	Sujet d'étude MGL 804	DOCUMENT NO.	DATE 2011-03-03	VERSION 1.0
	TITRE Modèle de mesure du processus maintenance de logiciel		PAGE 1	PAGES 22


Sujet d'Étude individuel

Cours MGL 804

'MODÈLE DE MESURE DE LA MAINTENANCE DE LOGICIEL'


Réalisé par : GHANIA HACHICHI

2011-03-17

	Sujet d'étude MGL 804	DOCUMENT NO.	DATE 2011-03-03	VERSION 1.0
	TITRE Modèle de mesure du processus maintenance de logiciel		PAGE 2	PAGES 22

SOMMAIRE

	Résumé	3
1.	Introduction	3-4
2.	La mesure en génie logiciel	5
3.	Mesure de la maintenance de logiciel	6
3.1	Concept de la mesure de la maintenance de logiciel (ISO/IEC 14764)	6
3.2	Problématique liée à la maintenance	6
3.3	La mesure du processus maintenance	6-7
3.3.1	processus de la maintenance	7-8
3.3.2	Les activités de la maintenance	8-9
3.4	Modèles de mesure de la maintenance	9
3.4.1	survole de littérature	9-10
3.4.2	Les travaux sur les modèles de mesure de la maintenance dans la Littérature	10-11
3.4.3	Techniques de modélisation des estimations	12
3.4.4	Pourquoi estimer la maintenance?	12-14
3.4.5	Estimation de l'effort de la maintenance	14-16
3.4.6	Estimation de la taille fonctionnelle	16-17
4.	Métriques proposés par l'ISBSG	17-19
	Conclusion	20

	Sujet d'étude MGL 804	DOCUMENT NO.	DATE 2011-03-03	VERSION 1.0
	TITRE Modèle de mesure du processus maintenance de logiciel		PAGE 3	PAGES 22

Résumé

Les coûts de maintenance de logiciels dépassent généralement les coûts de développement de logiciels et Ferens [7] indique que l'attention accordée à ces frais n'a pas été proportionnelle à leur importance: par exemple, seule une poignée de modèles d'estimation ont été proposés pour la maintenance des logiciels alors qu'un grand nombre ont été proposés pour les projets de développement [8]. Ces modèles d'estimation ont pour objectif de diminuer les temps d'exécution et le coût de maintenance

Ce travail est effectué dans le but de présenter un modèle de mesure des processus pour les mainteneurs en se basant sur la revue de la littérature spécialisée en mesure de la maintenance.

1. Introduction


Le développement des logiciels comprend les phases de l'ingénierie des exigences, architecture, conception, implémentation, tests, déploiement des logiciels, et maintenance. En génie logiciel, la maintenance du logiciel désigne les modifications apportées à un logiciel, après sa mise en œuvre, pour en corriger les fautes, en améliorer l'efficacité ou autres caractéristiques, ou encore adapter celui-ci à un environnement modifié (ISO/IEC 14764) [1]. La maintenance est la dernière étape du cycle de vie du logiciel. Après que le produit ait été livré, la phase de maintenance met le logiciel à jour en fonction des changements de l'environnement et des besoins des utilisateurs changeants.

Les phases précédentes, doivent être faites de telle manière que le produit soit facilement maintenable. En conception par exemple, il est recommandé de prévoir que la structure à construire doit être modifiable. De même la phase de mise en œuvre devrait créer un code qui peut être facilement lu, compris, et changé. La maintenance ne peut s'effectuer de manière efficace si les phases antérieures ne sont faites correctement.

La maintenance et l'évolution des logiciels furent abordées pour la première fois par le Dr. Lehman, en 1969. S'étendant sur une vingtaine d'années, ses recherches aboutirent à la formulation de huit règles de l'évolution d'un logiciel (1997). Elles ont mis en lumière le fait que la maintenance est un processus évolutif et que les logiciels évoluent avec le temps. En évoluant, ils deviennent plus complexes à moins qu'une action spécifique soit engagée pour en réduire la complexité.[1]

Selon le modèle de qualité logicielle de la norme ISO/IEC 9126, l'un des facteurs de qualité d'un produit logiciel est la maintenabilité.

La maintenabilité est définie dans la norme comme étant l'ensemble d'attributs portant sur l'effort nécessaire pour faire des modifications données. Autrement dit c'est la facilité avec laquelle un logiciel peut être maintenu. Les critères de ce facteur de qualité sont :

	Sujet d'étude MGL 804	DOCUMENT NO.	DATE 2011-03-03	VERSION 1.0
	TITRE Modèle de mesure du processus maintenance de logiciel		PAGE 4	PAGES 22


- ✓ Facilité d'analyse : effort nécessaire pour diagnostiquer les déficiences et causes de défaillance ou pour identifier les parties à modifier
- ✓ Facilité de modification : effort nécessaire pour modifier, remédier aux défauts ou changer d'environnement
- ✓ Stabilité : risque des effets inattendus des modifications
- ✓ Facilité de test : effort nécessaire pour valider le logiciel modifié

La maintenabilité est donc un des facteurs qui influencent les coûts de maintenance. Il est essentiel pour les gestionnaires de systèmes de bien contrôler cet aspect de la qualité des logiciels. Pour faire une gestion rationnelle de la maintenabilité, il est primordial de disposer d'une mesure quantitative. Une telle mesure permettrait de définir des critères quantitatifs de maintenabilité pour les logiciels à développer et d'assurer un suivi durant la maintenance du logiciel. De plus, cette mesure pourrait servir à identifier les systèmes nécessitant une réingénierie afin d'améliorer la maintenabilité et de réduire les coûts de maintenance. [2]

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 du processus de développement des produits et des logiciels a une incidence sur le coût, l'échéancier, la qualité et la fonctionnalité des livrables. 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 logiciels, permettant ainsi aux gestionnaires d'atteindre leurs propres objectifs techniques et d'affaires.

Les mesures constituent une partie fondamentale dans plusieurs domaines scientifiques en général et en génie logiciel en particulier. Elles sont utilisées pour contrôler, gérer et améliorer la qualité des besoins exigés pour la création du système logiciel (ressources), les méthodes suivies (processus) et le produit résultant de cet ensemble de processus ainsi elles permettront d'améliorer la précision du coût, du calendrier et la satisfaction du client.

