

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

DÉPARTEMENT D'INFORMATIQUE
MAÎTRISE EN INFORMATIQUE DE GESTION

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

**ÉLABORATION D'UN OUTIL D'ÉVALUATION ET
D'AMÉLIORATION DU PROCESSUS DE MAINTENANCE
DES LOGICIELS : UNE ÉTUDE EXPLORATOIRE.**

par
MOHAMED ZITOUNI

MARS 1996

Rapport final d'activité de synthèse

approuvé par le Directeur de recherche : Alain ABRAN,
Ph.D en génie informatique.

Mr Abran est
Professeur au Département d'informatique de l'Université du Québec à Montréal (UQAM);
Directeur du LRGL (Laboratoire de recherche en gestion des logiciels) de l'UQAM;
Président du CIM (Centre d'intérêt sur les métriques) de Montréal (Canada).

Sommaire

Dans les dépenses allouées par les organisations au secteur du logiciel d'une façon générale, c'est leur maintenance qui accapare les budgets les plus importants. Dans grand nombre de ces organisations, les coûts associés à la maintenance du logiciel ont atteint des sommets : entre 50% et 70% des budgets d'opération de l'informatique.

Faute d'un modèle de gestion spécifique à la fonction de maintenance, les gestionnaires continuent d'administrer à cette fonction un traitement de crise en attendant l'arrivée sur le marché d'hypothétiques solutions rattachées à de nouvelles technologies de développement. Mais le poids des dépenses devrait pousser les dirigeants des organisations à accorder une priorité plus élevée à la fonction de maintenance des logiciels. C'est pourquoi il devient important d'étudier comment améliorer cette fonction de l'organisation et comment réduire ses coûts.

La réalisation de ce projet a consisté à effectuer une étude exploratoire sur l'élaboration d'un outil d'évaluation et d'amélioration du processus de la maintenance des logiciels. La finalité de cet outil est double. D'abord disposer de la capacité d'informer les gestionnaires et les dirigeants d'une organisation sur l'état de la fonction de maintenance de logiciels en opération. Pour cela, il faut évaluer le processus de maintenance, c'est-à-dire évaluer l'ensemble des activités de maintenance selon une échelle de maturité du processus, en utilisant par exemple une échelle allant d'un niveau le moins performant (niveau 1) au niveau le plus performant (niveau 5). Ensuite, dans le cas où l'organisation souhaite améliorer son processus de maintenance, disposer de la capacité de proposer, à partir des résultats de l'évaluation, un ensemble d'activités à réaliser et d'étapes à suivre afin d'amener le processus de maintenance à un niveau supérieur de performance.

La présente étude exploratoire propose un objectif précis : élaborer un prototype de modèle de maturité applicable au processus de la maintenance des logiciels. Ce prototype est largement inspiré du modèle CMM (*Capability Maturity Model*) du SEI (*Software Engineering Institute*) de l'Université Carnegie Mellon de Pittsburgh (Pennsylvanie, États-Unis), utilisé pour évaluer et améliorer le processus de développement des logiciels. Pour réaliser cet objectif, il a fallu identifier, décrire, structurer, modéliser et insérer les composantes du prototype de modèle à élaborer en se servant de la structure interne du modèle de maturité du SEI. Les résultats de cette adaptation de modèle ont débouché sur une proposition d'un nouveau modèle pour l'évaluation et l'amélioration des processus de maintenance des logiciels.

Cette étude s'est appuyée sur une démarche de type exploratoire faisant largement appel à la structure d'un modèle existant et à des ouvrages de référence portant spécifiquement sur le sujet de la maintenance et plus particulièrement sur des aspects de gestion et de pratiques les plus courantes appliquées au sein de la fonction de maintenance du logiciel dans les organisations.

La réalisation de l'étude a permis d'identifier des secteurs-clés et pratiques-clés de maintenance du logiciel non abordés dans le modèle de maturité CMM du SEI de l'Université de Pittsburgh. Ce modèle, rappelons-le a été conçu et élaboré dans un contexte de problématiques liées au développement du logiciel seulement.

La version 1.0 du modèle ainsi élaboré se compose de 21 secteurs-clés, 63 objectifs-clés et 312 pratiques-clés de maintenance répartis sur les niveaux de maturité 2 à 5. Un glossaire de termes spécifiques à la maintenance et contenant 112 descriptions facilite la compréhension des objectifs-clés et des pratiques-clés contenus dans la version 1.0 du modèle. D'autre part, des analyses du nouveau modèle ont permis de constater une approche multi-niveaux des secteurs-clés, c'est-à-dire la présence des pratiques-clés d'un secteur-clé dans plusieurs niveaux de maturité, contrairement au modèle CMM du SEI qui lie l'appartenance d'un secteur-clé à un seul niveau de maturité. Et enfin plusieurs pistes de recherche sont suggérées pour donner plus d'envergure aux résultats obtenus.

Remerciements

Je tiens tout d'abord et avant tout à remercier mon directeur de recherche, **Alain Abran**, à qui je dois beaucoup, beaucoup plus que la réalisation de ce travail. Notre rencontre a été fortuite au départ. J'ai suivi un de ses cours de maîtrise traitant d'évolution des technologies nouvelles. Lors d'une des séances de cours, il a parlé de maintenance de logiciels, un domaine que je courtisais fortement pour en faire la base de mon activité de synthèse. Notre relation venait de se nouer. J'ai commencé à lui parler de problématique et de contours du domaine de recherche qui m'intéressait, il m'a proposé le plus important : le cadre de recherche, une piste de solution et son encadrement. La relation a pris rapidement une trajectoire harmonieuse, amicale et solide. Je dois dire combien il est agréable d'évoluer dans un climat de confiance et de motivation. On y découvre sans explication aucune que ce qui peut être considéré souvent comme une corvée peut se transformer en plaisir avec son lot de satisfaction et de bonne ambiance. J'envisage de poursuivre des études de doctorat sous sa direction.

Pierre Bourque, Directeur-adjoint du LRGL (Laboratoire de Recherche en Gestion de Logiciels) de l'UQAM, est arrivé juste pendant l'élaboration de ma proposition de recherche pour laquelle il m'a aidé autant dans la forme que dans le fond. C'est une personne qui a l'oeil des gens expérimentés que l'on retrouve dans les entreprises et qui se font toujours un plaisir d'aider lorsque quelqu'un sollicite leur aide. Lors de la simulation de ma présentation devant un parterre considéré comme «acquis», il fut le plus « virulent » de tous : les questions étaient nombreuses, directes et appropriées. Pierre est un expert dans le domaine de l'ingénierie du logiciel. J'ai beaucoup appris avec lui. Depuis, je n'ai pas cessé de le consulter. C'est l'occasion pour moi de lui dire merci.

Mon travail de maîtrise a été enrichi par un groupe d'experts spécialisés dans le domaine des modèles de recherche en génie logiciel. Ces experts étaient chargés de passer au peigne fin le modèle élaboré tant dans le fond que dans la forme. Tout en remerciant l'ensemble du groupe, je souhaite particulièrement m'attarder un moment sur **Alain April** de la compagnie Bell Canada. Je sais qu'il n'est jamais suffisant de remercier quelqu'un pour sa disponibilité surtout lorsqu'il n'en dispose que de très peu. Il a réussi à insérer mon stress et mes préoccupations entre deux priorités et m'a consacré tout le temps nécessaire. Je lui dois reconnaissance et sympathie.

J'ai remarqué **Robert Dupuis** pour la première fois lors d'une conférence qui étalait pas moins de quatre vice-présidents informatiques de grandes sociétés (Banque nationale, Hydro-Québec, Desjardins et Bell Canada). Ce personnage barbu, en jean et à l'allure nonchalante a pris la parole pour tenir un propos intelligent, mêlé de subtilité, d'humour et surtout d'audace car il brisait en douceur l'élan d'autosatisfaction affiché par la brochette d'experts. J'ai gardé ce personnage dans ma tête sous la rubrique «originalité du corps enseignant de l'UQAM» jusqu'à ce que je le retrouve lors du cours de maîtrise consacré à la méthodologie de la recherche appliquée. Un professeur agréable et plaisant qui se trouve aujourd'hui à la tête du programme de maîtrise. En dehors, nos conversations ont constamment débordé de la sphère du programme de maîtrise ou de l'enseignement en général. A mon plus grand plaisir, bien entendu. Je lui dis merci pour ses précieux conseils.

Marc Bouisset est incontournable dans mon cheminement de maîtrise. Il a commencé d'abord par me faire l'inventaire de mes faibles capacités professionnelles pour me convaincre de l'utilité des études de maîtrise. Il a eu raison sur toute la ligne car l'ensemble de mon parcours professionnel se limitait à la connaissance des gros systèmes. La dernière fois que j'avais mis les pieds dans une université remontait à 1978 (Université Claude Bernard, Lyon I), un vrai fossile. En matière de connaissances ou d'expertise sur ce qui est d'actualité? Rien. Rien sur de nombreux points : Bureautique, réseautique, intégration de systèmes, télécommunications, orienté objet, veille technologique, etc. J'ai compris que ma formation et mon expérience antérieures servaient à peine de fond et de support pour ce qui me restait à faire, c'est-à-dire tout. Il fallait se reconstruire professionnellement. Si je n'avais pas croisé Marc, alors directeur

du programme de la maîtrise de l'époque, je serais certainement encore en face d'un 3270, avec l'enthousiasme des gens heureux en plus. Un lien amical fort caractérise notre relation. Je lui dis merci.

S'il y a un lieu commun au Département d'informatique où l'on passe et l'on repasse, c'est bien le secrétariat du Département. On commence par y mettre les pieds pour des requêtes précises; on continue d'y retourner pour des requêtes moins précises et on finit par y aller pour dire bonjour seulement. C'est le plaisir que l'on prend dans un endroit où les gens sont serviables et sympathiques. Rien ne se fait sans l'apport de toutes et de tous bien sûr mais lorsque c'est bien fait, de surcroît et avec la générosité en prime, alors un merci s'impose de lui-même. Merci à **Julie Hudon, Marjolaine Riendeau, Michèle Hébert et Monique Bissonnette.**

Le contexte de proximité dans lequel s'est effectué ce travail a été important : le laboratoire de recherche en gestion des logiciels de l'UQAM a mis à ma disposition équipements et capacité d'écoute. L'entraide et l'esprit de solidarité a joué souvent. Je dis merci aux étudiants de maîtrise du laboratoire : **David Dery, Aâdil Dakka, Olavo Mendez, Marcella Maya, Jude Nsiempba,** etc.

Un merci à **Pédro Zuniga**, technicien en Bureautique du laboratoire.

Un gros merci aussi à **Danielle Pinsonneault** pour l'immense aide apportée.

Je dédie enfin ce travail au **Castor**, à la **Belle** et au **Roi**.

Mohamed Zitouni.

Table des matières

Sommaire	v
Remerciements	vii
Liste des figures	xiii
Liste des tableaux	xi
Préambule	1
Chapitre 1 - Présentation du sujet.....	3
1.1 Problématique	3
1.1.1 La fonction de maintenance du logiciel dans l'organisation	5
1.1.2 Problèmes liés à la maintenance du logiciel	6
1.1.3 Absence d'outil d'évaluation et d'amélioration de la maintenance des logiciels	8
1.2 Importance du projet	9
1.2.1 Intérêts pour l'entreprise	9
1.2.2 Intérêts pour les chercheurs	9
1.2.3 Intérêts de l'auteur	10
Chapitre 2 - Contexte théorique.....	11
2.1 La maintenance du logiciel.....	11
2.1.1 Définition et types de maintenance.....	11
2.1.2 Particularités entre développement et maintenance.....	13
2.1.3 Positionnement de la maintenance dans le cycle de vie du logiciel	14
2.1.4 Processus de la maintenance	15
2.2 Évaluation et amélioration de processus : le modèle de maturité CMM du SEI	16
2.2.1 Historique du modèle.....	16
2.2.2 Concepts du modèle.....	17
2.2.3 Niveaux de maturité du modèle	19
2.2.4 Finalités du modèle	22
Chapitre 3 - Méthode d'élaboration du modèle CMM du SEI.....	25
3.1 Les fondements méthodologiques.....	25
3.1.1 Structure du modèle de maturité du SEI.....	28
3.1.2 Contenu du modèle de maturité du SEI.....	29

Chapitre 4 - Démarche d'intervention.....	31
4.1 Pourquoi le modèle CMM du SEI ne s'applique pas à la maintenance du logiciel	31
4.2 Élaboration d'un nouveau modèle de maintenance du logiciel.....	33
4.1.1 Flux d'entrée.....	35
4.1.2 Traitement.....	36
4.1.3 Sorties.....	36
4.3 Construction par itération et nouvelles versions des prototypes	36
4.4 Rôle des experts.....	37
4.5 Envergure de l'étude et limites de la méthode	39
Chapitre 5 - Résultats : la version 1.0 du modèle de maturité de la maintenance logiciel.....	43
5.1 Présentation du modèle.....	43
5.2 Contenus global et détaillé du modèle de la maintenance.....	45
5.3 Exemple de secteur-clé du modèle de la maintenance.....	46
5.3.1 Cadre de présentation des secteurs-clés du modèle de la maintenance.....	46
5.3.2 Contenu du secteur-clé "Organisation de la fonction de maintenance"	47
Chapitre 6 - Analyse des résultats	51
6.1 Analyse quantitative du modèle	51
6.1.1 Secteurs-clés du modèle de la maintenance.....	51
6.1.2 Pratiques-clés par niveau de maturité.....	52
6.1.3 Pratiques-clés par ouvrage de référence (auteur)	53
6.2 Comparaison par rapport au modèle CMM du SEI.....	55
6.3 Approche de secteurs multi-niveaux	57
6.4 Pistes potentielles de recherche.....	59
6.5 Diffusion des résultats.....	61
Bibliographie.....	63
Annexes	67
A - Guide des experts	69
B - Contenu du modèle de maturité de la maintenance version 1.0	77
C - Glossaire des termes employés dans le modèle de la maintenance version 1.0.....	119
D - Article ayant fait l'objet de publication dans le cadre de cette étude.....	153

