entre les variables ajustées de la complexité fonctionnelle des projets (selon les deux niveaux définis) et les FFP. Cet effet d'interaction est représenté par le terme multiplicatif dans le modèle. Alors, les

points de fonction étendus FFP, l'ajustement de la complexité et le terme multiplicatif sont considérées comme variables indépendantes et l'effort comme variable dépendante.

La fonction qui représente le modèle est :

$$Y = B_0 + B_1X_1 + B_2X_2 + B_3X_1 \cdot X_2 \text{ (équation 3.6)}$$

Donc :

X1 : les points de fonction étendus FFP

X2 : l'ajustement de la complexité, AjustComple

B0, B1, B2, B3 : une constante et trois coefficients des variables indépendantes respectivement.

On a défini une variable « dummy » pour les deux catégories du facteur ajusté de la complexité fonctionnelle des projets.

AjustComple =1 si la complexité fonctionnelle du projet est Haute
=0 autrement

La valeur de la catégorie « Basse » est représentée par la valeur 0 dans la variable précédente.

Tableau 3.18 – Catégories de la variable qualitative Complexité après l'ajustement

Complexité	Valeur
Haute	1
Basse	0

Le modèle obtenu pour le facteur ajusté de la « complexité fonctionnelle des projets » a la forme suivante :

$$Y = B_0 + B_1X_1 + B_2X_2 + B_3X_1 \cdot X_2$$

Donc :

$\text{LogEffort} = -0.3592 + 1.0822 \text{ LogFFP} + 1.3611 \text{ AjustComple} - 0.4079 \text{ LogFFP} * \text{ AjustComple}$ (équation 3.7)

Pour la complexité Haute, on obtient alors :

$$\begin{aligned} Y &= B_0 + B_1 X_1 + B_2(1) + B_3 X_1 \\ &= (B_0 + B_2) + (B_1 + B_3) X_1 \\ \text{Log } Y &= 1.0019 + 0.6743 \text{ Log } X_1 \text{ (équation 3.8)} \end{aligned}$$

Donc :

$$\text{Effort} = \text{Taille}^{0.6743} * 10.0438 \text{ (équation 3.9)}$$

Pour la complexité Basse, on obtient alors :

$$\begin{aligned} Y &= B_0 + B_1 X_1 + B_2(0) + B_3(0) \\ &= B_0 + B_1 X_1 \\ \text{Log } Y &= -0.3592 + 1.0822 \text{ Log } X_1 \text{ (équation 3.10)} \end{aligned}$$

Donc :

$$\text{Effort} = \text{Taille}^{1.0822} * 0.4434 \text{ (équation 3.11)}$$

Ce modèle au complet présente un coefficient de détermination $R^2 = 0.95$, ce qui est supérieur à celui du modèle à une seule variable et un Standard-erreur égale à 0.14. Le coefficient de la variable qualitative lié à l'ajustement de la « complexité fonctionnelle des projets » est statistiquement significatif, le paramètre p value est < 0.05 et les autres aussi. La figure suivante révèle que l'effet de la complexité fonctionnelle du projet dépend considérablement de leur taille fonctionnelle.