Mais s'il semble correct de dire que les mesures facilitent la maîtrise, cela veut sans doute dire qu'elles sont, non seulement inutiles mais elles peuvent créer de fausses assurances qui conduisent facilement à des catastrophes quand elles sont employées sans un cadre théorique.[3]

	Sujet d'étude MGL 804	DOCUMENT NO.	DATE 2011-03-03	VERSION 1.0
	TITRE Modèle de mesure du processus maintenance de logiciel		PAGE 5	PAGES 22

2. La mesure en génie logiciel


Jusqu'à la Renaissance européenne, les grandeurs étaient évaluées en comparaison avec des références humaines, comme le pied, le pouce ou la ligne (1/12e de pouce) pour les longueurs (souvent les organes des rois et empereurs), le journal pour la surface (champ gérable par une personne s'en occupant quotidiennement). Des mesures de même nature et de valeurs voisines avaient des appellations différentes selon les provinces, voire les villes ou les villages d'une même région. A l'inverse, le contenu physique de mesures de même nom différait en général selon les lieux et aussi selon la corporation intéressée ou l'objet mesuré. La nécessité de mesurer s'est alors fait ressentir autour de deux besoins : les déplacements afin de repérer sa position dans l'espace et le commerce afin de garantir la loyauté des échanges. Il a donc fallu inventer les différentes unités (mètre et kilogramme notamment) utiles au commerce [4]. Un système de référence basé sur des objets ayant la même valeur pour tous les peuples et tous les temps _Universel_ accessible à tous et reconnu par tous [5]. La mesure n'est pas limitée à un domaine particulier. On mesure souvent sans se rendre compte de l'acte lui-même lorsque l'on s'intéresse à une valeur qualitative plutôt qu'uniquement à une représentation numérique.

Vu son jeune âge, le génie logiciel n'est pas un domaine complètement mature. De ce fait découlent plusieurs problèmes. Puisque la mesure joue habituellement un rôle important dans le génie [Vincenti, 1990 ; Kirby et Al., 1990], son utilisation, en génie logiciel semble essentielle, pourtant sa concrétisation n'est pas toujours simple. Oman et Pfleeger(1994) ont identifiés six raisons pour lesquelles on applique des mesures en génie logiciel :

- ✓ Mesurer pour mieux comprendre les activités de développement et de maintenance de logiciel et évaluer ainsi la situation courante et prévoir les comportements futurs.
- ✓ Mesurer pour expérimenter et tester les hypothèses et en bâtir de nouvelles
- ✓ Mesurer pour contrôler le projet, l'évaluer et prédire ce qui se produira dans le futur et de rectifier les erreurs si nécessaire.
- ✓ Mesurer pour améliorer la qualité de produit. La mesure fournit des réponses sur l'efficacité des techniques et des outils, la productivité des activités de développement et donc la qualité du produit.
- ✓ Mesurer pour mieux évaluer le processus, comprendre les changements qui lui sont apportés et ainsi l'améliorer.
- ✓ Mesurer pour prévoir de nouvelles activités, l'effort requis, les coûts de développement et autres.

La mesure en génie logiciel est venue donc pour contrôler les coûts, améliorer la qualité et respecter les délais de livraison.

Comme la maintenance est 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 [6], la mesure au sein de ce processus s'avère primordiale afin de pouvoir livrer des produits et service de qualité.

	Sujet d'étude MGL 804	DOCUMENT NO.	DATE 2011-03-03	VERSION 1.0
	TITRE Modèle de mesure du processus maintenance de logiciel		PAGE 6	PAGES 22

3. Mesure de la maintenance de logiciel

3.1 Concept de la mesure de la maintenance de logiciel (ISO/IEC 14764)

La qualité de logiciel possède une considération importante dans la maintenance du produit logiciel. Les mainteneurs doivent avoir un programme de qualité de logiciel qui inclut les six caractéristiques de la qualité logicielle décrites dans la norme ISO 9126-1. Le processus de mesure de la norme ISO/IEC 12207 Amd 1(F.3.1.6) doit être implémenté pour identifier, définir, sélectionner, appliquer, valider et améliorer la mesure de logiciel pour la maintenance.

Une partie de la mesure de logiciel, le mainteneur doit déterminer l'effort de ressource pour la maintenance corrective, préventive, adaptative et perfective. Les données doivent être collectées, analysées et interprétées dans le but de faciliter l'amélioration du processus de maintenance et obtenir une meilleure compréhension de l'endroit où les coûts de maintenance sont dépensés.

3.2 Problématique liée à la maintenance

La maintenance des logiciels constitue une partie importante (de 50% à 80%) du coût total du cycle de vie d'un logiciel d'une part et d'autre part, la maintenance joue un rôle primordial dans le cycle de vie du produit logiciel. D'après les estimations, plus de 100 billion de lignes de code sont en production à travers le monde. Comme jusqu'à 80% de ces logiciels ne sont ni structurés, ni corrigés et non pas bien documentés. Seule la maintenance peut atténuer ces problèmes.

Quatre grands problèmes peuvent ralentir le processus de maintenance:


- code non structurées, les programmeurs de maintenance ayant une connaissance insuffisante du système,
- documentation étant absente, n'est pas à jour, ou, au mieux, insuffisante,
- la maintenance des logiciels a une mauvaise image.

Le succès de la maintenance dépend à ce que ces problèmes soient résolus plus tôt dans le cycle de vie.

D'une autre part, l'une des préoccupations majeures de tout organisme de maintenance est de comprendre et estimer le coût de maintenance de systèmes logiciels. [10]

3.3 La mesure du processus maintenance

Pour la mesure de la maintenance, ces questions doivent se poser : Quels sont les processus et les principales activités qui sont susceptibles d'être nécessaires pour mesurer les aspects de performance de la maintenance de logiciels ? Lesquels sont importants pour les clients et les parties prenantes?

	Sujet d'étude MGL 804	DOCUMENT NO.	DATE 2011-03-03	VERSION 1.0
	TITRE Modèle de mesure du processus maintenance de logiciel		PAGE 7	PAGES 22

Il y a plusieurs facteurs avant de mettre en œuvre la mesure de la maintenance. Les mesures doivent être bien définies afin de caractériser au mieux le logiciel, les services ainsi que les processus de maintenance [9].