LISTE DES FIGURES

Figure 1.1 - Espace de la problématique.....	4
Figure 2.1 - Cycle de vie du logiciel d'après la norme IEEE.....	15
Figure 2.2 - Processus de maintenance et instruments de gestion de la maintenance.....	16
Figure 2.3 - Cadre conceptuel de maturité de processus (<i>Framework</i> de Watts Humphrey).....	18
Figure 3.1 - Représentation interne globale du modèle de maturité du SEI.....	26
Figure 3.2 - Cadre méthodologique d'élaboration du modèle de maturité du SEI.....	27
Figure 3.3 - Structure globale du modèle de maturité du SEI.....	28
Figure 3.4 - Contenu global et détaillé du modèle de maturité du SEI.....	29
Figure 4.1 - Origine des requêtes de maintenance et la fonction de maintenance.....	32
Figure 4.2 - Cadre d'élaboration des livrables du projet de recherche.....	34
Figure 4.3 - Étapes d'élaboration du modèle de maturité de la maintenance.....	37
Figure 4.4 - Positionnement par rapport aux étapes d'élaboration du modèle.....	42
Figure 5.1 - Cadre global de présentation des résultats de l'étude.....	44
Figure 5.2 - Cadre standard de présentation du contenu du modèle de la maintenance version 1.0.....	46
Figure 5.3 - Codification des objectifs-clés et des pratiques-clés.....	48
Figure 6.1 - Pratiques-clés par niveau de maturité du modèle de la maintenance version 1.0.....	53
Figure 6.2 - Pratiques-clés par ouvrage de référence (auteur) du modèle de la maintenance version 1.0.....	54

LISTE DES TABLEAUX

Tableau 2.1 - Effort et types de maintenance (répartition de l'effort en pourcentage).....	12
Tableau 2.2 - Caractéristiques d'améliorations majeures <i>versus</i> activités de maintenance.....	14
Tableau 2.3 - Les secteurs-clés par niveau de maturité du modèle de maturité du SEI	21
Tableau 2.4 - Résultats du modèle de maturité du SEI auprès des organisations	23
Tableau 4.1 - Types de maintenance à réaliser selon l'origine de la requête et l'activité qui l'encadre.....	32
Tableau 4.2 - Secteurs d'activité de gestion du logiciel (P = présent, A = absent).....	33
Tableau 5.1 - Contenus global et détaillé du modèle de maturité de la maintenance.....	40
Tableau 5.2 - Description des pratiques-clés du secteur-clé 10 du modèle de la maintenance.....	50
Tableau 5.3 - Comptabilisation du secteur-clé 10 "Organisation de la fonction de maintenance logiciel"	50
Tableau 6.1 - Les secteurs-clés du modèle de maturité de la maintenance (version 1.0).....	52
Tableau 6.2 - Contenus du modèle de maintenance <i>versus</i> contenus du modèle CMM du SEI	56
Tableau 6.3 - Caractéristiques de qualification de processus par niveau de maturité	57
Tableau 6.4 - Pratiques-clés du secteur-clé : Planification de la maintenance logiciel	58
Tableau 6.5 - Approche des secteurs multi-niveaux du modèle de la maintenance version 1.0.....	59

Préambule

Dans le domaine des technologies de l'information, l'étude de la fonction de maintenance du logiciel et les stratégies de remplacement de cette fonction dans l'organisation sont un sujet important autant pour les chercheurs que pour les gestionnaires sur le terrain.

Dans les dépenses allouées par les organisations au secteur du logiciel, c'est leur maintenance qui consomme les budgets les plus importants. Dans grand nombre d'entreprises, les coûts de la maintenance du logiciel ont grimpé rapidement pour représenter entre 50% et 70% des budgets d'opérations de l'informatique et vont probablement continuer de croître. Devant des dépenses de cet ordre, il devient important pour les organisations qu'un effort soit fait pour étudier comment améliorer la fonction de maintenance du logiciel et comment réduire ses coûts.

En plus de l'absence de modèle spécifique à la gestion de maintenance, plusieurs facteurs contribuent à aggraver le problème. Les changements rapides subis dans l'environnement de travail et les besoins des utilisateurs sans cesse croissants obligent les organisations à procéder à des modifications et à des améliorations des logiciels d'une façon continue pour s'adapter au changement. Cette dynamique génère des besoins de contrôle supplémentaires et des interactions de modules et de logiciels qui aboutissent fréquemment à une plus grande complexité des logiciels et par conséquent à des coûts plus importants pour leur maintenance.

Dans le domaine du développement, la qualité d'un logiciel est tributaire de la qualité de son processus de développement, c'est-à-dire qu'un logiciel est considéré de qualité optimale lorsque chaque activité par laquelle il a transité pour être développé est optimale. Certains chercheurs considèrent que la maintenance peut et devrait être gérée d'une façon "presque" similaire à celle du développement. De ce fait, le cadre conceptuel et le modèle qui sont à la base des outils d'évaluation et

d'amélioration du processus de développement de logiciels peuvent se prêter à un exercice similaire pour le processus de la maintenance des logiciels.

La présente étude exploratoire porte sur l'élaboration d'un outil d'évaluation et d'amélioration du processus de la maintenance du logiciel dont la structure s'inspire largement du modèle CMM (*Capability Maturity Model*) du SEI (*Software Engineering Institute*) de l'Université Carnegie Mellon de Pittsburgh (Pennsylvanie, États-Unis).

Le chapitre 1 de ce document présente le projet de l'étude avec une description de la problématique et des intérêts que suscite la recherche d'une solution d'amélioration de la fonction de maintenance tant dans le milieu de la recherche que dans les organisations en général. Dans le chapitre 2 sont exposés les fondements conceptuels. Ces fondements reposent sur deux aspects : la maintenance du logiciel et le modèle de maturité du SEI. Le chapitre 3 présente les fondements méthodologiques sur lesquels s'appuie la présente élaboration du modèle de maturité de la maintenance des logiciels. Le chapitre 4 décrit la démarche d'intervention pour construire le modèle. Dans le même chapitre sont décrits la construction et les ajustements du modèle qui se réaliseront par itération à partir de données provenant d'ouvrages spécifiques à la maintenance et appuyées par l'apport d'un groupe d'experts. Le chapitre 5 décrit les résultats de l'étude en mettant l'accent sur une partie représentative des résultats. Le chapitre 6 est consacré à l'analyse du nouveau modèle de maintenance proposé. Dans ce chapitre, l'analyse est d'ordre quantitatif et comparatif par rapport au modèle de maturité du SEI qui a servi de base pour sa construction.

Chapitre 1

Présentation du sujet

Ce premier chapitre du rapport vise à préciser l'objet de la présente étude. Son contenu s'articule autour de trois aspects : la problématique à laquelle est confrontée la maintenance du logiciel dans les organisations, les objectifs visés par la présente étude, l'importance du projet et les différents intérêts que suscite sa réalisation.

1.1 Problématique

Il ne fait aucun doute pour les organisations que les technologies de l'information détiennent un rôle important dans chaque stratégie d'affaires [Benjamin, R.I., 1984; Porter and Millar, 1985]. Le fléchissement économique de la dernière décennie, qui a mis à rude épreuve les profits des entreprises, a fini par sensibiliser les directions des organisations sur la contribution des technologies de l'information comme instrument d'amélioration du fonctionnement des différentes fonctions de l'organisation. Des liens directs doivent être tissés entre le Département de l'informatique et l'organisation à travers des stratégies conformes, [King 1978; Boynton et Zmud, 1987; Cash et al., 1988]

A travers l'ouvrage de Swanson & Beath (1989), il ressort qu'une des fonctions de l'organisation qui devient de plus en plus importante aux yeux des gestionnaires et des dirigeants est la maintenance des logiciels. Cela est dû principalement aux coûts qu'elle engendre et à l'absence de modèle de maturité,

comparativement à la fonction de développement, pour articuler et contrôler l'exécution de ses différentes activités.

Or, l'élaboration d'un outil d'évaluation et d'amélioration de la fonction de maintenance du logiciel peut contribuer à accroître la qualité et la productivité des logiciels en opération. Ceci aura pour effet d'assainir son fonctionnement et de réduire ses coûts. Et dans une optique beaucoup plus large, les technologies de l'information dans l'organisation verront leur rôle et leur mission plus renforcés.

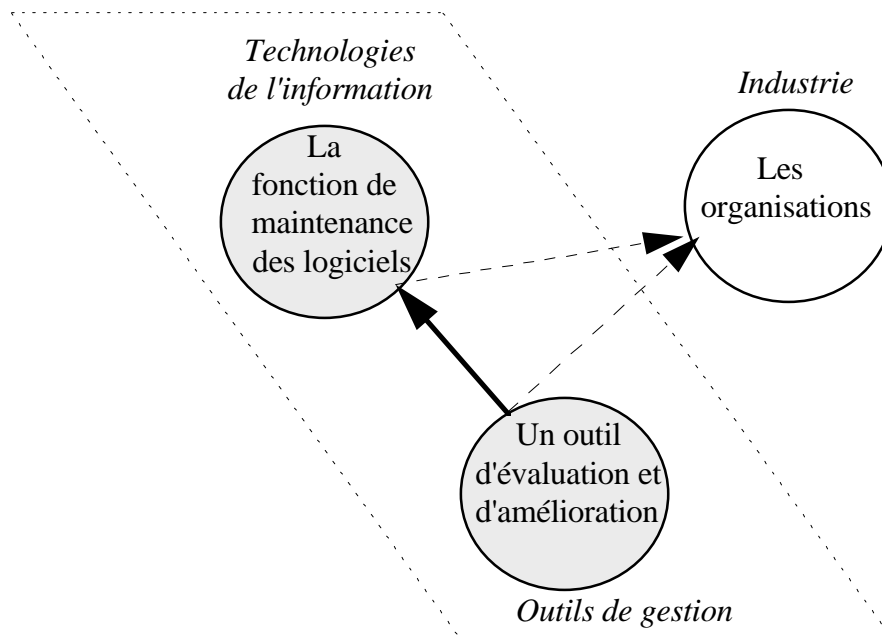


Figure 1.1 - Espace de la problématique

Comme l'illustre la figure 1.1 consacrée à l'espace de la problématique, la fonction de maintenance du logiciel et l'outil d'amélioration de cette fonction seront traités dans cette étude. Les différents impacts produits sur les résultats de l'organisation ne seront ni traités ni vérifiés dans cette étude.

1.1.1 La fonction de maintenance du logiciel dans l'organisation

L'analyse de la littérature traitant de la fonction de maintenance a permis d'identifier trois indicateurs-clés qui démontrent l'importance de la fonction de maintenance du logiciel aux yeux des gestionnaires et des dirigeants des organisations : le volume des budgets qui lui sont alloués, le volume des portefeuilles de logiciels en opération et le volume du personnel mis à contribution pour prendre en charge, réaliser et gérer les tâches de la fonction de maintenance. Voici quelques précisions sur ces trois indicateurs :

– *les budgets de la maintenance* : les recherches ont confirmé qu'en moyenne, certaines organisations affectent à la maintenance de leurs logiciels entre 50% et 70% des budgets d'opération de l'informatique [Arthur, 1988; Port, 1988; Swanson & Beath, 1989; Sharpe et al., 1991]. Les coûts de maintenance du logiciel sont en constante croissance [Boehm, 1987; Arthur, 1988]. Selon Boehm (1987), les dépenses totales en matière de logiciel à travers le monde s'élevaient en 1980 à 80 milliards de dollars américains et ces chiffres avaient grimpé à 140 milliards de dollars pour l'année 1985. Avec un taux de croissance de 12%, ce chiffre avoisinerait les 450 milliards de dollars pour 1995. Une amélioration de la productivité du logiciel représenterait donc des économies substantielles.

– *le portefeuille de logiciels à maintenir* : une proportion importante des logiciels sont dans les mains de la maintenance, comparativement aux nouveaux développements. On estime à environ 75 milliards le nombre de lignes de code en usage à travers le monde [Harrison, 1987]. Alors qu'il semble évident que des programmes structurés sont plus pratiques pour leur maintenance, on estimait qu'environ 75% des applications existantes ne sont pas structurées [Schneidewind, 1987]. De plus, le résultat de modifications et d'améliorations fréquentes portées aux applications rend la maintenance de leur structure initiale difficile et coûteuse.

– *le personnel dédié à la maintenance* : l'activité de maintenance est réalisée par un nombre de plus en plus croissant de personnes dans l'organisation. Quels progrès avons-nous réalisés depuis 1954, demandait Boehm (1986) ? Les estimations du nombre de programmeurs dans le monde étaient de

3.25 millions en 1984. Avec un taux de croissance annuel de 7%, cela donnera une population de programmeurs de 10 millions d'ici l'an 2000. (Il n'y a pas de données actualisées pour 1996).

1.1.2 Problèmes liés à la maintenance des logiciels

Dans l'ouvrage de Swanson & Beath (1989) traitant de problèmes rencontrés dans 12 organisations, un nombre important de ces problèmes sont directement liés à la maintenance du logiciel :

- l'absence de modèle de maturité de la maintenance, c'est-à-dire l'absence d'un ensemble fondamental de concepts et de principes qui gouvernent la fonction de maintenance;
- la planification des tâches de maintenance est souvent inadéquate;
- les difficultés à définir les besoins en formation;
- l'absence ou la faible vérification systématique et périodique des logiciels maintenus;
- l'approche de l'organisation de la maintenance se limite à la nature de la tâche seulement, c'est-à-dire excluant les aspects organisationnels;
- le faible niveau de recherche couvrant le domaine de la maintenance;
- l'absence d'un cadre formel de contrôle de processus;
- la faible connaissance des logiciels, de la programmation et de l'environnement de traitement de l'information;
- l'absence de reconnaissance de la fonction maintenance et de ses réalisations.