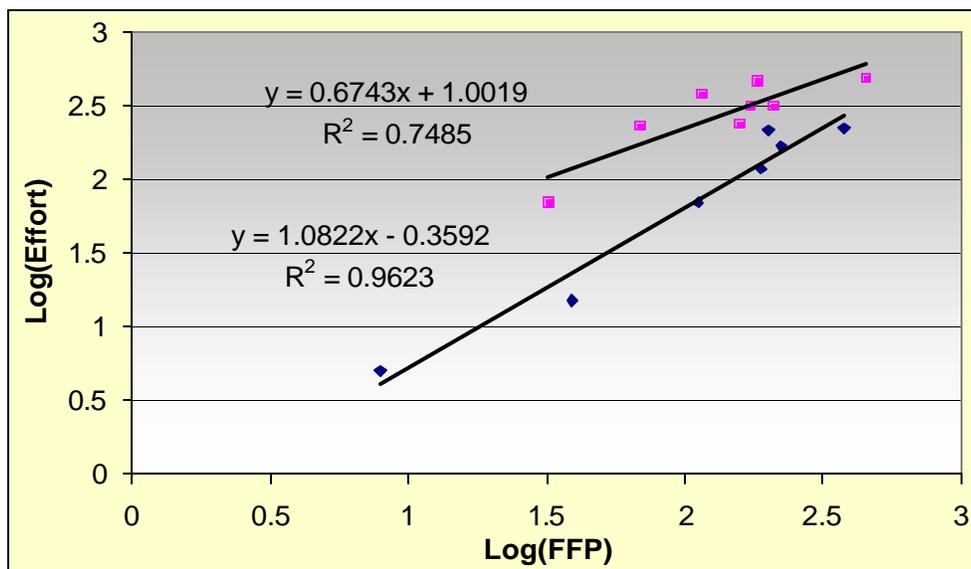


Figure 3.14 – Représentation graphique de l'équation 3.8 et 3.10

On a corroboré cet effet d'interaction lorsqu'on évaluait le coefficient du terme produit (B3) dans le modèle, lequel a un intervalle de confiance qui n'inclut pas la valeur zéro; donc il est significatif.

Toutefois, on doit être prudent lors de l'interprétation des applications qui ont une complexité fonctionnelle haute car l'étendue de l'échantillon analysé pour la variable dépendante (LogEffort) n'est pas assez grand par rapport à celui de la variable indépendante (LogFFP). Il existe donc moins de variations dans les observations de la variable dépendante à l'intérieur de l'étendue de la variable indépendante.

Toutes les remarques précédentes sur la validation du modèle sont valables et, en plus, on peut vérifier l'hypothèse standard faite à priori sur les termes du modèle (dont le modèle qui décrit les données est linéaire), au moyen de l'analyse des résidus. Les résultats obtenus de cette analyse nous révèlent que les variables indépendantes ont une dispersion des résidus comprise entre -2 et $+2$ et aléatoirement distribuées autour de zéro. Ceci nous permet d'assurer

qu'approximativement 95% de l'échantillon des variables est compris dans ces valeurs.

Ce modèle donne des résultats satisfaisants et démontre que le facteur ajusté de la complexité fonctionnelle a un effet sur la productivité.

Résultats obtenus de la régression linéaire multiple au moyen de l'outil Winstat :

Tableau 3.19 – Régression linéaire multiple : ajustement du facteur
« complexité fonctionnelle »

Multiple Regression					
X-variable: AjustCompl, Log(FFP), Log(FFP)*AjustCompl					
Y-Variable: Log(Effort)					
Method: Stepwise					
Steps	P	R-Square	Corrected		
Log(FFP)(+)	6.9172E-05	0.716482032	0.694672958		
AjustCompl(+)	6.50391E-05	0.928597148	0.916696672		
Log(FFP)*AjustCompl(+)	0.04874411	0.950628564	0.937163627		
Summary					
	N	R	R-Square	Std.Error	
normal	15	0.975001828	0.950628564	0.14062568	
corrected		0.968072119	0.937163627		
Equation					
	Coefficient	95% Conf. (±)	Std.Error	T	P
Constant	-0.359192591	0.454615211	0.20654901	-1.73901866	0.10990084
AjustCompl	1.361111724	0.865158846	0.39307463	3.46273103	0.00530754
Log(FFP)	1.082206116	0.218832477	0.09942393	10.8847646	3.1491E-07
Log(FFP)*AjustCompl	-0.407904144	0.40522785	0.18411045	-2.2155403	0.0487441
Analysis of variance					
	Sum of Squares	Degrees of Freedom	Mean Square	F	P
Regression	4.188485748	3	1.396161916	70.6002972	1.8006E-07
Residue	0.217531394	11	0.019775581		
Total	4.406017142	14	0.31471551		

On peut voir que ce modèle présente un coefficient de détermination plus significatif ($R^2 = 0.95$) que celui du modèle de régression simple, un intervalle (écart) de fiabilité de prédiction (PRED) égal à 60% (environ 75% des observations) et une erreur relative moyenne (ERM) de 24.09%.