À l'appui de la gestion quantitative, l'organisme de maintenance logiciel doit définir et mettre en œuvre:[14]

- a) les définitions des mesures (les mesures sont généralement divisés en deux catégories:
 - i) les mesures externes, qui sont visibles par les clients
 - ii) des mesures internes, qui sont plus techniques et visant les caractéristiques des produits et du logiciel lui-même);
- b) les mesures cibles;
- c) les points de référence pour chaque mesure;
- d) un référentiel de mesure
- e) des modèles de mesure qui permettent d'estimer et anticiper les performances des processus de maintenance.


Les organismes de maintenance doivent utiliser et analyser les données recueillies sur la performance de l'exécution de ses processus afin de développer une connaissance quantitative de la qualité de:

- a) produits livrables;
- b) ses services;
- c) la performance de l'exécution de ses processus, et
- d) les technologies utilisées

3.3.1 Processus de la maintenance

La norme internationale (ISO/IEC 14764) distingue six processus de maintenance logicielle :

- ✓ L'implémentation;
- ✓ L'analyse et la résolution de problèmes;
- ✓ La modification du logiciel;
- ✓ L'acceptation de la modification par le demandeur;
- ✓ La migration;

	Sujet d'étude MGL 804	DOCUMENT NO.	DATE 2011-03-03	VERSION 1.0
	TITRE Modèle de mesure du processus maintenance de logiciel		PAGE 8	PAGES 22

- ✓ Et finalement, la mise à la retraite.

Il y a un certain nombre de processus, d'activités et de règles qui sont propres à la maintenance du logiciel, par exemple :

- ✓ La transition : la passation coordonnée du logiciel, de l'équipe de développement à l'équipe de maintenance;
- ✓ Les ententes de services applicatives;
- ✓ La priorisation des requêtes de modification et des rapports de problèmes;
- ✓ L'acceptation ou le rejet d'une demande de modification selon le travail et la complexité plus ou moins grande qu'elle implique son renvoi à l'équipe de développement.

3.3.2 Les activités de la maintenance


Les activités de la maintenance peuvent être regroupées, selon ISO14764, en quatre catégories principales:

- **Maintenance *corrective*** - Maintenance corrective implique la modification réactive exécutée après la mise en production d'un logiciel pour corriger les problèmes découverts par sa clientèle [9]. Le coût de la maintenance corrective est presque toujours porté par le fournisseur.
- ***maintenance Adaptive*** - Maintenance adaptative implique la modification d'un logiciel exécutée après sa mise en production et qui vise à maintenir le logiciel utilisable dans un environnement technique changé ou en cours d'évolution [9]. Généralement Le client supporte le coût de la maintenance adaptative, mais il dépend aussi du contrat entre le fournisseur et le client.[15]
- ***maintenance perfective*** - maintenance perfective implique la modification d'un logiciel après sa mise en production pour améliorer son exécution ou sa maintenance [9]. Le coût de la maintenance perfective peut être porté soit par le fournisseur ou par le client, selon le contrat établi entre les deux [15].
- ***Maintenance préventive***- implique la modification d'un logiciel après sa mise en production pour détecter et corriger les défauts latents avant qu'ils deviennent des défaillances [9].

Les travaux de maintenance adaptative et perfective sont considérés comme des améliorations au logiciel existant.

Le financement pour des améliorations provient presque toujours de la clientèle.

Le travail des catégories est défini en termes de temps de changement (proactif ou réactif) et le but de changement (correction ou amélioration)

	Sujet d'étude MGL 804	DOCUMENT NO.	DATE 2011-03-03	VERSION 1.0
	TITRE Modèle de mesure du processus maintenance de logiciel		PAGE 9	PAGES 22

	Correction	Amélioration
Proactive	Préventive	Perfective
Réactive	Corrective	Adaptative

Catégories de maintenance selon ISO 14764 [9]

3.4 Modèles de mesure de la maintenance


3.4.1 survole de littérature

La maintenance des logiciels est un processus dynamique et de diverses techniques comme les modèles paramétriques basés sur l'analyse de régression, les réseaux neuronaux (neural networks), logique floue (fuzzy logic) et le raisonnement par cas en fonction (case based reasoning) sont utilisés pour tenter de prédire avec précision l'effort de maintenance de logiciels. Historiquement, le SLOC (Source lignes de code) et la méthode point de fonction (FP) sont les métriques de produits les plus populaires utilisés dans la formulation de divers modèles d'estimation de l'effort. En utilisant seulement SLOC ignore la contribution d'autres mesures internes de produits, tels que la complexité dans la détermination de l'effort de programmation. Ce qui est nécessaire est une mesure plus représentative du produit qui est à la fois simple à calculer et fournit une réponse rapide [2]. Les praticiens ont lutté avec trois questions fondamentales [17]

- Quels sont les modèles d'estimation des coûts de logiciels à utiliser?
- Quel Mesure de logiciel à utiliser - lignes de code (LOC), points de fonction?
- Qu'est-ce qu'une bonne estimation?

Les études d'Abran et Pfleeger ont démontrées que la maintenance du logiciel est différente du développement à cause des caractéristiques suivantes Abran et Al [13] :

- La taille et la complexité de chaque demande de travaux de maintenance sont, en général traitées par une ou deux ressources;
- les demandes de travaux de maintenance sont plus ou moins aléatoires et ne peuvent être comptabilisées individuellement dans le processus annuel de planification budgétaire;
- les demandes d'améliorations mineures (adaptative) de la catégorie d'amélioration sont examinées avec les clients et les priorités peuvent être affectées;
- La charge de travail de maintenance n'est pas géré en utilisant des techniques de gestion de projet, mais plutôt avec des techniques de gestion de la file d'attente;

	Sujet d'étude MGL 804	DOCUMENT NO.	DATE 2011-03-03	VERSION 1.0
	TITRE Modèle de mesure du processus maintenance de logiciel		PAGE 10	PAGES 22

- La maintenance a une portée plus large de la gestion de configuration avec plus de considérations opérationnelles.

Un grand nombre de modèles a été proposé pour estimer l'effort de développement de logiciels et les coûts en utilisant FPA, mais il y a très peu de modèles qui peuvent être utilisées pour estimer l'effort de maintenance et le coût de la taille du logiciel mesurée par ces nouvelles méthodes de dimensionnement [12]

Le développement des modèles d'estimation en s'appuyant sur la taille fonctionnelle des améliorations du logiciel, en particulier comme l'un des paramètres essentiels dans la modélisation de la relation entre la taille fonctionnelle et l'effort des travaux de maintenance, fait l'objet de plusieurs travaux de recherche.