Selon Swanson & Beath (1989), ces problèmes font souvent l'objet d'un traitement de crise par des gestionnaires qui se voient contraints de générer une panoplie de solutions *ad hoc* dans le but de réorganiser la fonction de maintenance. Ce traitement de crise limite l'ambition des gestionnaires de contrôler la fonction de maintenance et les laisse inadéquatement préparés face à de multiples problèmes dont :

- Comment identifier les besoins en maintenance ?
- Comment planifier la maintenance ?

- Comment organiser le personnel de la maintenance ?
- De quelle information a-t-on besoin pour gérer la maintenance ?
- De quels outils ont besoin les responsables de la maintenance ?
- Comment motiver les responsables de la maintenance ?
- Quelle est la nature et l'environnement de la tâche de maintenance ?
- Jusqu'où s'étend l'environnement de la maintenance ?
- Comment anticiper les événements ?

Selon la littérature consultée, il n'existe pas de cadre formel de contrôle de processus regroupant l'ensemble des activités de maintenance. Selon Swanson & Beath (1989), la recherche d'un cadre approprié est sans cesse remise en cause par l'attente d'arrivée sur le marché de nouvelles technologies de développement en mesure de résoudre les problèmes générés par l'obligation de maintenir les logiciels.

D'autre part, la complexité organisationnelle de la maintenance vient du fait qu'elle ne se limite pas à son seul aspect technique. La maintenance est plus qu'une tâche individuelle, elle est aussi organisationnelle, car elle devrait incorporer aussi les utilisateurs et le personnel informatique [Swanson et Beath, 1989].

L'interdépendance des logiciels revêt aussi un caractère important pour l'organisation. Les logiciels se développent et deviennent dépendants les uns des autres. Leur regroupement est plus profitable pour une organisation, car elle est mieux servie par des logiciels groupés que par des logiciels pris séparément [Swanson et Beath, 1989]. Cette nouvelle approche qui consiste à aborder la gestion de la maintenance par portefeuille de logiciels s'ajoute aux critères de diversité et de complexité de la fonction de maintenance déjà mentionnés.

Ce large éventail de problématiques liées à la complexité de la fonction, à la diversité des problèmes et à l'interdépendance des logiciels touche l'ensemble des activités de maintenance,

c'est-à-dire son processus. Or, il n'existe, ni dans le domaine de la recherche ni sur le marché actuel, d'instrument pour diagnostiquer l'état de la maintenance dans une organisation donnée ni pour indiquer quelles seraient les recommandations bénéfiques pour améliorer ce processus.

1.1.3 Absence d'outil d'évaluation et d'amélioration de la maintenance des logiciels

Dans le domaine du développement, de tels cadres de gestion existent bien et selon la littérature consultée, la fonction "développement" ou "fabrication de logiciel" reçoit plus d'attention que la fonction "maintenance" [Swanson et Beath, 1989]. L'hypothèse implicite retenue est que la maintenance peut et devrait être gérée "plus ou moins" de la même manière que le développement [Cash et al., 1988].

L'élaboration du modèle de maturité CMM (*Capability Maturity Model*) par le SEI (*Software Engineering Institute*) de l'Université de Pittsburgh (Pennsylvanie, États-Unis) pour l'évaluation du processus de fabrication de logiciels qualifie l'ensemble des activités de développement de "processus de développement" et considère que la qualité d'un système informatisé est tributaire de la qualité du processus utilisé pour son développement [Paulk, 1993].

Or, étant constitué de l'ensemble des activités de maintenance, le processus de maintenance peut-il, lui aussi, se prêter à des opérations d'analyse, d'évaluation et d'amélioration à partir d'un instrument conçu, élaboré et adapté du modèle de maturité du SEI ? C'est l'élaboration d'un tel modèle qui fait l'objet de la présente étude.

1.2 Importance du projet

Le projet revêt de l'intérêt pour plusieurs types d'intervenants : pour l'entreprise en termes de qualité, productivité et gains financiers; pour les chercheurs en termes de contribution à l'ouverture de nouvelles pistes de recherche; pour l'auteur de la présente étude en termes d'apprentissage et d'opportunités intellectuelles et professionnelles.

1.2.1 Intérêts pour l'entreprise

Les entreprises disposeront d'un outil de gestion en mesure d'évaluer et d'améliorer spécifiquement et exclusivement la fonction de maintenance des logiciels. Elles veilleront avec plus de visibilité sur la productivité et la qualité de leurs portefeuilles de logiciels. En effet, dans le secteur des logiciels, le potentiel pour créer des liens stratégiques existe dans de nombreux portefeuilles de logiciels. Nolan (1979) précise que la tâche globale de développement et de maintenance du logiciel est à la base de l'élaboration et du déploiement du portefeuille d'applications pour le Département informatique et que c'est à travers ce portefeuille d'applications que les services informatiques sont mis à la disposition de l'organisation. D'où l'importance stratégique de la fonction de maintenance de voir vivre et évoluer des portefeuilles d'applications stables, de qualité, contrôlables et rentables.

1.2.2 Intérêts pour les chercheurs

Comme domaine de recherche, exception faite pour des aspects précis comme la qualité, la complexité, la réingénierie, *etc.*, la maintenance des logiciels n'a représenté jusqu'ici que peu d'intérêt et est même considérée comme un secteur relativement négligé [Hale et Haworth, 1988]. La performance réelle de cette activité au sein de l'organisation a été peu étudiée car, comparativement au secteur du développement, il n'existe pas d'outils d'évaluation, peu de techniques de mesure et peu de banques de données historiques propres au processus de maintenance.

1.2.3 Intérêts de l'auteur

La motivation et l'enthousiasme qui m'animent devant un tel domaine de recherche s'appuient sur deux raisons essentielles :

D'abord l'intérêt particulier que j'accorde au domaine de la maintenance est dû à mon propre cheminement professionnel. J'ai été moi-même affecté à des tâches de maintenance avant d'assumer des responsabilités de supervision d'une unité de maintenance où je ne disposais d'aucun cadre de quelque nature que ce soit pour mener à bien mon mandat. Résultats : j'ai administré aux tâches de maintenance un traitement de crise peu efficace et peu performant. J'ai certes éteint les feux les plus critiques en assurant le minimum de temps d'interruption des logiciels et en réalisant les différents changements apportés aux applications mais dans l'ensemble, les résultats n'étaient pas performants. La suite de mon cheminement professionnel s'est orientée plus tard vers l'analyse de logiciels et la gestion de projets mais, je reste convaincu de l'existence de gisements d'amélioration inexplorés dans le domaine de la maintenance. Ce projet me permet d'y contribuer sous un angle plus global et mieux structuré.

Ensuite, c'est mon souhait d'entreprendre des études de doctorat dans le domaine des processus de la maintenance des logiciels. Il est souhaitable d'amorcer une vaste enquête auprès d'un lot important d'organisations pour compléter et ajuster les résultats de cette étude. Buts ? Découvrir comment fonctionne le processus de maintenance dans les entreprises aujourd'hui. Car, en plus de traiter une masse significative d'informations qui permet d'ajuster le modèle, il est utile d'établir un diagnostic actualisé de l'état du processus de la maintenance dans les organisations. Les résultats de cette large enquête réalisée dans un cadre de recherche me permettront de rendre davantage crédible un modèle de la maintenance de logiciel.

Chapitre 2

Contexte théorique

Le but de ce chapitre est de présenter le cadre conceptuel dans lequel s'inscrit ce projet de recherche. Nous aborderons deux aspects : la maintenance du logiciel et un outil d'évaluation et d'amélioration de logiciel. Le premier aspect nous donnera en plus d'une définition et les types de maintenance répertoriés jusqu'à ce jour, les particularités entre développement et maintenance, le positionnement de la maintenance dans le cycle de vie du logiciel ainsi que le processus de maintenance lui-même, c'est-à-dire l'ensemble des activités de maintenance qui le composent. Le deuxième aspect fera un historique du modèle utilisé comme architecture de la présente étude, présentera ses concepts, ses niveaux de maturité ainsi que ses finalités.

2.1 La maintenance du logiciel

2.1.1 Définition et types de maintenance

D'après la norme IEEE Standard 729-1983, la maintenance du logiciel est définie comme l'ensemble des travaux réalisés sur un logiciel après sa mise en opération.

Dans un contexte de comparaison illustré par le tableau 2.1, Zvegintzov (1991) fait la synthèse des différents types de tâches de maintenance provenant de plusieurs enquêtes de spécialistes dans le domaine de la maintenance des logiciels [Lientz & Swanson, 1980; R. Ball, 1987; Dekleva, 1990].

Types	Lientz et Swanson (1980)	Richard K. Ball (1987)	Dekleva (1990)
Corrections	22%	17%	16%
Améliorations	59%	39%	43%
Adaptations	6%	9%	8%
Mise au point	4%	5%	5%
Documentation	6%	5%	6%
Ré-ingénierie	n/a	10%	9%
Support usagers	n/a	12%	12%
Autres	3%	3%	1%

(a)

Types	Lientz et Swanson (1980)	Richard K. Ball (1987)	Dekleva (1990)	Alain Abran (1990)
Corrective	22%	17%	16%	21%
Adaptative	59%	39%	43%	60%
Perfective	16%	29%	28%	3%
Support aux usagers	3%	15%	13%	15%

(b)

Source : Abran et Nguyenkim, 1993

Tableau 2.1 - Effort et types de maintenance (répartition de l'effort en pourcentage)

Les types de maintenance, comme le montre le tableau 2.1a, correspondent à des tâches de correction suite à des erreurs intervenues dans les logiciels, d'amélioration lorsqu'il s'agit d'ajouter de nouvelles «fonctionnalités» aux logiciels, d'adaptation pour répondre à des changements à apporter aux logiciels ou à l'environnement, de mise au point avant la livraison des logiciels, de documentation des logiciels pour leur utilisation et pour leur suivi, *etc.* Mais l'ensemble de ces tâches peuvent être regroupées, comme le montre le tableau 2.1b, en trois sous-catégories principales [Lientz et Swanson, 1980; Martin et McClure, 1983; Arthur, 1988] :

– *la maintenance corrective* : elle consiste à apporter des corrections à un programme dans l'immédiat ou à court terme dans le but de garder les logiciels en opération. Ce type de maintenance est considéré comme réactif;

– *la maintenance adaptative* : elle consiste à apporter des compléments ou des modifications aux fonctionnalités du système pour le rendre conforme aux besoins nouveaux de l'organisation;

– *la maintenance perfective* : elle consiste à apporter des modifications d'ordre technique sans remettre en cause les fonctionnalités du système. Cette amélioration génère souvent une meilleure performance.

Une quatrième activité appelée *support aux usagers*, mise en relief plus récemment [Abran et Nguyenkim, 1991,1993], englobe le support en information aux usagers, comme le montre la répartition de l'effort par catégorie ou type de maintenance du tableau 2.1b. Ce quatrième type de maintenance regroupe plusieurs activités : les demandes de rapports *ad hoc*, le temps consacré à fournir des explications aux usagers sur le fonctionnement des applications et des différents cadres et règles dans lesquels s'exécutent et s'opèrent ces logiciels, les demandes d'études d'impacts en cas de changements envisagés dans l'avenir, *etc.*

Mais l'importance de ce quatrième type d'activité est due au fait qu'elle couvre jusqu'à 25% des efforts de maintenance [Abran et Nguyenkim, 1991,1993].

2.1.2 Particularités entre développement et maintenance

Du côté de l'industrie, la frontière entre les activités de développement et les activités de maintenance est mince. Plusieurs chercheurs considèrent que les travaux d'améliorations majeures sont considérés comme des projets de développement plutôt que comme des activités de maintenance, car en matière de gestion de projets, les caractéristiques de travaux d'améliorations majeures sont similaires aux caractéristiques de projets de développement [Desharnais, 1988; Johnson, 1991; Abran et Nguyenkim, 1993].

D'autre part, pour les requêtes de maintenance de petite taille, les caractéristiques de complexité, taille, visibilité, contrôle, approbation, approche organisationnelle, *etc.*, ne sont pas communes aux deux types d'activités, comme l'illustre le tableau 2.2.

Caractéristiques	Projet de développement ou de maintenance (amélioration majeure)	Requêtes de maintenance de petite taille (RMPT)
Complexité <i>forte/faible</i>	nécessite une équipe composée d'informaticiens et d'utilisateurs.	Une ou deux personnes peuvent suffire.
Taille <i>grande/faible</i>	nécessite une planification financière, organisationnelle et structurelle.	quelques heures ou quelques jours de réalisation peuvent suffire.
Visibilité <i>importante/moindre</i>	nécessite une visibilité corporative	se limite aux opérations
Contrôle <i>formel/informel</i>	nécessite un gestionnaire de projets, un comité de contrôle.	un gestionnaire de premier niveau suffit
Approbation <i>formelle/informelle</i>	nécessite l'approbation de la haute direction.	la direction informatique ou la direction de l'utilisateur d'où émane la requête suffit.
Approche Org. <i>structurée/non struct.</i>	Gestion de projets	Gestion de files d'attente

Tableau 2.2 - Caractéristiques d'améliorations majeures *versus* activités de maintenance

Or ces requêtes de maintenance de petite taille se distinguent des améliorations majeures apportées aux logiciels et considérées comme des projets de développement, comme le montre le tableau 2.2, peuvent représenter jusqu'à 50% des coûts totaux des budgets informatiques des organisations [Abran et Nguyenkim, 1991, 1993].