Tableau 3.20 – Modèle linéaire multiple : ajustement du facteur « complexité fonctionnelle »

Équation	R^2	ERM(%)	PRED (+/- 25%)		PRED (+/- 30%)		PRED (+/- 35%)	
			# Projets	%	# Projets	%	# Projets	%
3.7	0.95	24.09	9	60	11	73.33	13	86.66

CHAPITRE IV

COMPARAISON DES RÉSULTATS AVEC D'AUTRES ÉTUDES EMPIRIQUES

Les résultats de ce projet sont comparables à ceux d'autres études empiriques réalisées très récemment dans l'industrie par [MAY95] et [SIL00] pour mesurer la taille fonctionnelle du logiciel au moyen de la méthode des points de fonction étendue proposée par le LRGL. Dans chacune des études, cette mesure a eu pour but de contrôler des coûts élevés des activités de maintenance et, donc, d'améliorer sa productivité.

Ces études empiriques ont utilisé le nombre total de points de fonction résultant ou équivalent, après la mesure des logiciels, pour faire des études de productivité (points de fonction par jour-personne) pour les activités de maintenance et plus particulièrement du type adaptatif au moyen des modèles statistiques. Plusieurs des logiciels mesurés étaient d'une petite envergure, laquelle pu être considérée adéquatement lors de la mesure grâce à la structure de granularité plus profonde et cohérente de la méthode FFP.

Les recherches réalisées par [MAY95] et [SIL00] ont été effectuées avec des données empiriques d'activités de maintenance adaptative dans des domaines différents respectivement : des logiciels d'informatique de gestion (MIS) et des logiciels en temps réel. Il existe par contre des procédures similaires lors de l'analyse des données ainsi que des résultats similaires qui peuvent être comparés dans certains cas avec notre étude réalisé dans le domaine intergiciel « middleware » (logiciel avec une architecture client-serveur fonctionnant dans un environnement multi-plateformes et permettant aux applications de texte d'utiliser des services fournis par un serveur local ou distant).

L'approche utilisée pour mesurer les divers logiciels et pour construire des modèles de productivité a été un peu semblable dans les trois études.

Premièrement, les principaux mesureurs étaient des étudiants chercheurs de niveau gradués en génie logiciel et mandataires des projets du Laboratoire de recherche en gestion du logiciel (LRGL), qui se familiarisaient avec la méthode de mesure FFP version 1.0 et version préliminaire 2.0 respectivement. Ces étudiants ont également reçu un support du domaine des applications et des systèmes mesurés de la part des experts des diverses entreprises.

Quant à l'ensemble de données utilisé dans les trois cas, on remarque que l'étude qui comportait plus de livrables de maintenance adaptative (504 livrables au cours de la période 1990-1994) était réalisée par [MAY95]. Ces activités de maintenance représentaient des additions ou des modifications fonctionnelles effectuées sur 23 applications du domaine de l'informatique de gestion MIS. Cependant, malgré qu'il y ait une banque de données d'envergure, l'information disponible n'a pas été pertinente que dans certains cas. La taille des différentes applications a été exprimée en *points de fonction équivalents*. Ce qui veut dire que l'entreprise a suivi une procédure de mesure des points de fonctions différente de celle utilisés dans l'étude réalisée par [SIL00] et dans l'étude en cours. La procédure utilisée par [MAY95] était la prise du nombre de lignes de code de chaque application (en excluant les lignes de commentaires) pour ensuite les traduire en points de fonction selon des équivalences par langage de programmation établies d'après les études réalisées par Jones (1986) [MAY95]. Cependant, la taille de l'application utilisée dans le contexte de maintenance était celle de la taille fonctionnelle de la modification (taille du changement fait sur une application).