Au début des années 1980, quelques tentatives ont été faites au profit de l'estimation de maintenance de logiciels. Il y avait le modèle COCOMO-M (aintenance), avec une seule entrée maintenance, le trafic variation annuelle. Dans ce modèle, c'est le coût du cycle de vie entier de maintenance qui est estimé, sur une période de temps, avec un tel modèle, aucune tentative n'est faite pour estimer un projet de maintenance à la fois. Ferens [7] rapporte que SLIM a une seule entrée d'entretien unique, alors que Price-S, SEER-SEM et CHECKPOINT ont des entrées multiples. Toutefois, Ferens [7] souligne que la précision de ces modèles d'estimation de la maintenance au début de leurs analyses était faible et qu'en général, leur performance n'a pas encore été démontrée [8].

3.4.2 Les travaux sur les modèles de mesure de la maintenance dans la Littérature [17]

Bailey et Basili ont identifié près de 100 caractéristiques environnementales en tant que contributeurs possibles à la variance de l'effort prédit. Ils ont reconnu que la détermination d'un sous-ensemble des attributs peut être fait par l'intuition d'experts, l'analyse factorielle ou par l'utilisation de matrices de corrélation. Ils ont finalement formé 3 groupes en utilisant seulement 21 des attributs d'origine.

Jorgensen a rapporté des expériences du développement et avait utilisé onze différents modèles de prédiction de l'effort de maintenance de logiciels. Une revue des études sur l'estimation de l'effort de développement de logiciels est présentée dans Les modèles ont été développés par l'application de l'analyse par régression, réseaux de neurones et reconnaissance des formes. Dans une autre publication, l'exactitude de prédiction des modèles ci-dessus a été comparée.

Sneed a proposé des moyens (et présenté «Softcalc 'une méthode de mise en œuvre) afin d'étendre les méthodes actuelles d'estimation des coûts pour couvrir les estimations des coûts de maintenance. Lucia et al. ont présenté un modèle d'estimation de l'effort de la maintenance de logiciel qui a été construit en utilisant une analyse de régression et validé par un processus de maintenance adaptative massive utilisée par EDS Inc.

	Sujet d'étude MGL 804	DOCUMENT NO.	DATE 2011-03-03	VERSION 1.0
	TITRE Modèle de mesure du processus maintenance de logiciel		PAGE 11	PAGES 22

Niessink et Vliet [19] ont présenté les résultats de deux programmes de mesures qui visaient à étudier les facteurs de coût possible pour la maintenance des logiciels. Ils ont utilisé l'analyse par composantes principales et analyse de régression multiple pour analyser les ensembles de données. Ils ont noté que l'existence d'un processus appliqué de manière cohérente est une condition importante pour un programme de mesure de succès.


Fuzzy et modèles neuronaux ont été utilisés pour estimer les lignes de codes, l'effort, maintenabilité du logiciel, intelligibilité de logiciel etc. Récemment Ahn et al. ont proposé un projet de modèle d'estimation de l'effort pour la maintenance de logiciel qui repose sur les points de fonction pour calculer le volume de la fonction maintenance. Dix facteurs d'ajustement de valeur (VAF) ont été considérés et regroupés en trois catégories de caractéristiques de maintenance, qui sont compétences de l'ingénieur (domaine personnes), les caractéristiques techniques (domaine de produits) et de l'environnement de maintenance (domaine de processus).

Les informations adéquates relatives à la taille, la complexité, la fiabilité, la maintenabilité des systèmes, du logiciel est souvent absentes. C'est encore aggravée par des facteurs humains et de gestion liées aux activités de maintenance telles que focus gestion, l'attitude des clients, l'attitude des ingénieurs, le besoin pour des équipes de support multi-sites, etc. Ces facteurs rendent le problème de l'objectivité d'estimer l'effort de maintenance de logiciels presque insoluble. Les coûts de support logiciel sont souvent financés par des «niveau d'effort" contrats de type (LOE) et non sur la base des exigences spécifiques de soutien. Le besoin est de développer un modèle de système dynamique comme proposé dans pour prédire l'effort de maintenance de logiciels, basée sur des données qualitatives et quantitatives.

Sur la base des travaux antérieurs ci-dessus les trois défis principaux suivants ont identifiés que l'estimation de l'effort doit permettre à améliorer les modèles actuels pour plus précise prédiction de l'effort de logiciels / coût:

- ✓ modèles d'estimation doit être en mesure de traiter l'information vague. En effet, la plupart des attributs du projet logiciel sont mesurés sur une échelle composée de valeurs linguistiques telles que basse et haute.
- ✓ modèles d'estimation doit traiter correctement les incertitudes dans les estimations.
- ✓ modèles d'estimation doivent apprendre des situations antérieures pour tenir compte de l'évolution constante du développement des logiciels et de la technologie de maintenance.

Des modèles d'estimation sont basés sur le concept générique de la productivité défini comme le rapport (ratio) des sorties et des entrées. Dans les projets de logiciels, la productivité est définie comme le rapport entre le produit logiciel développé et les ressources nécessaires pour le produire. Alors que «l'effort» est la mesure généralement acceptée de l'entrée ou de ressources (souvent mesurées en heures-personnes, jours ou mois), pour une sortie ou d'un produit, la taille du logiciel est reconnu comme un facteur clé dans les modèles d'estimation de l'effort. [11]

	Sujet d'étude MGL 804	DOCUMENT NO.	DATE 2011-03-03	VERSION 1.0
	TITRE Modèle de mesure du processus maintenance de logiciel		PAGE 12	PAGES 22

3.4.3 Techniques de modélisation des estimations

Pour établir des modèles d'estimation, la techniques de régression linéaire est utilisée afin d'analyser les données historique. Cette technique est aussi appliquée en combinaison avec la transformation logarithmique des données collectées ensuite les résultats démontreront la corrélation entre la taille fonctionnelle et l'effort et entre la taille fonctionnelle et le coût à travers le coefficient de corrélation R^2 . La qualité de prédiction de ces modèles est meilleure par rapport à celle des modèles qui ne l'utilise pas [12].