2.1.3 Positionnement de la maintenance dans le cycle de vie du logiciel

Dans la norme IEEE, *standard 1074-1991*, le cycle de vie du logiciel est présenté comme une suite d'activités dont l'activité de maintenance. La Figure 2.1 situe l'activité de maintenance dans le cycle de vie du logiciel juste à la fin de la phase d'implantation et de contrôle du logiciel.

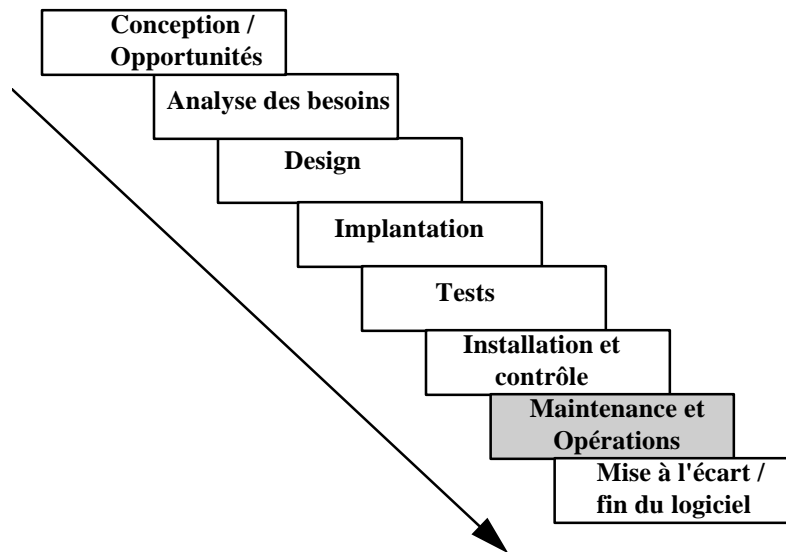


Figure 2.1 - Cycle de vie du logiciel d'après la norme IEEE

Dans la norme IEEE, standard 1074-1991, l'activité de maintenance du logiciel regroupe une grande partie des phases du cycle de vie du logiciel, c'est-à-dire que sa réalisation peut nécessiter de parcourir partiellement ou totalement la plus grande partie du cycle de vie du logiciel.

2.1.4 Processus de la maintenance

D'une part, la norme IEEE, standard 1219-1992 présente (figure 2.2a) le processus de maintenance comme regroupant sept phases ou activités qui vont de l'identification du problème se présentant sous forme de besoin de correction, de modification ou de complément à apporter à un programme jusqu'à la livraison, c'est-à-dire jusqu'à l'implantation finale de la solution.

D'autre part, la norme IEEE, standard 1219-1992 présente (figure 2.2b) huit instruments de gestion de la maintenance ou domaines impliqués dans la gestion de la maintenance du logiciel. Jusqu'à présent, il existe un bon nombre de règles, procédures et pratiques de gestion qui aident à gérer un ou plusieurs aspects à la fois de la maintenance du logiciel mais la littérature n'identifie, à

ce jour, aucun outil d'évaluation et d'amélioration du processus de maintenance dans son ensemble.

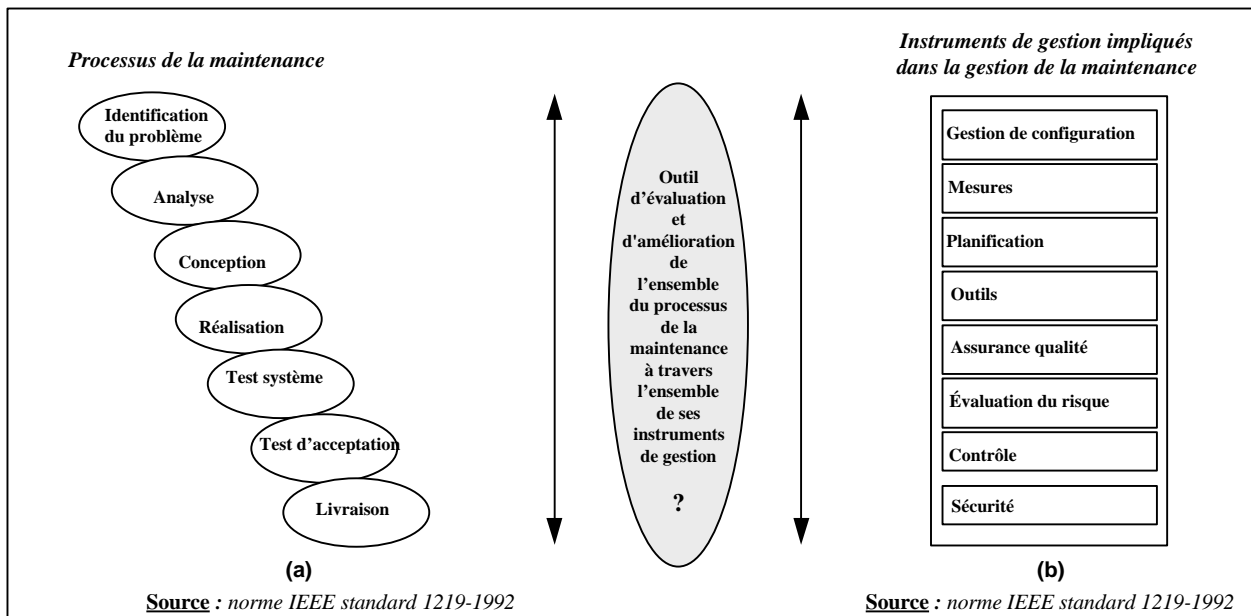


Figure 2.2 - Processus de maintenance et instruments de gestion de la maintenance

2.2 Évaluation et amélioration de processus : le modèle de maturité CMM du SEI

La deuxième partie des fondements conceptuels est basée sur le modèle de maturité CMM (*Capability Maturity Model*) du SEI (*Software Engineering Institute*) de l'Université de Carnegie Mellon de Pittsburgh (Pennsylvanie, États-Unis)

La description de ce modèle de maturité, son historique, ses concepts, sa structure et ses finalités, sont largement influencés par les articles de Watts. S. Humphrey et Paulk et al. [Humphrey, 1988; Paulk et al., 1993a].

2.2.1 Historique du modèle

La *crise du logiciel* des années 80 et l'absence de résultats substantiels en matière de productivité et de qualité des logiciels ont poussé les praticiens à se demander si le problème fondamental ne provenait pas plutôt de l'inhabileté à gérer le processus à travers lequel les logiciels se développent [Paulk, 1993]. La qualité d'un logiciel est tributaire de la qualité du processus utilisé pour son développement [Paulk, 1993].

Reconnaissant le besoin d'améliorer le processus de développement des logiciels conçus pour des applications militaires, le Département Américain de la Défense (DoD-*Department of Defense*) démarre au début des années 80 une série de programmes d'amélioration de la fiabilité, maintenabilité et testabilité de ses logiciels. En 1984, avec les fonds de l'administration militaire américaine, est mis sur pied le SEI (*Software Engineering Institute*) au sein de l'université Carnegie Mellon de Pittsburgh, (Pennsylvanie, États-Unis). Sa mission essentielle est d'élaborer des standards d'excellence en génie logiciel et d'accélérer le transfert des méthodes et techniques avancées en génie logiciel vers la pratique, c'est-à-dire vers l'industrie.

Ce n'est qu'en 1986, avec l'assistance de la Mitre Corporation (entreprise para publique fédérale), que le SEI démarre l'élaboration d'un cadre conceptuel (appelé cadre de référence ou "*Framework*") portant sur la maturité du processus de développement des logiciels pour aider les praticiens à améliorer leur processus de développement [Humphrey, 1988]. Plus tard, le SEI fit évoluer ce cadre de référence vers un modèle de maturité de processus plus connu aujourd'hui sous l'appellation CMM ou *Capability Maturity Model*.

Ce modèle contient des recommandations directement liées à des dimensions clés du processus de développement de logiciels qui ont démontré leur capacité d'amélioration du processus de développement des logiciels d'une organisation [Paulk, 1993].

2.2.2 Concepts du modèle

Le modèle CMM du SEI repose sur trois concepts : performance, aptitude et maturité. Ces concepts caractérisent le processus de développement des logiciels dans les organisations. Le concept de performance du processus se reconnaît dans les résultats "effectivement atteints" par le processus de développement. Le concept d'aptitude du processus décrit les résultats "potentiellement attendus" qui peuvent être réalisés en appliquant le processus. Le concept de maturité implique un potentiel de croissance du processus, c'est-à-dire d'amélioration de l'aptitude.

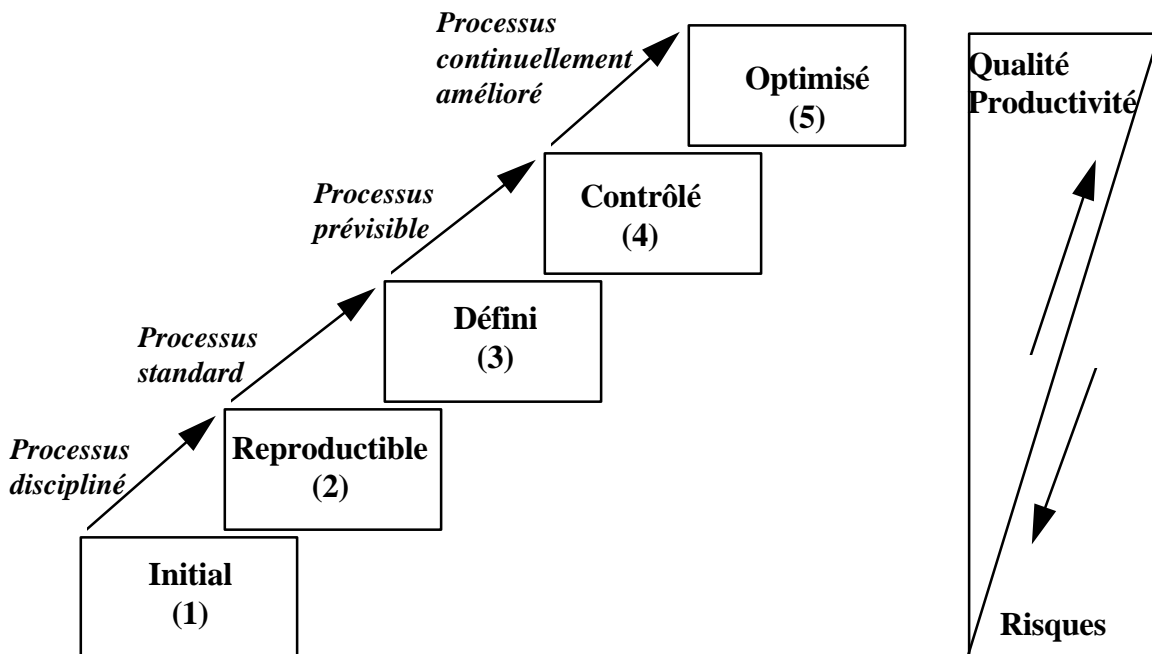


Figure 2.3 - Cadre conceptuel de maturité de processus (*Framework* de Watts Humphrey)

La Figure 2.3 illustre la maturité du processus de développement sur une échelle de 1 à 5 où chaque niveau est considéré comme le degré pour lequel le processus se qualifie. On dit que le niveau 1 est le niveau initial, le niveau 2 est le niveau reproductible, le niveau 3 est le niveau défini, le niveau 4 est le niveau contrôlé et le niveau 5 est le niveau optimisé.

Le modèle de maturité du SEI est l'aboutissement de l'effort pour formaliser toutes les activités qui permettent de caractériser, d'analyser et d'améliorer le processus de production du logiciel [Fugetta, 1993]. Ce modèle organise ces activités en cinq niveaux de maturité. Il sert donc de cadre conceptuel pour classer les processus de développement selon leur degré d'évolution et suggère des améliorations appropriées aux problèmes qui caractérisent chaque niveau de maturité.

En effet, dans une organisation immature, le processus de développement d'un projet est improvisé par les gestionnaires et les «développeurs». Beaucoup d'effort sont consacrés pour réagir aux situations de crise, ce qui, inévitablement influence trois éléments : les coûts, les délais et la qualité. Le dernier élément est souvent sacrifié pour ne pas aggraver les deux premiers. A ce premier niveau, les risques sont très élevés et la productivité est très faible. Par contre, une organisation mature a un processus de développement bien défini et renforcé. Il est rigoureusement respecté et sa mise en oeuvre est facilitée par une infrastructure de support. La qualité du processus et des produits est évaluée à l'aide de moyens objectifs, les problèmes potentiels sont identifiés et résolus à l'avance par des méthodes quantitatives, ce qui rend les coûts et les délais beaucoup plus réalistes et réalisables dans les limites du prévisible. Dans ce cas, la qualité des produits est atteinte, les risques sont faibles et la productivité est élevée.

2.2.3 Niveaux de maturité du modèle

Comme le montre la figure 2.3 dans laquelle est présenté le cadre conceptuel de maturité de processus, le CMM du SEI se compose de 5 étages appelés aussi niveaux de maturité :

– *Niveau 1* ou “*niveau initial*” : ce niveau est à la base des niveaux. Il est caractérisé par une approche *ad hoc* de développement. Dans ce cas, même si des procédures formelles existent, elles ne sont pas respectées, et le sont encore moins en situation de crise. A cause du manque de structure et de contrôle adéquats, les résultats de projets similaires peuvent varier largement en termes de productivité

et de qualité. L'aptitude du processus de niveau initial est une caractéristique engendrée par les individus de l'organisation et non pas par des mécanismes de l'organisation.

– *Niveau 2* ou “*niveau reproductible*” : c'est à ce niveau que les individus acquièrent une expérience qui leur permet de réaliser des projets similaires avec la même performance. L'aptitude du processus reproductible peut être qualifiée de disciplinée. Mais le processus, à ce stade, manque encore de visibilité puisqu'il dépend toujours des individus. Par conséquent, des risques peuvent survenir lorsque des défis se présentent à l'organisation comme l'introduction de nouveaux outils ou de nouvelles méthodes de travail, le développement d'un nouveau type de produit ou un changement organisationnel important. Devant de telles difficultés, la performance est sujette à dégradation.