Ainsi, avec cette information, nous pourrions probablement déduire que les résultats de l'étude réalisée par [MAY95] peuvent être comparables à ceux de notre étude seulement à un certain niveau. On pourrait peut-être l'expliquer par le fait que l'étude réalisée par [MAY95] a utilisé la taille fonctionnelle de la modification mesurée par

PFE sur des applications et une taille approximative de certaines applications. Ceci pouvait impliquer des résultats statistiquement faibles avec un pourcentage d'erreur un peu élevé lors de la détermination d'un modèle de productivité. Cependant, dans notre étude, nous avons utilisé la taille fonctionnelle de chaque application pour construire des modèles de productivité.

Dans les trois études, nous avons aussi remarqué dans certains cas des résultats opposés de même que nous avons construit des modèles de coût unitaire moyen, des modèles de régression simple et multiple en considérant certains facteurs suggérés par la littérature.

Dans l'étude réalisée par [MAY95], plusieurs modèles de productivité ont été construits avec les points de fonction, l'effort et d'autres facteurs ayant une influence sur les activités de maintenance. Aucun de ces résultats n'a cependant pu démontrer statistiquement que la qualité de la relation entre la taille fonctionnelle et l'effort était significative lorsque cette relation est utilisée comme la base d'un modèle de productivité, ni que ces facteurs étudiés ont un effet sur la productivité des activités de maintenance adaptative mesurées.

Contrairement, dans l'étude réalisée par [SIL00], des modèles de régression multiple ont démontré que le facteur de complexité fonctionnelle de l'application a une influence sur la productivité des activités de maintenance adaptative mesurées et qu'il peut être utilisé comme facteur pour faire des prédictions. Ces résultats sont plus comparables à ceux obtenus dans ce projet.

Toutefois, dans l'analyse des activités de maintenance chez ABC Inc., nous avons aussi remarqué l'impact sur notre ensemble de données de la combinaison de ce facteur de complexité fonctionnelle des projets avec un autre facteur: le niveau d'expertise du programmeur. On a pu ainsi confirmer des comportements similaires dans les deux études lors de la considération du facteur de complexité fonctionnelle sur la productivité des activités de maintenance. Il faudrait cependant apporter des

améliorations dans la procédure d'estimation de ces facteurs afin d'éviter qu'ils soient des valeurs subjectives et n'apportent pas des indicateurs de performance ou de mesures de productivité pour la planification des futurs projets de maintenance.

Un élément important qui n'a pas été mentionné auparavant est celui des modèles construits à partir du coût unitaire moyen. Les résultats de ces modèles dans les trois études sont très semblables indiquant qu'ils ne rencontrent pas les critères statistiques de bons modèles de productivité.

Les résultats de notre étude sont importants pour la communauté scientifique qui a souligné le manque de techniques de mesure et de données empiriques pour identifier ces caractéristiques particulières des activités de maintenance qui influencent la qualité de la relation entre la taille fonctionnelle et l'effort, relation qui peut être utilisée comme modèle de productivité.

CHAPITRE V

SOMMAIRE DES RÉSULTATS

Au début de cette étude, nous nous sommes fixé comme objectif principal de construire des modèles de productivité pour estimer la maintenance au sein de la compagnie ABC Inc. Pour cela nous avons appliqué la méthode de points de fonction étendus à un échantillon des applications CENTREX afin de mesurer leur taille fonctionnelle. Par la suite, nous avons fait des analyses statistiques lesquelles nous permettaient d'établir des modèles de productivité. Ces résultats sont une base solide pour estimer et valider l'effort à fournir au processus d'entretien dans la compagnie ABC Inc., ce qui nous permet de corroborer et d'offrir une méthode uniforme pour quantifier la taille fonctionnelle d'un intergiciel. Cela minimisera les risques et les incertitudes dans l'estimation de la productivité liés aux projets de maintenance.

En même temps, l'objectif relié à l'efficacité de la méthode des points de fonction étendus pour mesurer la relation positive entre l'effort et la taille fonctionnelle dans divers contextes de logiciels, et en particulier dans le domaine d'une architecture multi-plateforme (intergiciel) était illustré grâce à l'obtention de la quantification de la taille fonctionnelle d'un échantillon du logiciel CENTREX avec un niveau de fiabilité acceptable.