3.4.4 Pourquoi estimer la maintenance?

Grady et Caswell [Gra87] discutent la mise en place d'un programme corporatif de la mesure du logiciel, dans lequel sont décrits les formulaires et la collecte des données pour la mesure de la maintenance du logiciel. Il y a des mesures du logiciel qui sont communes à tous les domaines, incluant les catégories suivantes identifiées par le Software Engineering Institute-SEI: la taille, l'effort, l'échéancier et la qualité [Pig97]. Ces mesures constituent un bon point de départ pour le mainteneur [16].

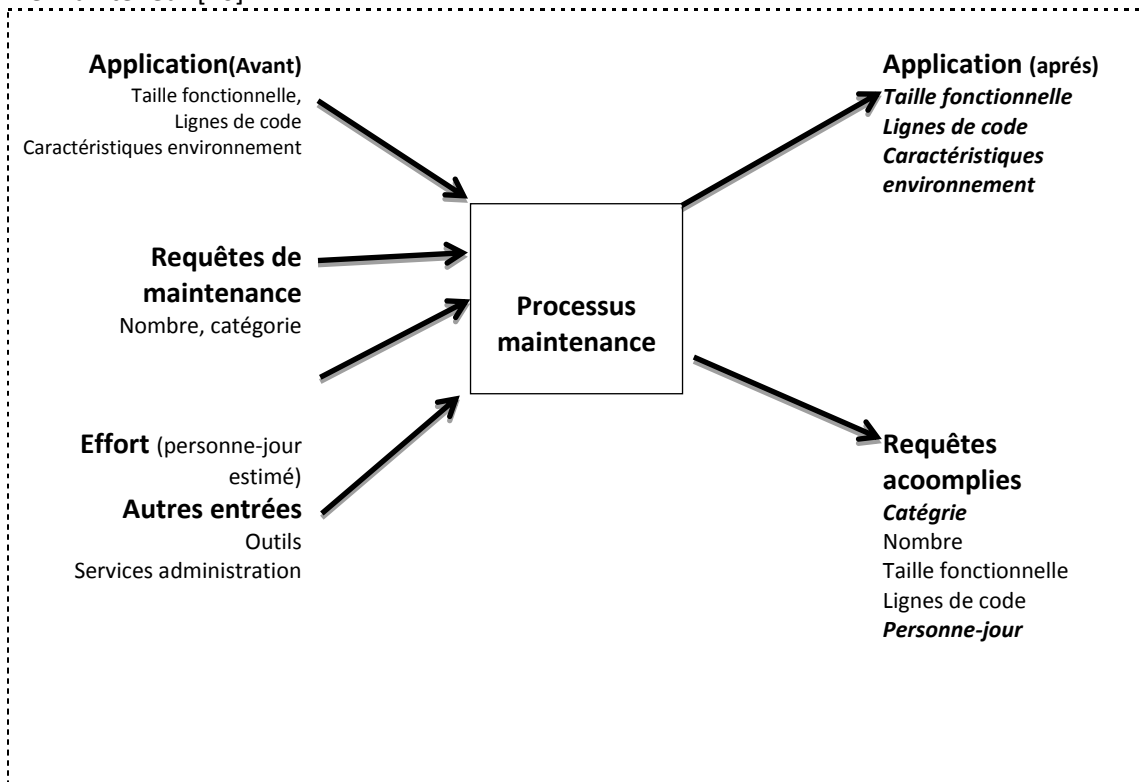



Figure 1 : processus maintenance [21]

	Sujet d'étude MGL 804	DOCUMENT NO.	DATE 2011-03-03	VERSION 1.0
	TITRE Modèle de mesure du processus maintenance de logiciel		PAGE 13	PAGES 22

La figure 1 montre les caractéristiques mesurable en gras italique, du processus de maintenance.


- Taille fonctionnelle : mesurée en nombre de points de fonction (FP)
- Lignes de code : LOC, nombre total des lignes de code de l'application, commentaires et espaces exclus
- Effort requis pour accomplir une requête de maintenance : mesuré en personne-jour.
- *catégories de maintenance*
- *Caractéristiques de l'environnement* : un questionnaire auquel le gestionnaire en charge de chaque application doit répondre.

Le questionnaire comprend les informations concernant [20]

- Identification de l'application
- Contraintes techniques (temps de réponse, sécurité, nombre d'utilisateurs, plateformes)
- Outils et techniques de maintenance (méthodologie de développement, cas d'utilisation)
- Facteur en relation avec le Personnel (nombre de programmeurs, expérience)

La table, ci-dessous présente l'objectif de la mesure de maintenance logiciel selon l'étude [20]

Objectif	Question	Métriques
Maximiser satisfaction client	Combien de problèmes affectent le client?	-Carnet de changement courant -Fiabilité logiciel
	Combien de temps faut-il pour fixer une urgence ou un problème urgent?	Changer cycle de temps depuis date d'approbation et depuis date écrite
Minimiser le coût	Combien coûte une livraison de logiciels de maintenance?	\$/ livraison
	Comment les coûts sont attribués?	\$/activité
	Quel genre de changements est à apporter?	Nombre de changements par type
	Combien d'effort déployé par type de changement?	Personnel-jours-prévus / changement par type
	Combien de demandes de changement dupliquées ou non valides sont-elles évaluées?	% requêtes de changement invalides et dupliqués fermées par mois.
	Quelle est la difficulté de la livraison?	-Évaluation de complexité maintenabilité logiciel

	Sujet d'étude MGL 804	DOCUMENT NO.	DATE 2011-03-03	VERSION 1.0
	TITRE Modèle de mesure du processus maintenance de logiciel		PAGE 14	PAGES 22

Minimiser le calendrier		-utilisation ressource ordinateur
	Combien de changements sont apportés au contenu de livraison prévue?	% contenu de changements par livraison
	Répondons-nous aux délais de livraison?	% livraison à temps

a) Maximiser la satisfaction du client

À travers les enquêtes et les interviews, la satisfaction du client peut être déterminée en cas où :

- ✓ le système ne comporte pas des problèmes qui pourraient affecter sa capacité de compléter ses tâches.
- ✓ La résolution des réclamations du client ne prend pas longtemps.
- ✓ Le fournisseur respecte ses engagements

Des métriques ont été proposées dans cette étude [20], pour maximiser la satisfaction du client comme le nombre de problèmes qui affectent le client, le temps nécessaire pour résoudre ces problèmes et si oui/non les délais de livraison sont respectés.

b) Minimiser les coûts

Le coût de maintenance est distribué par les différentes catégories d


En mesurant le coût pour chaque livraison et le pourcentage du coût que chaque activité contribue, les gestionnaires pourront diriger leurs efforts de réduction des coûts dans certains domaines.