– *Niveau 3* ou “*niveau défini*” : c'est à ce niveau que le processus de développement est structuré et visible. Le processus “défini” fournit à l'organisation des bases pour procéder à des améliorations plus importantes et continues. On peut, en effet, commencer à mesurer certaines tâches de façon efficace. A ce niveau de maturité, l'aptitude du processus est considérée comme standard et consistante.

– *Niveau 4* ou “*niveau contrôlé*” : c'est à ce niveau que le processus est instrumenté et qu'il dispose de mesures de productivité et de qualité bien définies et consistantes. Les variations de performance expliquées peuvent être distinguées des variations aléatoires et les circonstances exceptionnelles qui peuvent survenir sont identifiées et gérées. L'aptitude du processus au niveau “contrôlé” peut être jugée comme quantifiable et prévisible. Néanmoins, la mauvaise utilisation des données des mesures et le coût élevé de leur cueillette peuvent s'avérer dommageables.

– *Niveau 5* ou “*niveau optimisé*” : c'est à ce niveau que les données sont disponibles pour analyser et optimiser le processus par lui-même. L'organisation dispose de moyens pour identifier les faiblesses du processus et les corriger de façon proactive par l'adoption de nouvelles innovations en génie logiciel. L'aptitude du processus optimisant peut être décrite par l'amélioration continue.

En fait, trois types d'améliorations sont observés au fur et à mesure que le processus évolue dans l'échelle de maturité. D'abord la différence entre les objectifs fixés et ceux réalisés diminue. Ensuite la

variabilité des résultats obtenus par rapport à ceux prévus diminue. Et finalement les coûts et les délais diminuent alors que la qualité augmente.

Niveaux de maturité	Caractéristiques des niveaux	Secteurs-clés du modèle
1. Initial	<i>Ad hoc</i>	aucun
2. Répétable	Dépend des individus	1. Gestion de la configuration 2. Assurance de la qualité 3. Gestion des fournisseurs 4. Suivi & surveillance projets 5. Planification de projets 6. Gestion des exigences
3. Défini	Défini, institutionnalisé	7. Revues/révisions structurées 8. Coordination inter-groupes 9. Ingénierie des produits 10. Gestion intégrée du logiciel 11. Formation 12. Def. processus organisationnel 13. Res. processus organisationnel
4. Géré	Mesuré (quantitatif)	14. Gestion de la qualité 15. Gestion & contrôle processus
5. Optimisant	Rétroaction d'amélioration du processus	16. Gestion du changement 17. Innovations technologiques 18. Prévention des défauts

Tableau 2.3 - Les secteurs-clés par niveau de maturité du modèle de maturité du SEI

Comme le montre le tableau 2.3, chaque niveau de maturité répond à des caractéristiques spécifiques et couvre des secteurs-clés qui lui sont propres. Pour le modèle de maturité du SEI, un secteur-clé ne peut se situer à deux niveaux de maturité différents.

2.2.4 Finalités du modèle

Pour les organisations, l'apport de l'utilisation du modèle de maturité du SEI est triple :

1) Le modèle permet de situer la maturité du processus de développement d'une organisation, c'est-à-dire situer son niveau de performance. L'analyse des résultats obtenus par l'évaluation permet de fixer des priorités et d'élaborer des plans d'action pour atteindre des objectifs visés par l'organisation. L'analyse des résultats informe aussi sur les forces et les faiblesses du processus. Ces informations peuvent servir de support à la prise de décision par les gestionnaires en charge du processus ou par les dirigeants de l'organisation.

2) Le modèle permet d'identifier les différentes tâches à réaliser pour améliorer ce processus d'une façon générale. Cela peut servir de support pour mettre sur pied un programme d'amélioration de processus et peut donner des informations pertinentes de différentes natures comme le volume des besoins et des tâches à entreprendre pour réaliser les améliorations nécessaires.

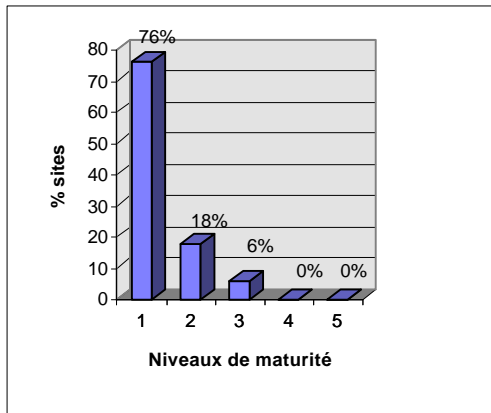
3) Le modèle permet d'amorcer une réflexion générale sur la façon de penser et de faire dans la tâche de chacun. Cette troisième finalité, indirecte et à caractère d'accompagnement, peut être offerte par l'implantation du modèle dans l'organisation. Elle peut servir alors de base et de support à un mouvement d'amélioration continue dans un cadre plus global de changement organisationnel.

Au sein des organisations, les résultats obtenus en utilisant le modèle de maturité du SEI comme support de programmes d'amélioration de processus sont révélateurs, comme le montre le tableau 2.4a, où une enquête menée auprès de 136 sites dévoile que 76% d'organisations ont leur processus de développement au niveau de maturité 1 du modèle, c'est-à-dire à au niveau le moins performant du modèle alors que seulement 18% se situent au niveau 2.

C'est le cas également de la compagnie *Lokheed Missiles & Space Co* dans une enquête portant sur 20 projets, illustrée par le tableau 2.4b, où les paramètres de coûts (millions de dollars), de temps (mois), de qualité (densité des erreurs en nombre de défauts par ligne de code), de productivité (mois/homme en nombre de lignes de code produites) des projets de développement présentent une incidence directement liée aux niveaux de maturité. Les résultats de chaque paramètre entre les niveaux 1 et 5 de maturité sont significatifs.

En matière de coûts globaux de développement dans les organisations, les économies réalisées sont importantes : le cas de la division de génie logiciel de la *Hugues Aircraft* qui a réalisé des économies annuelles de l'ordre de \$ 2 millions après avoir implanté le modèle de maturité CMM du SEI. [Humphrey, 1991].

(enquête auprès de 136 organisations)



(a)

Source : SEI - Carnegie Mellon University, 1990

(enquête sur 20 projets d'une seule organisation)

Niveaux (SEI)	Coûts (M\$)	Temps (Mois)	Qualité (Err/ksLOC)	Productivité (Mh/ksLOC)
5	1	16	.1	2
4	3	19	.3	6
3	7	25	1.0	14
2	15	32	3.0	30
1	33	40	9.0	66

(b)

Source : M.Pore, Lockheed Missiles & space Co, 1990

Tableau 2.4 - Résultats du modèle de maturité du SEI auprès des organisations

Chapitre 3

Méthode d'élaboration du modèle CMM du SEI

Le chapitre 3 traite de l'élaboration du modèle de maturité CMM du SEI sur lequel s'est appuyée la réalisation de la présente étude. Ce chapitre présente les fondements méthodologiques qui ont servi de base à la construction du modèle, la structure interne du modèle ainsi que ses contenus global et détaillé.

3.1 Les fondements méthodologiques

Les fondements méthodologiques pour l'élaboration de l'outil d'évaluation et d'amélioration du processus de maintenance proposés par la présente étude s'inspirent largement du modèle de maturité de processus CMM du SEI

Comme l'illustre la figure 3.1, le modèle se décompose en deux parties : une structure globale et un contenu global. La structure globale est considérée comme l'architecture même du modèle alors que la structure détaillée décrit les définitions, règles et caractéristiques des composantes du modèle ainsi que la nature des relations entre elles. Le contenu global décrit la relation hiérarchique entre les composantes du modèle alors que le contenu détaillé regroupe toutes les recommandations sous forme de pratiques-clés ou d'activités à réaliser pour atteindre des objectifs visés par les secteurs-clés.

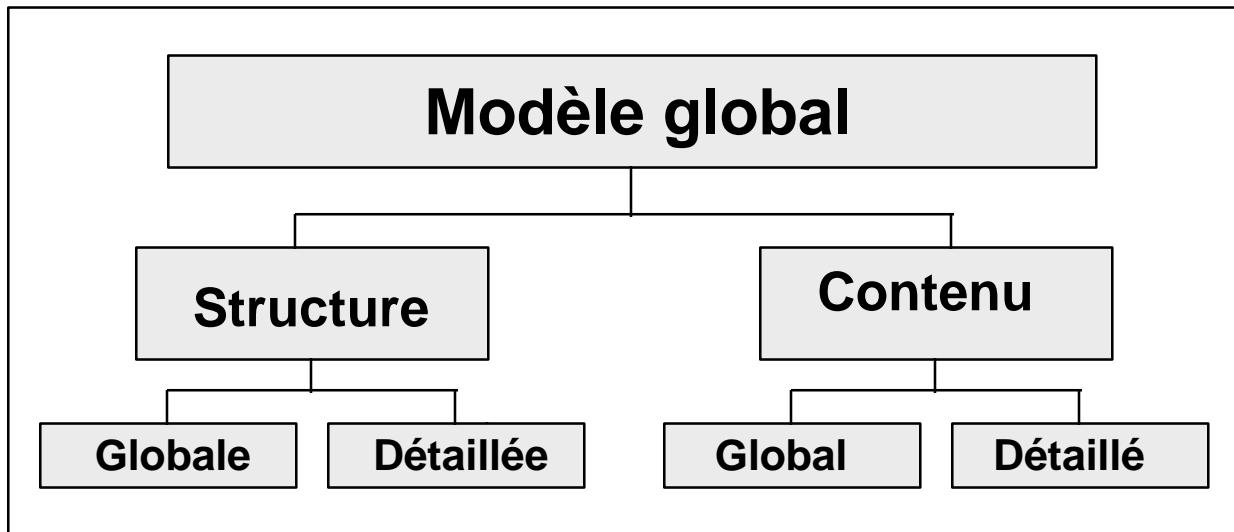


Figure 3.1 - Représentation interne globale du modèle de maturité du SEI

L'élaboration du modèle de maturité du SEI, illustrée par la figure 3.2, est basée sur une approche Delphi et sur un cadre de référence ou "*Framework*" de Watts Humphrey (1988) qui traite de la maturité du processus de développement. Le modèle s'est construit à partir d'opinions d'experts, de bases empiriques collectées auprès du secteur de l'industrie, d'opinions d'évaluateurs et de concepts idéalisés comme étant les meilleures façons de procéder en matière de développement de logiciels.

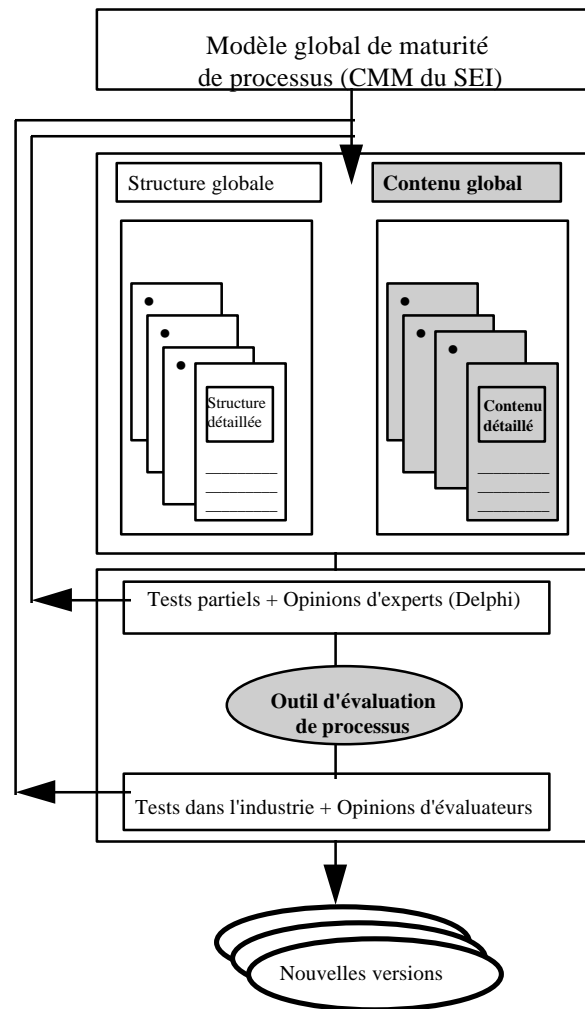


Figure 3.2 - Cadre méthodologique d'élaboration du modèle de maturité du SEI

Dans le cadre de cette adaptation de modèle, choisie comme approche pour réaliser la présente étude, le modèle global CMM du SEI ainsi que ses structures globale et détaillée seront retenues et réutilisées telles quelles. Seules les zones en gris de la figure 3.2, qui illustrent les contenus global et détaillé du modèle ainsi que l'outil d'évaluation et d'amélioration de processus, feront l'objet d'un traitement dans cette étude afin que le contenu et les résultats de l'étude soient spécifiques au domaine de la maintenance des logiciels.

3.1.1 Structure du modèle de maturité CMM du SEI

La structure du CMM est décrite dans un document interne du SEI [Paulk et al., 1991], représenté par la figure 3.3.