D'abord, pour répondre à ces objectifs, l'ensemble de données sélectionné dans l'étude comporte 15 projets, lesquels ont été développés et intégrés dans la Préversion 1 (Preview 1) de CENTREX entre mai 1998 et juillet 1999. Il a été difficile tout au début de déterminer ce qui devait être mesuré dans l'intergiciel CENTREX. Ceci est dû aux caractéristiques inhérentes à cette catégorie de logiciel, à la carence d'un support plus complet de la méthode par un expert et à la faiblesse de la

documentation des applications à mesurer avec la méthode de mesure FFP. La documentation disponible a parfois été seulement les interfaces reliées aux fonctionnalités des applications à mesurer, ce qui pouvait occasionner de mauvaises interprétations à certain moment par les mesureurs.

Les méthodes d'analyse statistiques utilisées et les résultats obtenus tout au long de notre étude, nous ont permis d'atteindre nos objectifs. L'utilité de la méthode des points de fonction étendus au moyen des modèles construits pour mesurer *la relation positive entre l'effort et la taille fonctionnelle* attribuée par la mesure (*hypothèse nulle formulée*) ainsi que l'analyse de l'intervention d'autres facteurs de productivité dans le domaine d'une architecture multi-plateforme, étaient vérifiées et validées.

La démarche de l'analyse de la qualité de la relation entre la taille fonctionnelle FFP et d'autres facteurs comme variables indépendantes et l'effort comme variable dépendante s'est faite en comparant les divers résultats des modèles établis à partir du coût unitaire et de la régression simple et multiple.

Concernant l'analyse de la qualité de la relation entre la taille fonctionnelle FFP et l'effort dans les modèles avec le coût unitaire moyen, cela nous a indiqué que l'écart entre l'effort estimé et l'effort réel est très élevé et que la fiabilité de ces modèles est très basse. Pour l'échantillon analysé, le rapport entre l'effort et les FFP n'est pas adéquat pour être utilisé comme modèle de productivité et d'estimation.

Quant aux modèles de régression linéaire, les résultats ont montré que, pour l'échantillon analysé, il existe un modèle simple qui peut décrire la relation positive entre la taille fonctionnelle et l'effort, à partir d'une fonction non linéaire : « Puissance ». L'accroissement de la taille fonctionnelle des projets et l'accroissement de l'effort requis pour les produire est bien représenté par ce modèle. Le niveau de qualité du modèle a été assez acceptable avec un coefficient de détermination $R^2 = 0.72$. Cependant, l'exactitude du modèle est un peu éloignée

des critères proposés par Conte et al. (1986) : « l'erreur relative moyenne (ERM) égale ou plus petite que 25% et la fiabilité de la prédiction (+/- 25% par rapport aux données pour au moins 75% des observations)». Le modèle simple présente un ERM de 60% et seulement 27% des efforts calculés sont à l'intérieur d'un intervalle de +/- 25% par rapport aux efforts réels.

En plus, plusieurs modèles de productivité ont été construits avec la variable LogFFP et les divers facteurs qui pourraient améliorer la qualité de la relation entre l'effort et la taille fonctionnelle. À cet effet, l'échantillon de données a été divisé en sous-ensembles ayant des caractéristiques similaires tels que le niveau d'expertise du programmeur et la complexité fonctionnelle des projets. Toutefois, aucun résultat significatif n'a été trouvé avec ces conditions, ce qui nous a conduit à construire un ajustement entre ces deux facteurs.

Pour compléter les analyses, le modèle de régression multiple a été construit avec l'ajustement de la complexité des projets, le Log de la taille fonctionnelle et de l'effort, qui donnent des résultats satisfaisants pour notre ensemble des observations. Ce modèle en a été un de régression multiplicative qui déterminait l'existence d'une relation d'interaction entre les variables indépendantes de notre étude, dont l'effet de la complexité fonctionnelle du projet dépend suffisamment considérablement de leur taille fonctionnelle. Le niveau de qualité du modèle a été assez acceptable avec un coefficient de détermination $R^2 = 0.95$. De plus, ce modèle est plus proche des limites proposées dans la littérature, présentant une erreur relative moyenne (ERM) de 24.09% et un intervalle (écart) de fiabilité de prédiction (PRED) égal à 60% (environ 75% des observations).