3.4.5 Estimation de l'effort

En fait, l'application effective de tout modèle d'estimation de l'effort exige de connaître d'avance les valeurs des variables indépendantes incluses dans le modèle; comme ces valeurs ne sont pas connues lorsque le modèle est appliqué, elles doivent être estimées.

L'estimation de l'effort, selon la littérature, peut dépendre de la taille fonctionnelle seule, ou taille fonctionnelle avec d'autres facteurs comme :

- Type de logiciel
- Expérience de l'équipe des mainteneurs
- Domaine d'expertise

	Sujet d'étude MGL 804	DOCUMENT NO.	DATE 2011-03-03	VERSION 1.0
	TITRE Modèle de mesure du processus maintenance de logiciel		PAGE 15	PAGES 22

- Implication du client

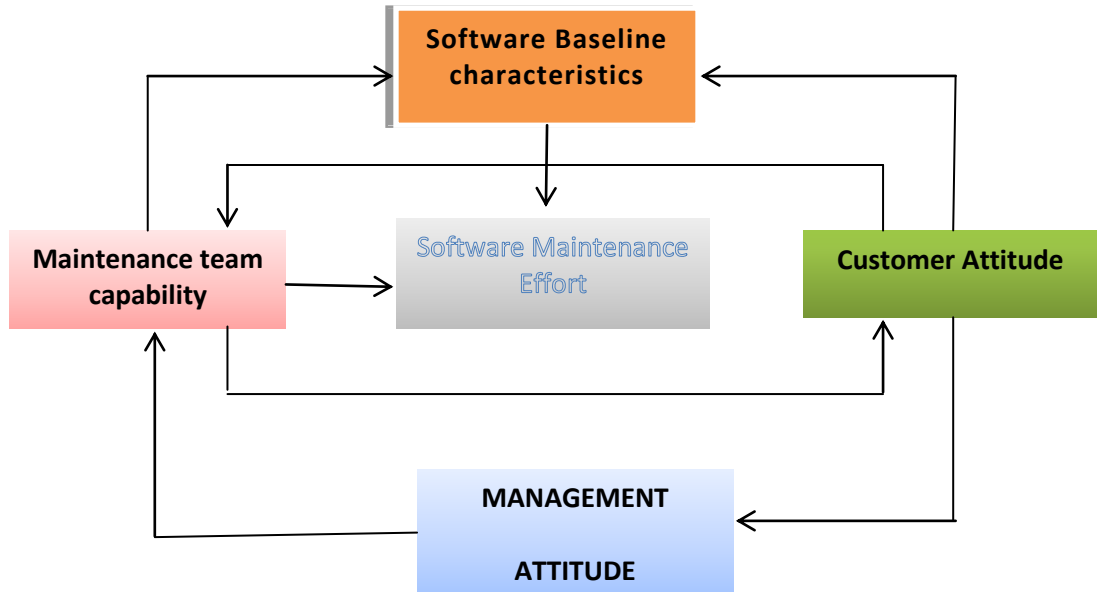



Figure2 : Estimation de l'effort de la maintenance logiciel [22]

Cette figure nous montre que l'effort de la maintenance est influencé par deux types de paramètres :

- **Capacité** : vue de l'angle du logiciel à maintenir et de l'équipe impliquée dans la maintenance de logiciel comme suit :
 - ✓ **Caractéristiques de base du logiciel** : si on ne pourrait les citer toutes, quelques une qui ont un impact direct sur l'effort sont la taille du logiciel, la complexité, la maintenabilité, l'historique de la maintenance et la documentation.
 - ✓ **Capacité de l'équipe de maintenance** : quelques attributs qui permettent de définir la capacité de l'équipe de maintenance sont l'expertise technique (connaissances de l'environnement de la maintenance comme les systèmes d'exploitation, bases de données, applications serveurs, langages de programmation, environnement de développement interactif, etc.), domaine d'expertise, connaissance d'application, attitude de programmeur
- **Attitude** : après avoir décrit les capacités des paramètres, nous avons besoin de voir l'attitude reliée aux paramètres et son impact sur l'effort.

	Sujet d'étude MGL 804	DOCUMENT NO.	DATE 2011-03-03	VERSION 1.0
	TITRE Modèle de mesure du processus maintenance de logiciel		PAGE 16	PAGES 22


- **Customer attitude**
 - ✓ **Alignement des domaines d'affaire TI** : parfois les équipes informatiques de l'organisation ne sont pas complètement en phase avec les utilisateurs professionnels. Cela conduit à des changements dus à une mauvaise interprétation) des caractéristiques des problèmes et des
 - b) la production des demandes de changement pour répondre aux exigences des utilisateurs professionnels non servis par le système d'exploitation.
 - ✓ **Implication** : le degré de réponse de l'équipe client est de répondre à toutes les questions soulevées par l'équipe de maintenance.
 - **Attitude de gestionnaire** : quelques attributs comme manque de personnel de maintenance, manque d'opportunité de formation pour l'équipe de maintenance, environnement fourni à l'équipe, à quel degré sont réaliste les dates de livraison, participation des développeurs.

3.4.6 Estimation de la taille fonctionnelle

La taille fonctionnelle est souvent utilisée comme variable indépendante et considérée comme entrée clé des modèles d'estimation de coût. La littérature a présenté un grand nombre de méthodes pour estimer la taille comme les lignes de code (LOC), l'analyse des points de fonction (FPA), les variantes de FPA, COSMIC Full Function Points (FFP) etc. Un grand nombre de modèles utilisés pour l'estimation de l'effort et du coût à partir de la taille mesurée par ces méthodes.

- Les premières méthodes d'estimation de la taille fonctionnelle des logiciels sont apparues dans les années 60. Elles se basent principalement sur la prédiction de la taille des composantes du logiciel telles que le nombre de lignes de code (LOC) des procédures, des modules, des objets existants. La taille estimée ainsi mesurée est bien utilisée pour prédire le temps et l'effort nécessaires mais l'inconvénient majeur de cette méthode est que la taille obtenue dépend fortement des technologies et des langages de programmation utilisés [22].