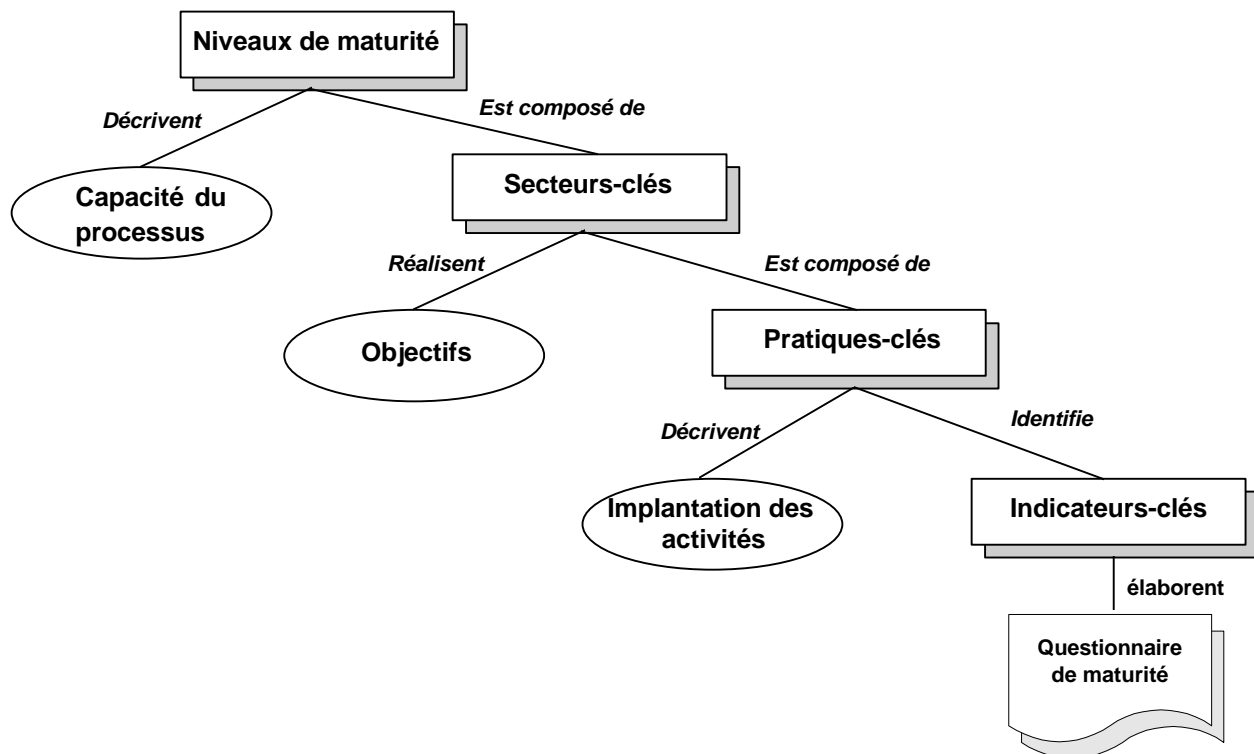


Figure 3.3 - Structure globale du modèle de maturité du SEI

Rappelons qu'à l'exception du niveau 1 qui constitue la base du modèle, chaque niveau de maturité est composé de secteurs-clés et que chaque secteur-clé vise un objectif à atteindre et est constitué de pratiques-clés. Ces pratiques-clés sont considérées comme des activités à réaliser, des procédures, règles et outils à mettre en oeuvre pour tendre vers l'objectif assigné par le secteur-clé. De ces pratiques-clés ainsi identifiées et décrites sont choisis et extraits les indicateurs-clés qui constituent la

base des questions qui composeront le questionnaire de maturité. Ce questionnaire de maturité est utilisé comme support d'évaluation du processus auprès des organisations.

3.1.2 Contenus du modèle de maturité CMM du SEI

Rappelons que le contenu global du modèle limite sa dimension à 18 secteurs-clés (voir tableau 2.3) et qu'il existe des liens d'appartenance entre ses différentes composantes, c'est-à-dire des relations de "composant-composé" entre les pratiques-clés, les secteurs-clés et les niveaux de maturité.

Exemple : le secteur-clé de la planification de projets appartient au niveau 2 de maturité du modèle alors que la gestion de la qualité appartient au niveau 4 de maturité du modèle.

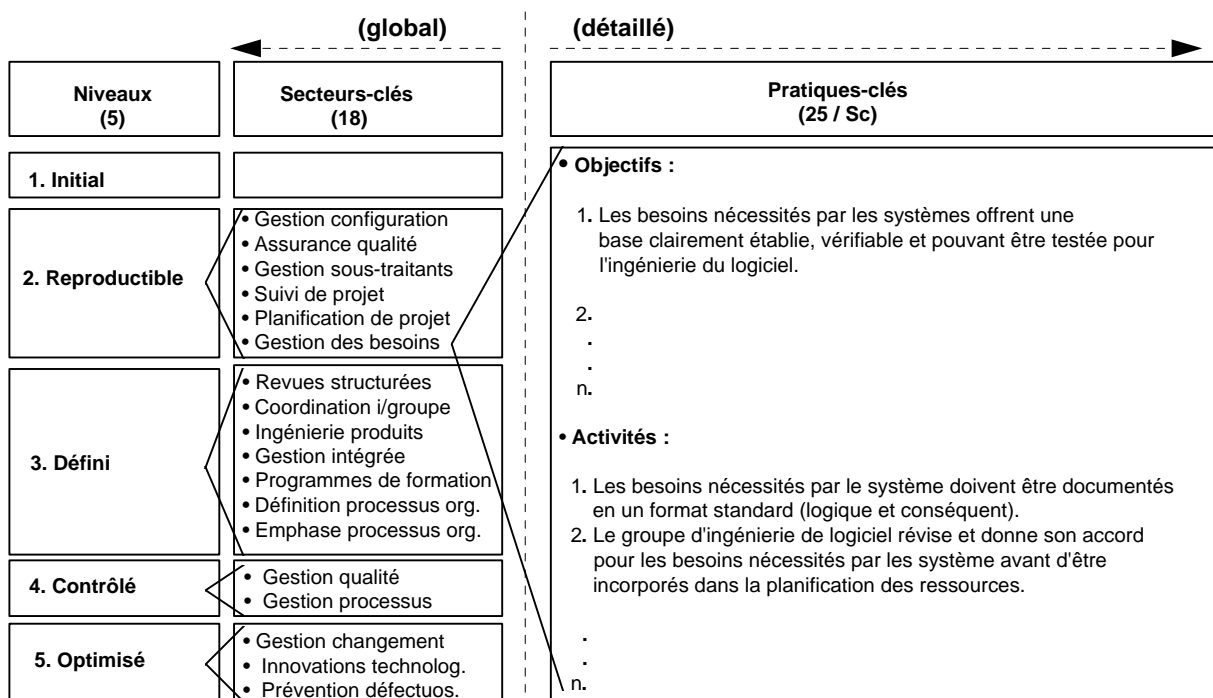


Figure 3.4 - Contenus global et détaillé du modèle de maturité CMM du SEI

Les secteurs-clés sont considérés comme des champs d'intervention où s'exécutent des pratiques-clés ou activités dont l'objectif est d'optimiser le secteur-clé auquel elles appartiennent.

Exemple : toutes les activités réalisées en planification de projet ont pour objectif d'optimiser le secteur de la planification de projet.

Une fois que l'ensemble des pratiques-clés d'un secteur-clé est réalisé, les objectifs du secteur-clé sont considérés comme atteints. Et c'est lorsque l'ensemble des objectifs de tous les secteurs-clés d'un niveau de maturité est atteint que le passage à un niveau de maturité supérieur se réalise.

La figure 3.4 illustre les liens d'appartenance entre les composantes du modèle.

Chapitre 4

La démarche d'intervention

Le chapitre 4 présente l'approche exploratoire préconisée pour réaliser l'étude. Cette approche a permis, pour élaborer le modèle de maturité de la maintenance logiciel, d'identifier et de décrire les différentes composantes du modèle : secteurs-clés, objectifs des secteurs-clés, pratiques-clés, *etc.*.. spécifiques au processus de la maintenance des logiciels.

4.1 Pourquoi le modèle CMM du SEI ne s'applique pas à la maintenance logiciel

Les activités de maintenance logiciel sont globalement structurées autour de la notion de requête de maintenance ou demande de services [Ball, 1990]. La figure 4.1 illustre le lien entre l'origine de la requête de maintenance et la fonction de maintenance telle que défini dans le chapitre 2. Les requêtes peuvent provenir des utilisateurs, du développement, des opérations, ou suite à des recommandations d'une étude de réingénierie, *etc.*

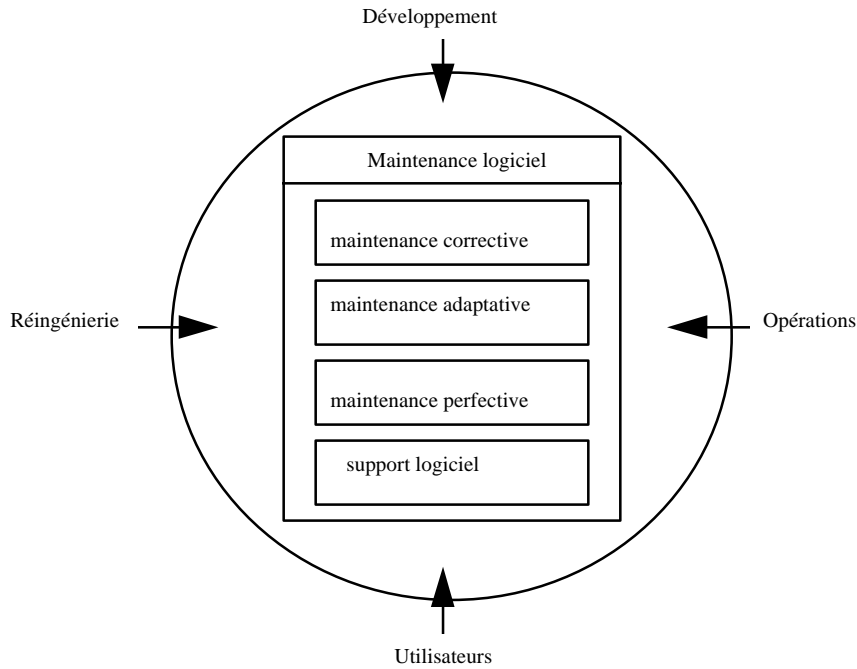


Figure 4.1 - Origine des requêtes de maintenance et la fonction de maintenance

Dans la réception de la requête de maintenance, le passage du stade requête au stade réalisation génère des cadres de transition, processus, règles et procédures à appliquer, activités à réaliser, *etc.*, pour prendre en charge la requête et la réaliser. Le tableau 4.1 illustre, à partir des définitions des catégories de maintenance logiciel, l'origine des requêtes de maintenance, les activités générées et les catégories de maintenance mises à contribution.

Origine des requêtes	Activités à réaliser	Types de maintenance déclenchée
Développement	Gestion de la transition vers la maintenance	corrective, support
Utilisateurs	Utilisation opérationnelle du logiciel	corrective, adaptative, support
Opérations	Gestion du logiciel (performance, transition,..)	perfective, support
Réingénierie	Analyse du logiciel	projet de maintenance ou développement

Tableau 4.1 - Types de maintenance à réaliser selon l'origine de la requête et l'activité qui l'encadre

A cela s'ajoutent les différentes activités spécifiques de gestion de logiciel à réaliser pendant la durée de vie du logiciel. Ces activités commencent une fois le logiciel sorti de la sphère de développement, c'est-à-dire une fois dans les mains de la production, des opérations et du support.

Le tableau 4.2 indique les secteurs à travers lesquels nombre d'activités se réalisent pour gérer le logiciel [Swanson, 1989; Ball, 1990]. Ces secteurs d'activités ne sont pas présents dans le processus de développement du logiciel [Ball, 1990].

Secteurs d'activités	Gestion du logiciel (maintenance)	Développement du logiciel (création)
Gestion des problèmes	P	A
Acceptation du logiciel	P	A
Gestion de la transition vers la maintenance	P	A
Rôle de l'utilisateur, des opérations et du support	P	A
Planification de la maintenance	P	A
Gestion du personnel de maintenance	P	A

Tableau 4.2 - Secteurs d'activité de gestion du logiciel (P = présent , A = absent)

Si on compare les secteurs d'activités représentés par le tableau 4.2 par rapport aux secteurs du modèle CMM du SEI représentés par le tableau 2.3 (voir chapitre 2), on observe que le modèle CMM du SEI, dans son contenu actuel, axé fortement sur le développement de logiciels et traité sous l'angle de gestion de projets uniquement, ne répond pas *a priori* à des préoccupations importantes de la maintenance du logiciel [Zitouni, 1995].

4.2 Élaboration d'un nouveau modèle de maintenance logiciel

L'approche choisie est une approche basée sur une construction de modèle par itération avec comme base de modèle les concepts et les structures globale et détaillée du modèle de maturité du SEI, comme le montre la figure 4.2.