Cette expérience suggère d'être prudent lors de l'interprétation des applications ayant une complexité fonctionnelle haute car l'étendue de l'échantillon analysé pour la variable dépendante (LogEffort) n'est pas assez grand par rapport à celui de la variable indépendante (LogFFP). Il existe donc moins de variations dans les observations de la variable dépendante à l'intérieur de l'étendue de la variable indépendante.

Les résultats des analyses effectuées sur les divers facteurs (qui pourraient améliorer la qualité de la relation entre l'effort et la taille fonctionnelle), nous ont permis de constater que la relation entre la taille fonctionnelle et l'effort peut être influencée positivement par certains facteurs comme celui de la complexité fonctionnelle du projet, tel que mentionné dans la littérature. Toutefois, la mesure de ce facteur doit se faire également en appliquant des méthodes plus uniformes et reconnues dans le domaine du génie logiciel, ce qui n'était pas notre cas qui constituait une première approche uniquement.

Un autre aspect à considérer lors de l'interprétation des résultats est la carence de documentation de la spécification fonctionnelle des applications à mesurer qui rendait l'assignation de niveaux de complexité fonctionnelle plus subjective et les sessions de mesures de la taille fonctionnelle plus lourdes et plus dépendantes de la connaissance du domaine d'affaire de la part des programmeurs. Il sera nécessaire de considérer la possibilité d'intégrer au sein de la compagnie des méthodes plus uniformes afin de mesurer adéquatement les applications et plus particulièrement la complexité fonctionnelle des applications pour éviter la dépendance de l'expertise des analystes-programmeurs.

Enfin, la méthode de points de fonction étendus nous a permis de trouver des modèles d'estimation de productivité pour la maintenance chez ABC Inc., soit la forme de régression simple ou multiple. Ces modèles sont valides seulement pour le contexte analysé, mais ils pourraient toutefois être utilisés et comparés avec les résultats d'autres recherches d'analyse de productivité dans la maintenance au LRGL, augmentant ainsi son ensemble de données et sa maturité d'expertise dans ce domaine.

RÉFÉRENCES

- [ABR93] Abran, Alain et H. Nguyenkim. 1993. «Measurement of the Maintenance Process from a Demand based Perspective». *Software maintenance: research and Practice*, Vol. 5, p. 63-90
- [ABR97] Abran, Alain, Marcela Maya, Jean-Marc Desharnais et Denis St-Pierre. 1997. «Adapting Function Points to Real-time Software». *American Programmer*, Vol. 10, no 11, p. 32-43.
- [AST96] D'Astous, Patrick. 1996. «Les aspects de l'échange d'Information dans un processus de génie logiciel». Département de Génie Électrique et de Génie Informatique. École Polytechnique de Montréal, Novembre.
- [BOE81] Boehm, Barry. 1981. *Software Engineering Economics*, Prentice-Hall, New York, 1981
- [BOU99] Bourque, P., R. Dupuis, A. Abran, J. W. Moore, and L. L. Tripp. 1999. «The Guide to the Software Engineering Body of Knowledge». *IEEE Software*, Vol. 16, p. 35-44, November/December 1999.
- [DES97] Desharnais, Jean-Marc. 1998. «Mesure de la taille fonctionnelle des logiciels temps réel». Département d'informatique, Montréal.
- [CHA91] Samprit Chatterjee, Bertran Price. 1991. *Regression Analysis by Example*. Second Edition. Wiley series in probability and mathematical statistics.
- [HIL99] Hilburn, Thomas B., Iraj Hirmanpour, Soheil Khajenoori, Richard Turner, Abir Qasem. 1999. «A Software Engineering Body of Knowledge Version 1.0.»_Technical Report, CMU/SEI-99-TR-004, ESC-TR-99-004, April.
- [ISBSG98] International Software Benchmarking Standards Group Ltd, Release 5. March 1998.
- [MAY95] Maya M. 1995. «La technique étendue des points de fonction dans la construction des modèles de productivité en maintenance adaptative». UQAM – Rapport d'activité de synthèse. décembre 1995
- [MEL97] Robert Meli. 1997. «Early and Extended Function Point : A method for Function Points Estimation». *IFPUG Fall Conference*, Scottsdale, Arizona USA, September15-19.