- La deuxième génération des méthodes, en particulier, FPA et UCP, se base, quant à elle, sur les artefacts produits durant la phase d'analyse et de conception du logiciel : la méthode FPA (Functional Point Analysis apparue en 1979) se concentrant sur les exigences fonctionnelles de l'utilisateur et la méthode UCP (Use Case Point en 1993) se concentrant sur les cas d'utilisation. En général, ces méthodes évaluent la complexité du logiciel (à l'aide de 3 niveaux allant de simple à complexe en passant par moyen) en mesurant une liste de facteurs techniques et contextuels sur base de l'expérience de l'évaluateur et sur base de données statistiques. La complexité du logiciel ainsi mesurée représente la taille fonctionnelle. Néanmoins, l'inconvénient de ces méthodes est que la taille mesurée varie considérablement en fonction de l'expérience des évaluateurs. De plus, elles ne permettent pas de mesurer la taille de certains types d'applications telles que les logiciels de systèmes distribués avec des contraintes temps réel ou les logiciels de gestion caractérisés par une grande quantité de transactions de données.

	Sujet d'étude MGL 804	DOCUMENT NO.	DATE 2011-03-03	VERSION 1.0
	TITRE Modèle de mesure du processus maintenance de logiciel		PAGE 17	PAGES 22

COSMIC-FFP est une méthode récente d'estimation qui présente certains avantages permettant de relever le défi.

- ▶ COSMIC présente l'avantage de pouvoir s'appliquer tôt dans le cycle de développement du logiciel.
- ▶ COSMIC peut s'appliquer sur un plus grand ensemble de composants/applications logiciels : elle permet de mesurer tous les logiciels caractérisés par des mouvements de données tels que les logiciels de gestion et les logiciels des systèmes en temps réel. Elle s'applique également aux nouveaux logiciels aussi bien qu'aux logiciels existants en maintenance corrective ou évolutive.

Dans un contexte de maintenance, la taille fonctionnelle d'une modification est calculée :


$$Taille_{UFC} \text{ (Changement)} = \Sigma \text{taille (données_mouvement_ajoutées)} + \Sigma \text{taille (données_mouvement_changées } i) + \Sigma \text{taille (données_mouvement_supprimées } i)$$

4. Métriques proposés par l'ISBSG

L'ISBSG propose les métriques dérivés les plus utiles pour la gestion de la maintenance et activités de support suivants :

Pour l'organisation

- **La productivité globale calculée**
Dérivation: Taille du portefeuille / M & S de l'effort pour l'organisation exprimée en FP/personnel année ou SLOC / Personnel Yr effort.
Définition: Il s'agit d'une mesure de la rentabilité de l'appui et la fonction maintenance. La mesure sera exprimée en FP / Effort total du personnel (EST) par an ou SLOC / TSE par an. Selon la mesure de dimensionnement déclaré par l'organisation qui soumet les données.
Impact: Aucun, la productivité sera cependant un élément important de données de référence dans les comparaisons entre et au sein des organisations.
- **Proportion de maintenance**
Dérivation: Effort Maintenance / l'effort M & S pour l'organisation, exprimée en pourcentage.
Définition: Il s'agit de la proportion de l'effort consacré par le personnel aux activités de maintenance.
Impact: Non connu- ce sera un champ utile de comparaison et d'analyse. suivant l'analyse des données une relation peut émerger.
- **Proportion Amélioration mineure**
Dérivation : Effort de maintenance adaptative + effort maintenance perfectif / l'effort M & S pour l'organisation, exprimée en pourcentage.

	Sujet d'étude MGL 804	DOCUMENT NO.	DATE 2011-03-03	VERSION 1.0
	TITRE Modèle de mesure du processus maintenance de logiciel		PAGE 18	PAGES 22

Définition: Il s'agit de la proportion de l'effort de M & S consacrés à des améliorations mineures.

Impact: Non connu- il peut être un champ utile aux fins de comparaison. Après l'analyse des données, une relation entre ce domaine et coûts peuvent émerger.

- **La capacité du personnel**

Dérivation : taille d'équipe * Heures par personnel-an / l'effort de M & S

Définition: Cette mesure indique la capacité du personnel étant employé par une Organisation de la maintenance, mais qui est utilisée pour d'autres fins. Il est probable qu'il y aura des capacités de secours optimales, qui permettront de répondre rapidement aux problèmes tout en minimisant coûts. Ce qui peut être révélé par l'analyse de la base de données.

Impact: Si ce chiffre est trop important, les coûts seront plus élevés que nécessaire.

- **Densité de défauts de l'organisation**

Dérivation : Nombre de défauts par année/Taille du portefeuille
exprimé en erreurs par1000Fp ou des erreurs / KSLOC.

Définition: Le nombre d'erreurs pour 1000 points de fonction découvert au cours de l'année de collecte au sein du portefeuille de logiciels de l'organisation.

Impact: plus la densité d'erreur est élevée, plus grand sera l'effort et, par conséquent les coûts nécessaires à la maintenance et de soutien.

- **Taux d'appel Organisation**

Dérivation : Nombre d'appels / Taille du portefeuille
Exprimé en # appels /1000FP ou #appels / KSLOC.

Définition: Le nombre d'appels traités au cours de l'année liés à la perception de la taille du portefeuille de l'organisation.

Impact: On s'attend à ce que plus le nombre d'appels est élevé, grand sera l'effort requis pour les activités M & S.

Pour l'application

- **productivité de l'application**


Dérivation : Taille de l'application / effort pour l'application
Exprimée en 1000FP/ personnel.

Définition: Cette zone d'init la productivité de la fonction M & S associée à une application particulière.

Impact: La baisse de la productivité implique des coûts plus élevé

- **Proportion maintenance de l'application**

Dérivation : effort Maintenance / effort pour l'application
Exprimée en pourcentage.

	Sujet d'étude MGL 804	DOCUMENT NO.	DATE 2011-03-03	VERSION 1.0
	TITRE Modèle de mesure du processus maintenance de logiciel		PAGE 19	PAGES 22

Définition: Cette zone exprime la proportion en pourcentage de l'effort consacré à la maintenance par opposition aux activités d'amélioration mineure

Impact: Non connu. Une analyse ultérieure des données peut révéler certaines relations.

- **Proportion de demandes d'amélioration mineure de l'application**

Dérivation : l'effort de maintenance adaptative + perfectif / Effort M & S pour l'application exprimée en pourcentage.