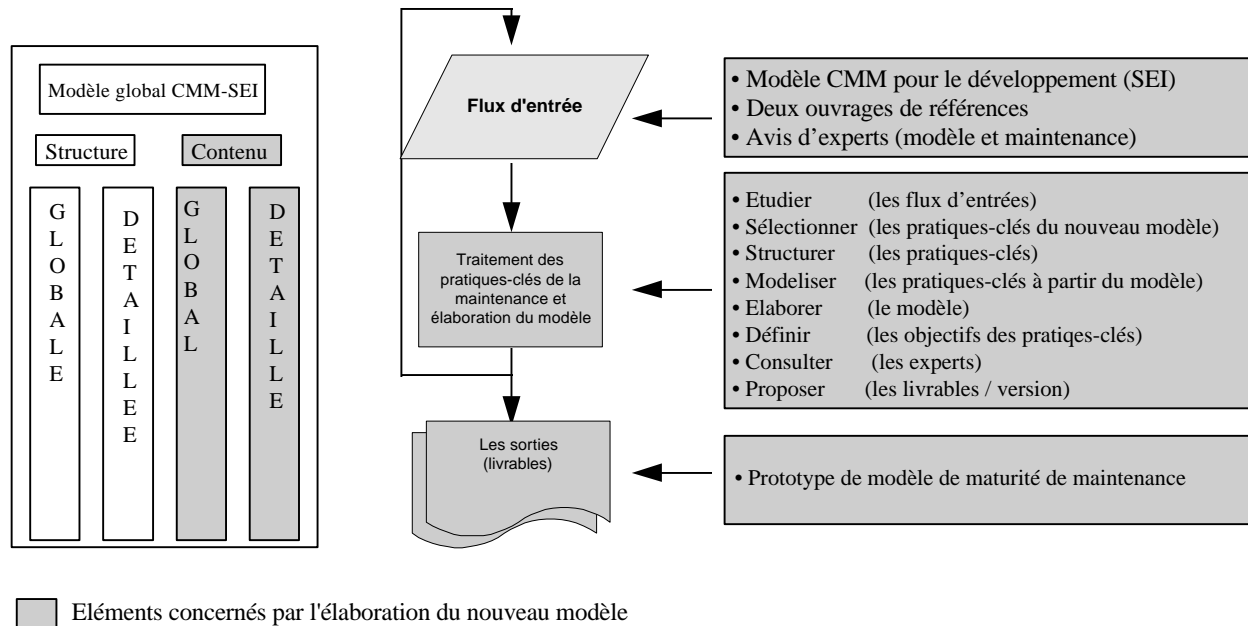


Figure 4.2 - Cadre d'élaboration des livrables du projet de recherche

L'intérêt de la construction de modèle par itération est mis en relief dans un article de Vennix et Gubbels (1992). La notion d'itération est par elle-même un processus d'apprentissage du problème. La plupart des caractéristiques d'un problème mal structuré (dans le sens mal formulé) sont obtenues pendant le processus d'itération de la construction du modèle, bien avant que le modèle soit achevé [Vennix et Gubbels, 1991].

La démarche d'intervention choisie pour la présente étude est basée sur une partie du modèle de maturité du SEI : les concepts du modèle global ainsi que les structures globale et détaillée seront réutilisés. Mais les contenus global et détaillé du modèle représentés par les zones en gris, comme illustré par la figure 4.2, traiteront exclusivement de la maintenance du logiciel et seront considérés comme la base de la réalisation du présent projet.

L'approche retenue pour élaborer le nouveau modèle de maturité adapté à la maintenance du logiciel consiste à identifier les pratiques-clés de maintenance à partir de flux d'entrée.

4.1.1 Flux d'entrée

Les flux d'entrée se présentent sous forme de modèle, d'ouvrages et d'expertise. Dans le cadre de cette étude, il s'agit de trouver des solutions à appliquer aux problèmes de la maintenance. Or beaucoup de solutions se trouvent dans la littérature à partir de laquelle il est possible d'identifier et d'extraire des pratiques-clés qui joueront le rôle de solutions aux problèmes posés par la maintenance des logiciels.

Dans une deuxième étape, il faut adjoindre à ces pratiques-clés un ou plusieurs objectifs à atteindre pour la réalisation d'un secteur-clé donné. Ce secteur-clé peut se trouver parmi les autres secteurs-clés du modèle de maturité du SEI, autrement il restera à être défini et à être positionné dans le nouveau modèle à un niveau de maturité correspondant aux caractéristiques du secteur-clé et du niveau de maturité en question.

Les flux d'entrées subiront des itérations au niveau du traitement pour donner naissance à de nouvelles versions du modèle et du questionnaire.

Les différents supports d'entrée choisis pour cette étude sont :

- *Le modèle de maturité du SEI pour le développement* : traite en réalité du processus de développement de logiciels et non de maintenance de logiciels avec une frontière non délimitée entre développement et maintenance. Ce qui donne la possibilité d'en extraire les éléments pertinents pour la problématique de maintenance.
- *Des ouvrages de référence* : deux ouvrages traitant de maintenance des logiciels dans deux secteurs importants du processus de maintenance dans son ensemble comme la gestion de la maintenance d'une manière générale [Swanson & Beath, 1989] et les pratiques courantes provenant d'expériences vécues en entreprise [Richard K. Ball, 1990]. Bien que relativement anciens, ces ouvrages sont considérés, selon l'avis d'experts comme des références dans le domaine en question.

– *les avis d'experts* : les experts choisis porteront des avis et des conseils sur le choix des solutions à appliquer à la maintenance du logiciel que l'on pourra qualifier de pratiques-clés du modèle. Cette expertise acquise au sein des organisations contribuera à ratisser le maximum de pratiques-clés mises en oeuvre et utilisées dans la gestion des activités de maintenance des logiciels. La cohérence de l'ensemble du modèle et du questionnaire sera soumise aussi à ces mêmes experts.

4.1.2 Traitement

Pour le traitement des flux d'entrée, il faut d'abord identifier les problématiques, inventorier les solutions correspondantes, choisir les composantes, les ordonner, les modéliser, les insérer à la bonne place dans le nouveau modèle, fournir ces composantes aux experts à des fins de contrôle et de validation. Ensuite, il faut définir des objectifs réalisables pour chaque secteur-clé et faire correspondre les pratiques-clés aux secteurs-clés.

4.1.3 Sorties

Les sorties se présentent sous forme d'un livrable obtenu après la fin du traitement des flux d'entrée. C'est un prototype de modèle de maturité de la maintenance du logiciel à l'image du modèle de maturité du SEI dans la forme mais dont le fond ne devra contenir que des composantes spécifiques à la maintenance des logiciels.

4.3 Construction par itération et nouvelles versions du prototype

A des fins d'ajustement et d'amélioration, la construction par itération des prototypes de modèle se fait sous forme de retour aux phases antérieures de leur élaboration. La nécessité des ajustements s'explique par les mises à jour (ajout, modification, suppression) de composantes apportées par les experts dans le but d'améliorer le modèle ou de l'opportunité de nouvelles pratiques-clés de maintenance provenant d'articles, d'études de cas, d'enquêtes, *etc.*, non prises en compte à l'étape initiale de l'élaboration du modèle.

4.4 Rôle des experts

Un groupe d'experts est chargé de donner des conseils et d'enrichir les résultats contenus dans la version préliminaire du modèle élaboré dans le cadre de cette étude. Ces conseils d'experts sont à la base des itérations d'amélioration et d'enrichissement du modèle. La figure 4.3 indique la position qu'occupe l'intervention des experts dans l'élaboration du modèle.

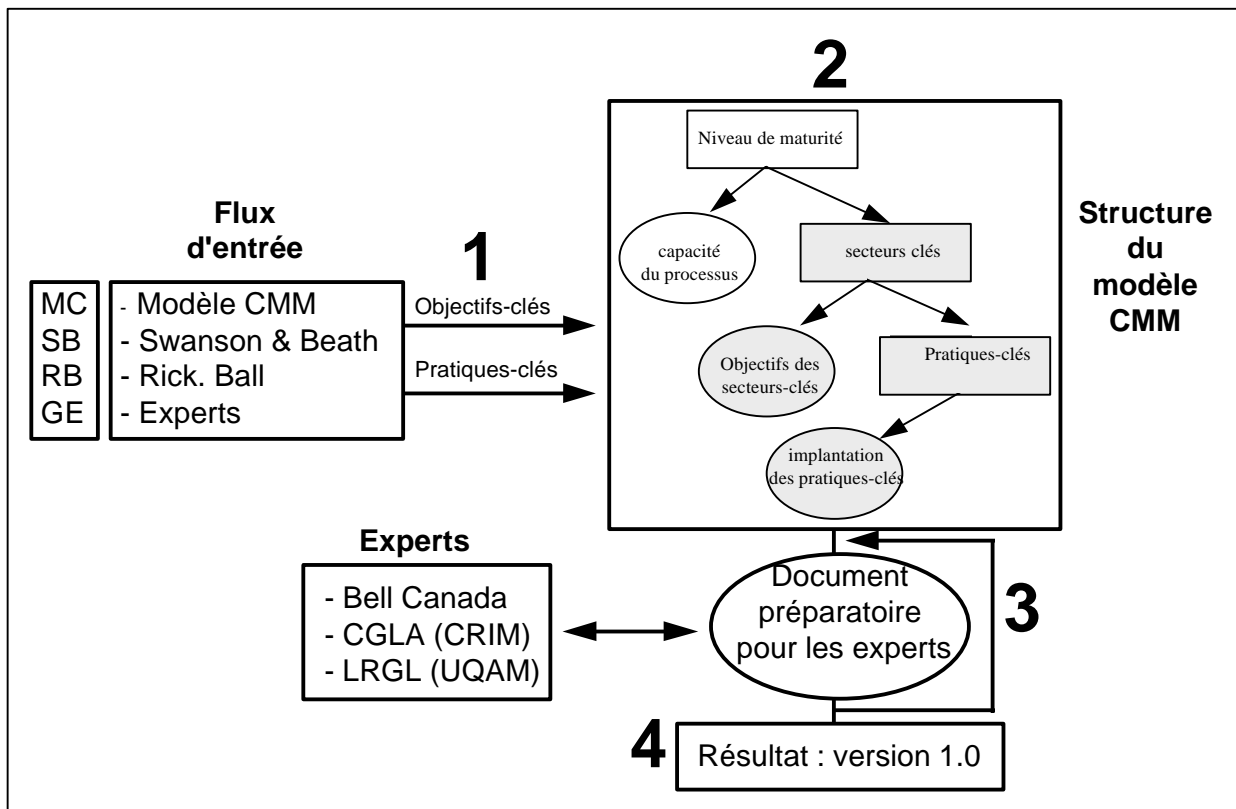


Figure 4.3 - Étapes d'élaboration du modèle de maturité de la maintenance

Les étapes d'élaboration se présentent comme suit :

- *Étape 1* : cette étape consiste à extraire les pratiques-clés et les objectifs-clés contenus dans les flux d'entrée.

- *Étape 2* : cette étape traite les extractions selon la structure interne du modèle CMM. Il s'agit d'étudier et d'ordonner l'extraction selon le niveau de maturité qui la caractérise et selon le secteur-clé auquel elle appartient. Cette étape fournit un document préparatoire de l'ensemble des résultats préliminaires et le confie au groupe d'experts pour étude et avis [cf. détails en annexe A].

- *Étape 3* : cette étape est caractérisée par des itérations de corrections et d'améliorations du contenu du document préparatoire par les experts, comme l'indique la figure 4.3. Le rôle des experts a consisté à parcourir le document préparatoire dans le but de se prononcer sur les six points suivants :
 - 1) Est-ce que les secteurs-clés spécifiques à la maintenance du logiciel décrits dans le document se trouvent bien dans les niveaux de maturité appropriés?

 - 2) Est-ce que les objectifs spécifiques à la maintenance du logiciel décrits dans le document se trouvent bien dans les secteurs-clés appropriés?

 - 3) Est-ce que les pratiques-clés spécifiques à la maintenance du logiciel décrites dans le document se trouvent bien dans les secteurs-clés appropriés?

 - 4) Y a-t-il d'autres objectifs spécifiques à la maintenance du logiciel à rajouter et à relier aux secteurs-clés appropriés?

 - 5) Y a-t-il d'autres pratiques-clés spécifiques à la maintenance du logiciel à rajouter dans les secteurs-clés appropriés?

- 6) Y a-t-il d'autres secteurs-clés spécifiques à la maintenance du logiciel à rajouter dans les niveaux de maturité appropriés?

Dans le contexte présent, l'expert se devait de signaler toute anomalie importante de cohérence ou anomalie de toute autre nature concernant le fond comme la forme. Les suggestions d'enrichissement de ce travail étaient les bienvenues.

Afin de faciliter la tâche de l'expert, une codification des objectifs, pratiques, secteurs et niveaux de maturité lui a permis de porter ses réflexions directement sur le document préparatoire. Dans le cadre de ces itérations et de cette démarche, les experts ont consacré globalement trois demi-journées chacun à des séances de travail qui ont permis de clarifier les différents points de vive voix. L'échéancier a été respecté.

- *Étape 4* : cette étape est considérée comme la fin du processus d'élaboration en fournissant la version 1.0 du contenu du modèle de maturité de la maintenance.

4.5 Envergure de l'étude et limites de la méthode

Dans le cadre de la présente étude, il est nécessaire de préciser et de clarifier un certain nombre de points (sept points) touchant la démarche d'intervention et la méthode utilisée.

1) *La définition du type de maintenance traitée* : la maintenance du logiciel est définie d'après la norme IEEE [ANSI/IEEE standard 729-1983] comme représentant l'ensemble des travaux réalisés sur un logiciel après sa mise en opération. Cette définition étant large, il est souhaitable que dans le cadre de cette activité de synthèse la notion « d'ensemble des travaux réalisés sur un logiciel après sa mise en opération » soit clarifiée.

L'activité de maintenance du logiciel n'a pas un contenu et un statut normalisés. Tant au niveau de la tâche qu'au niveau de l'environnement de la tâche, l'exécution de la maintenance est souvent différente d'une organisation à une autre. Ces différences reposent sur plusieurs aspects dont la nature des tâches de maintenance, le processus d'exécution de la maintenance, l'emplacement de la fonction de maintenance dans l'organigramme structurel d'une organisation, sa mission, ses ressources, *etc.* C'est pourquoi l'envergure de l'étude n'est pas limitée par un type particulier de maintenance à traiter mais par un nombre d'ouvrages identifiés au préalable et retenus comme support de réalisation de la présente étude.

Par conséquent, la réalisation de la présente étude a pris en compte tous les types de maintenance traités dans ces deux supports avec la caractéristique suivante pour en limiter l'envergure: « Il s'agit de toutes les activités de maintenance qui se trouvent dans les mains de programmeurs, d'analystes et de gestionnaires des logiciels d'applications ». Ce qui exclut notamment les équipements, l'installation et l'exploitation des ordinateurs et des réseaux.

2) *Les sources des composantes du modèle* : ces sources sont limitées au modèle de maturité du SEI, à deux ouvrages de référence en matière de maintenance de logiciels et à l'expertise de praticiens et de chercheurs dans le domaine.