- [MIK99] Maj. Mike N., U.S Army, Clark J., Spurlock M. 1999. «Curing the Software Requirements and Cost Estimating Blues». *Software Management & Cost Estimating*, November-Décember.
- [MOR98] Morris, Pam et Jean-Marc Desharnais. 1998. «Measuring all the Software not just what the Business Uses». *IFPUG Fall Conference*, Orlando, Florida, September 21-25.
- [MOT99] Motulsky, Harvey. 1999. *Analyzing Data with GraphPad Prism*. Graph Pad Software, Inc. March.
- [NET83] Neter, Jhon, William Wasserman, Michael H. Kutnen. 1983. *Applied linear regression models*.
- [ROC99] Roche, Stéfanie. 1999. «Profil Français : L'accessibilité aux services linguistiques», *ABC Inc.*, août.
- [OLI99] Oigny, Serge, Alain Abran et Denis St-Pierre. 1999. «Improving Software functional size Measurement». UQAM Software Engineering Management Research Laboratory.
- [SEL00] Sellami, Asma. 2000. «Analyse comparative des modèles de maintenance du logiciel entre SWEBOK, ISO/IEC 14764 et la littérature». UQAM – Mémoire. Juin.
- [SIL00] Silva, Ilionar. 2000. «Mesurer la taille fonctionnelle et bâtir un modèle de productivité pour les activités de maintenance adaptative en industrie». UQAM – Rapport de projet. Juillet.
- [WEI85] Weisberg, S. 1985. *Applied Linear Regression*, John Wiley & Sons, New York, NY.

Sites Web :

http://www.CENTREX.org/fr_index1.html, L'intégration parfaite des outils linguistiques aux applications de bureau. ABC Inc., 2000

<http://www.lrgl.uqam.ca/cim>

Laboratoire de Recherche en Gestion des Logiciels (LRGL) et Laboratoire de Métriques Appliquées en Gestion du Logiciel (LMAGL), La mesure du logiciel : Bref Historique de la Mesure de logiciel.

APPENDICE A – SOMMAIRE DES DONNÉES

A1. DONNÉES MESURÉES ET COMPLÉMENTAIRES

On présente les données de mesure disponibles pour l'étude en cours provenant de l'entreprise ABC Inc. La taille fonctionnelle des applications a été mesurée par les points de fonction étendus (COSMIC-FFP version 2.0 selon la norme ISO/IEC 14143-3). L'information des données complémentaires a été recueillie afin de corroborer leur influence sur la qualité de la relation entre l'effort et les points de fonction. L'ensemble de données utilisé dans l'étude comporte 15 projets, lesquels ont été développés et intégrés dans la version Preview 1 CENTREX entre mai 1998 et juillet 1999.

Applications	FFP	Effort Total (hrs)	Labour rate	Ligne Code	Ressource	Complexité	Expérience
A	111	70	0.631	770	AP	Low	Novice
B	379	219	0.578	1300	AP	Low	Avancé
C	190	119	0.626	927.5	AP	Medium	Intermédiaire
D	157	238	1.516	820	AP	Medium	Novice
E	202	216	1.069	600	AP	Low	Intermédiaire
F	224	167	0.746	1000	AP	Medium	Intermédiaire
G	183	460	2.514	1250	DH	High	Intermédiaire
H	8	5	0.625	150	DH	Low	Intermédiaire
I	115	383	3.330	520	DH	High	Intermédiaire
J	32	70.5	2.203	150	DH	High	Intermédiaire
K	450	495.25	1.101	1350	DH	High	Novice
L	39	15	0.385	260	DH	Low	Intermédiaire
M	172	312	1.814	1425	FM	High	Avancé
N	210	314.5	1.498	220	FM	High	Avancé
O	69	235	3.406	100	FM	High	Avancé