Définition: Cette zone exprime la proportion des efforts déployés aux améliorations mineures de l'application.

Impact: Non connu. L'analyse des données peut révéler une certaine relation.

- **Densité de défauts de l'application**

Dérivation : défauts détectés par an / Taille de l'application exprimé que erreurs/1000Fp ou erreurs / KSLOC.

Définition: Le nombre d'erreurs, dans chacune des catégories pour l'année dans laquelle les données ont été recueillies, liée à la taille de l'application.

Impact: plus élevée la densité d'erreurs, plus élevé sera le coût des activités M & S pour cette application.

- **Effort par défaut**

Dérivation : Maintenance corrective / défauts détectés par an

Définition: Cette zone révèle l'effort moyen de corriger les erreurs au sein d'une application.

Impact: Ce champ devrait être un facteur important du coût global M & S.

- **Proportions du langage de Programmation**

Dérivation langage primaire KSLOC / Taille de l'application exprimée en pourcentage.

Définition: Il s'agit simplement d'une mesure qui révèle la proportion du langage de programmation principal utilisé pour construire l'application.


Impact: Inconnu. L'analyse peut révéler une certaine relation.

- **Effort par emplacement**

Dérivation : Effort pour l'application / Nombre d'emplacements utilisateur

Définition: Cette zone exprime le coût moyen par emplacement où l'effort de soutien doit être déployé.

Impact: Aucun. Il s'agit d'un état de la situation, qui peut être un comparateur utile.

	Sujet d'étude MGL 804	DOCUMENT NO.	DATE 2011-03-03	VERSION 1.0
	TITRE Modèle de mesure du processus maintenance de logiciel		PAGE 20	PAGES 22

Conclusion


La mesure maintenance logiciel est un processus qui reste compliqué est immature. On se demande toujours quelles mesure choisir, quoi mesurer et comment.

La diversité des modèles de mesure complique de plus en plus le processus de mesure car pour le choix du modèle dépend de l'étude exploratoire qu'il faut effectuer sur un échantillon de données d'un ensemble de projets. Les modèles qui ne sont pas évolués ne peuvent être utilisés comme référence

RÉFÉRENCES

[1] http://fr.wikipedia.org/wiki/Maintenance_du_logiciel

[2] Marc Frappier « Gestion de la maintenabilité ». Département de mathématiques et d'informatique, Université de Sherbrooke, Sherbrooke, Québec, J1K 2R1,

	Sujet d'étude MGL 804	DOCUMENT NO.	DATE 2011-03-03	VERSION 1.0
	TITRE Modèle de mesure du processus maintenance de logiciel		PAGE 21	PAGES 22

[3] [Maffezzini.I, Premiana.A, Ventimiglia.B : Prolégomènes à une critique du génie logiciel. Partie 2 qualité et mesures des produits.]

[4] <http://www.deltamu.fr/Metrologie/Historique.aspx>

[5] <http://fr.wikipedia.org/wiki/M%C3%A9trologie>

[6] LAURA PRIMERA. “Mesurer la taille fonctionnelle d’un échantillon des applications CENTREX et établir un modèle de productivité pour ces activités de maintenance” chez ABC Inc. Avril 2001

[7] Ferens, D.V. (1999) “The Conundrum of Software Estimation Models”, IEEE AES Systems Magazine, March, 1999, 23-29.

[8] https://softwaretechnews.thedacs.com/stn_view.php?stn_id=6&article_id=17

[9] April.A. Abran.A. Améliorer la maintenance du logiciel. Copyright 2006 Loze-Dion éditeur INC

[10] Victor Basili, Lionel Briand, Steven Condon, Yong-Mi Kim, Walcélio L. Melo and Jon D. Valett.

“Understanding and Predicting the Process of Software Maintenance Releases”


[11] Alain Abran*,†, Ilionar Silva and Laura Primera. “Field studies using functional size measurement in building estimation models for software maintenance” JOURNAL OF SOFTWARE MAINTENANCE AND EVOLUTION: RESEARCH AND PRACTICE. J. Softw. Maint. Evol.: Res. Pract. 2002; 14:31–64 (DOI: 10.1002/smr.245). Ecole de Technologie Supérieure, Montreal, QB, Canada

[12] De Tranc-Cao, Ghislain Levesque. “Maintenance Effort and Cost Estimation Using Software Functional Sizes”. Software Engineering Management Research Laboratory (UQAM)

[13] A.April, A.Abran, Reiner R, Dumke.”Software Maintenance Productivity: How to assess the readiness of your organization” IWSM/MetriKon 2004

[14] Alain April , Alain Abran , Reiner R. Dumke. « Software Maintenance Capability Maturity Model (SM-CMM): Process Performance Measurement»

[15] Arun Mukhija. “Estimating Software Maintenance”. Seminar on software cost estimation WS 02/03. Requirements Engineering Research Group Institut für Informatik Universität Zürich Prof. M. Glinz. January 21, 2003.

	Sujet d'étude MGL 804	DOCUMENT NO.	DATE 2011-03-03	VERSION 1.0
	TITRE Modèle de mesure du processus maintenance de logiciel		PAGE 22	PAGES 22

[16]SWEBOK chapitre 6

[17] Ruchi Shukla, Arun Kumar Misra “Estimating Software Maintenance Effort - A Neural Network Approach “.Department of Computer Science and Engineering Motilal Nehru National Institute of Technology. Allahabad, India

[18] Andrea De Lucia, Massimiliano Di Penta, Silvio Stefanucci, Gabriele Venturi “Early Effort Estimation of Massive Maintenance Processes”. RCOST- Research Centre on Software Technology Department of Engineering, University of Sannio. Palazzo Bosco Lucarelli, Piazza Roma -82100 Benevento, Italy. 1063-6773/02 @ 2002 IEEE . Proceedings of the International Conference on Software Maintenance (ICSM'02)

[19] Frank Niessin, Hans VanVliet. “Predicting Maintenance Effort with Function Point”. Faculty of mathematicsand Computer science, Vrije Universiteit Amsterdam. IEEE 1997

[20] George E. Stark “Measurement for managing software maintenance”. IEEE 1996.

[21] Jean-Marc Desharnais, France Paré, Marcela Maya, Denis St-Pierre “Implementing a measurement program in software maintenance- an experience based on Basili’s approach”. IFPUG 1997 Spring Conference.