A2. VALIDATION ET TRANSFORMATION DES DONNÉES

Une transformation logarithmique a été effectuée aux variables FFP et Effort afin d'obtenir la linéarité dans le modèle de productivité à établir ainsi que de permettre une meilleure analyse et validation des résultats. Les catégories des variables qualitatives « Expérience » et « Complexité » ont été identifiées en suivant les critères établis dans les sections 3.2.3.1 et 3.2.3.2 respectivement.

Applications	Log(FFP)	Log(Effort)	Complexité	Complexité 1	Complexité 2	Expérience	Exp1	Exp2
A	2.045323	1.84509	Low	0	0	Novice	0	0
B	2.578639	2.34044	Low	0	0	Avancé	1	0
C	2.278754	2.07554	Medium	0	1	Intermédiaire	0	1
D	2.1959	2.37657	Medium	0	1	Novice	0	0
E	2.305351	2.33445	Low	0	0	Intermédiaire	0	1
F	2.350248	2.22271	Medium	0	1	Intermédiaire	0	1
G	2.262451	2.66275	High	1	0	Intermédiaire	0	1
H	0.90309	0.69897	Low	0	0	Intermédiaire	0	1
I	2.060698	2.58319	High	1	0	Intermédiaire	0	1
J	1.50515	1.84818	High	1	0	Intermédiaire	0	1
K	2.653213	2.69482	High	1	0	Novice	0	0
L	1.591065	1.17609	Low	0	0	Intermédiaire	0	1
M	2.235528	2.49415	High	1	0	Avancé	1	0
N	2.322219	2.49762	High	1	0	Avancé	1	0
O	1.838849	2.37106	High	1	0	Avancé	1	0

A3. AJUSTEMENT DES CRITÈRES UTILISÉS POUR CLASSIFIER LES DONNÉES À L'ÉGARD DE LA COMPLEXITÉ FONCTIONNELLE ET L'EXPÉRIENCE DES PROGRAMMEURS.

Cet ajustement donne une classification du facteur de « complexité fonctionnelle » à seulement deux niveaux à l'égard aussi du niveau d'expérience des programmeurs. Ceci nous permet de réduire une variable qualitative dans le modèle de régression à plusieurs variables et d'obtenir une interprétation plus facile de la validité des résultats.

Applications	Complexité	Expérience	Complexité/ Expérience	AjustCompl
A	Low	Novice	C1E1	Low
B	Low	Avancé	C1E3	Low
C	Medium	Intermédiaire	C2E2	Low
D	Medium	Novice	C2E1	High
E	Low	Intermédiaire	C1E2	Low
F	Medium	Intermédiaire	C2E2	Low
G	High	Intermédiaire	C3E2	High
H	Low	Intermédiaire	C1E2	Low
I	High	Intermédiaire	C3E2	High
J	High	Intermédiaire	C3E2	High
K	High	Novice	C3E1	High
L	Low	Intermédiaire	C1E2	Low
M	High	Avancé	C3E3	High
N	High	Avancé	C3E3	High
O	High	Avancé	C3E3	High

A4. MODÈLE DU COÛT UNITAIRE MOYEN POUR L'ENSEMBLE DE PROJETS.

Applications	Effort Estimé	ERM(%)
A	163.059	1.3294143
B	556.751	1.542242
C	279.11	1.3454622
D	230.633	0.0309538
E	296.738	0.373787
F	329.056	0.9703952
G	268.827	0.4155935
H	11.752	1.3504
I	168.935	0.5589164
J	47.008	0.3332199
K	661.05	0.3347804
L	57.291	2.8194
M	252.668	0.1901667
N	308.49	0.0191097
O	101.361	0.5686766

Équation	ERM(%)	PRED(+/- 25%)	
		# projets à l'intérieur	%
Effort = 1.47 * FFP	81.28	3	20