3) *Les différents champs concernés* : les types de maintenance concernés par la présente étude ont des caractéristiques d'activités de maintenance et non des caractéristiques de projets de développement. Pour ce qui est du type d'organisation dans lesquelles se déroulent ces activités de maintenance, n'ont été prises en compte que les organisations de taille importante dont les ressources en technologie de l'information sont significatives, c'est-à-dire des organisations qui fonctionnent avec des processus formels et structurés. Cette étude a exclu par conséquent les organisations de type PME (petites et moyennes entreprises).

4) *Les limites des ajustements* : seuls les ajustements des experts choisis dans le cadre de cette étude (chercheurs ou praticiens) ont été retenus.

5) *Les résultats obtenus* : il faut considérer les résultats obtenus par cette étude comme une tentative d'adaptation du modèle de maturité du SEI aux problématiques de la maintenance. Cette étude exploratoire ne présente pas ce modèle de maintenance logiciel comme la meilleure solution aux problèmes posés par la maintenance du logiciel dans les organisations.

6) *L'entreprise exploratoire* : la compagnie Bell Canada dont un expert en maintenance a été sollicité aux différentes étapes de réalisation de la présente étude.

7) *Les limites de la contribution des experts* : le groupe d'experts mis à contribution dans le cadre de cette étude a joué un rôle précis dans un cadre défini [cf. annexe A].

Concernant le cadre global d'intervention, ce travail s'est largement inspiré du SEI dans les étapes 1 et 2 (figure 4.4). L'intervention des experts se situe aux premiers livrables de la 2ème étape. Le positionnement de ce travail par rapport aux étapes de l'élaboration du CMM par le SEI est indiqué par un **X** dans la figure 4.4. Pour l'étape 3 d'élaboration du modèle, une méthode plus structurée comme une méthode Delphi pourrait être utilisée. C'est une approche à considérer dans les étapes ultérieures de consolidation et de stabilisation du modèle.

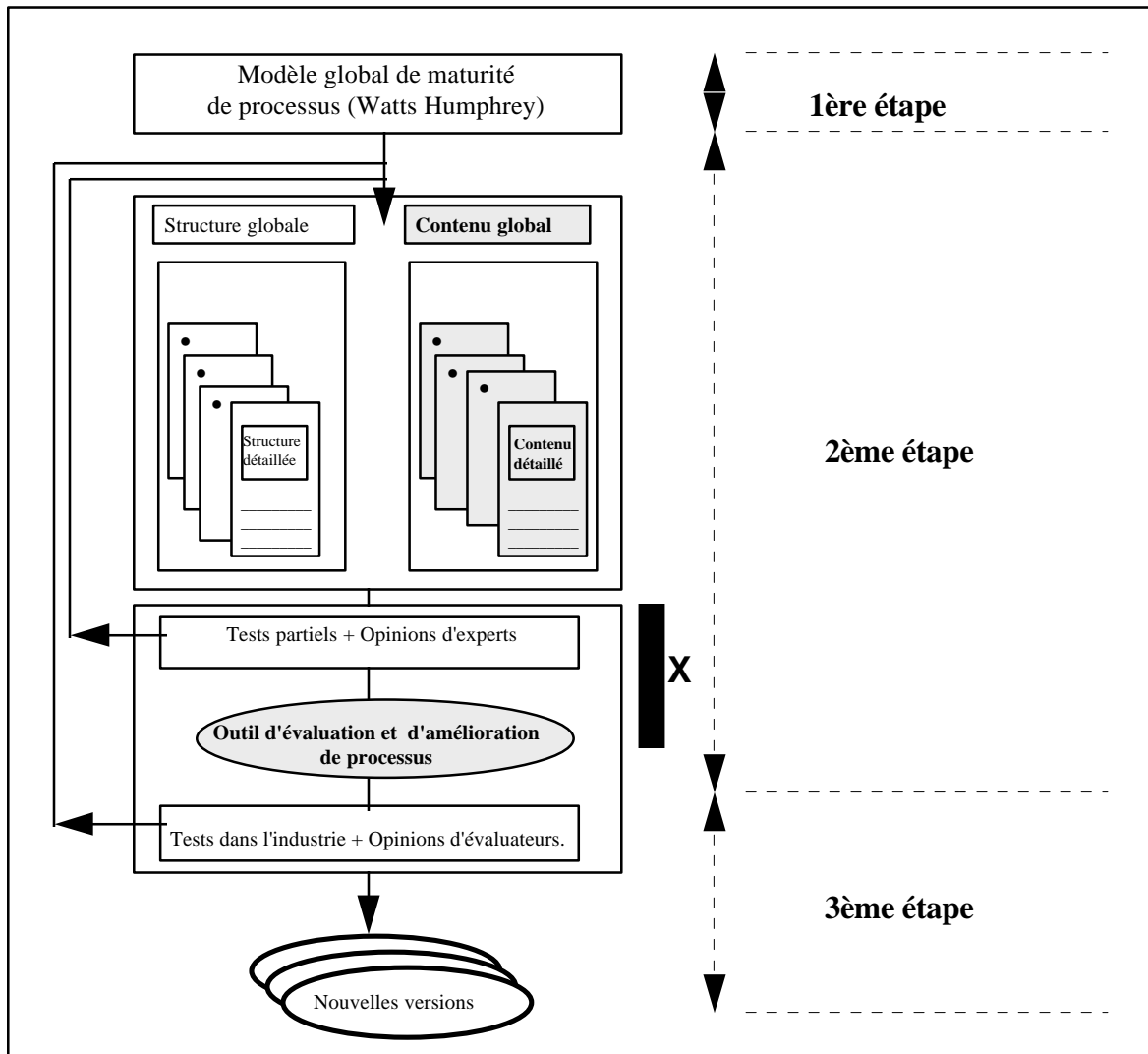


Figure 4.4 - Positionnement par rapport aux étapes d'élaboration du modèle CMM

Chapitre 5

Résultats : la version 1.0 du modèle de maturité de la maintenance logiciel

Le chapitre 5 présente la version 1.0 du modèle de maturité de la maintenance logiciel. Dans ce chapitre, on retrouve la présentation du modèle, les contenus global et détaillé du modèle ainsi que la comptabilisation des pratiques-clés à travers les secteurs-clés et les niveaux de maturité.

5.1 Présentation du modèle

La présentation du modèle nécessite de préciser nombre d'informations touchant autant la forme que le fond du modèle. Ces informations se trouvent dans le *guide des experts* [annexe A]. Il s'agit de la codification des objectifs-clés et des pratiques-clés, du cadre de présentation des secteurs-clés, ainsi que des précisions sur le rôle des experts dans le cadre de cette contribution d'enrichissement de contenu de modèle, de la limite de leur tâche, *etc.*

Le cadre global de présentation des résultats de l'étude est illustrée par la figure 5.1.

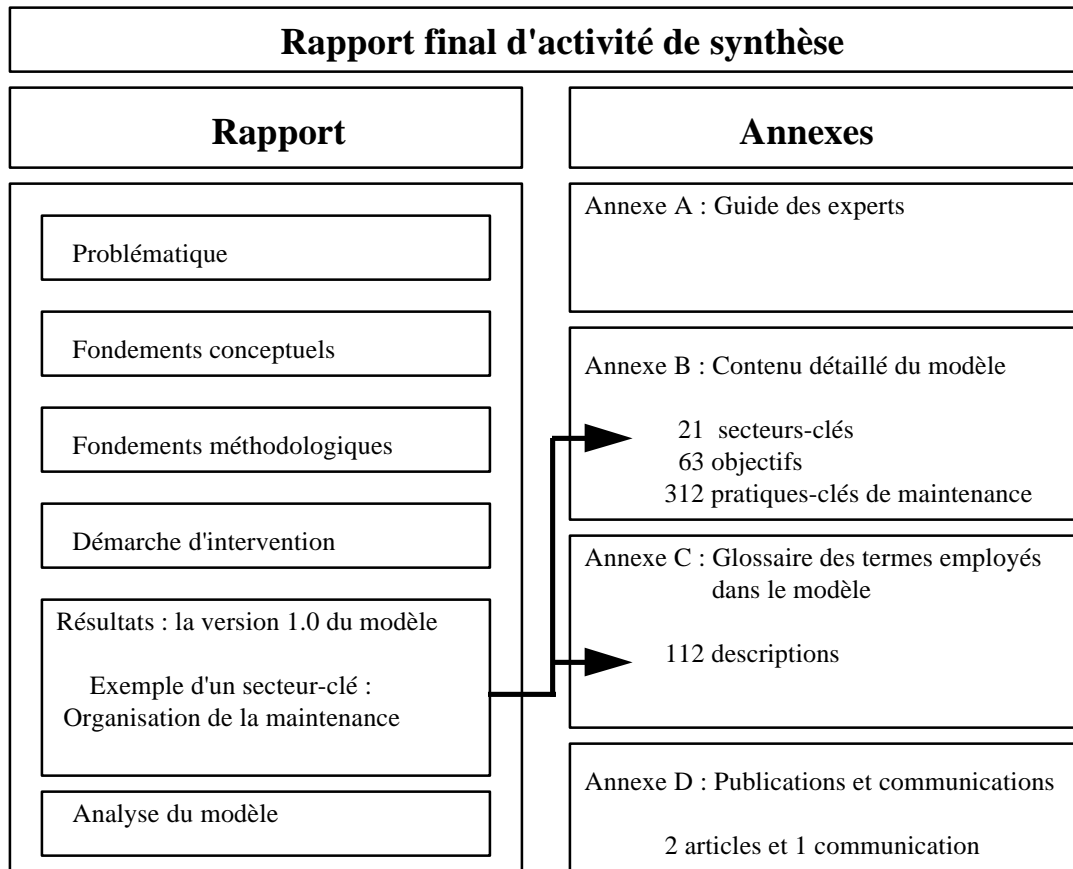


Figure 5.1 - Cadre global de présentation des résultats de l'étude

Comme l'indique la figure 5.1, le contenu détaillé du modèle de la maintenance logiciel qui décrit les objectifs-clés et les pratiques-clés de chaque secteur-clé à travers les différents niveaux de maturité s'appuie sur deux supports:

1) *Le contenu du modèle* [annexe B] : le document appelé "contenu du modèle de maintenance logiciel de la version 1.0" contient l'ensemble des objectifs-clés et des pratiques-clés des différents niveaux de maturité de l'ensemble des secteurs-clés du modèle de la maintenance logiciel.

2) *Le glossaire des termes du modèle de la maintenance* [annexe C] : le document appelé "glossaire des termes du modèle de la maintenance" contient une description de tous les termes utilisés dans le modèle et qui prennent une signification spécifique dans le contexte de la maintenance logiciel. Ces termes sont référencés dans le contenu détaillé du modèle.

5.2 Contenus global et détaillé du modèle de la maintenance

Le tableau 5.1 présente les contenus global et détaillé du modèle de maturité de la maintenance (version 1.0) avec le même ordonnancement que celui utilisé pour l'élaboration du modèle CMM du SEI présenté dans la figure 3.4 du chapitre 3. Le secteur-clé 07 Gestion de la transition vers la maintenance est illustrée avec ses objectifs-clés et ses pratiques-clés de maintenance.

La structure interne a, par contre, subi des modifications qui sont exposées au chapitre 6 de la présente étude (analyse des résultats).

← (global) →		(détaillé) →
Niveaux de maturité	Secteurs-clés de la maintenance	Objectifs-clés et pratiques-clés de la maintenance
<div style="border: 1px dashed black; padding: 2px; margin-bottom: 5px;">1. Initial</div> <div style="border: 1px dashed black; padding: 2px; margin-bottom: 5px;">2. Reproductible</div> <div style="border: 1px dashed black; padding: 2px; margin-bottom: 5px;">3. Défini</div> <div style="border: 1px dashed black; padding: 2px; margin-bottom: 5px;">4. Contrôlé</div> <div style="border: 1px dashed black; padding: 2px;">5. Optimisant</div>	<ul style="list-style-type: none"> •01. Gestion des spécifications •02. Planification de la maintenance logiciel •03. Suivi et supervision de la maintenance •04. Gestion de la sous-traitance •05. Assurance-qualité logiciel •06. Gestion de la configuration •07. Gestion de la transition vers la maintenance •08. Gestion des problèmes •09. Gestion du support aux usagers •10. Organisation de la fonction de maintenance •11. Rôle des intervenants autour de la aintenance •12. Priorités organisationnelles sur les processus •13. Définition du processus de l'organisation •14. Programme de formation •15. Gestion intégrée de la maintenance •16. Ingénierie des produits logiciel •17. Gestion quantitative du processus •18. Gestion de la qualité •19. Prévention des défauts •20. Gestion des changements technologiques •21. Gestion des améliorations du processus 	<ul style="list-style-type: none"> • Objectifs : 1. Le logiciel transféré au support et aux opérations est de qualité, complet et prêt selon les standards de l'acceptation de logiciel. 2. L'utilisateur est assuré que le logiciel qu'on lui livre répond bien à ses besoins et qu'il en est satisfait. · n. • Activités : 1. Un processus formel d'acceptation de logiciel est institué. 2. La phase d'acceptation est testée par l'équipe de support qui maîtrise les critères d'acceptation de logiciel et dispose de l'autorité finale de l'acceptation de logiciel. 3. Un rapport de post-audit d'acceptation de logiciel est institué. · n.

Tableau 5.1 - Contenus global et détaillé du modèle de maturité de la maintenance (version 1.0)

5.3 Exemple de secteur-clé du modèle de la maintenance

Prenons maintenant comme exemple de présentation le secteur-clé 10 du modèle de maturité de la maintenance (Organisation de la fonction de maintenance). Pour cela, il faut décrire au préalable le cadre de présentation standard élaboré pour contenir l'ensemble des secteurs-clés du modèle.

5.3.1 Cadre de présentation des secteurs-clés du modèle de la maintenance

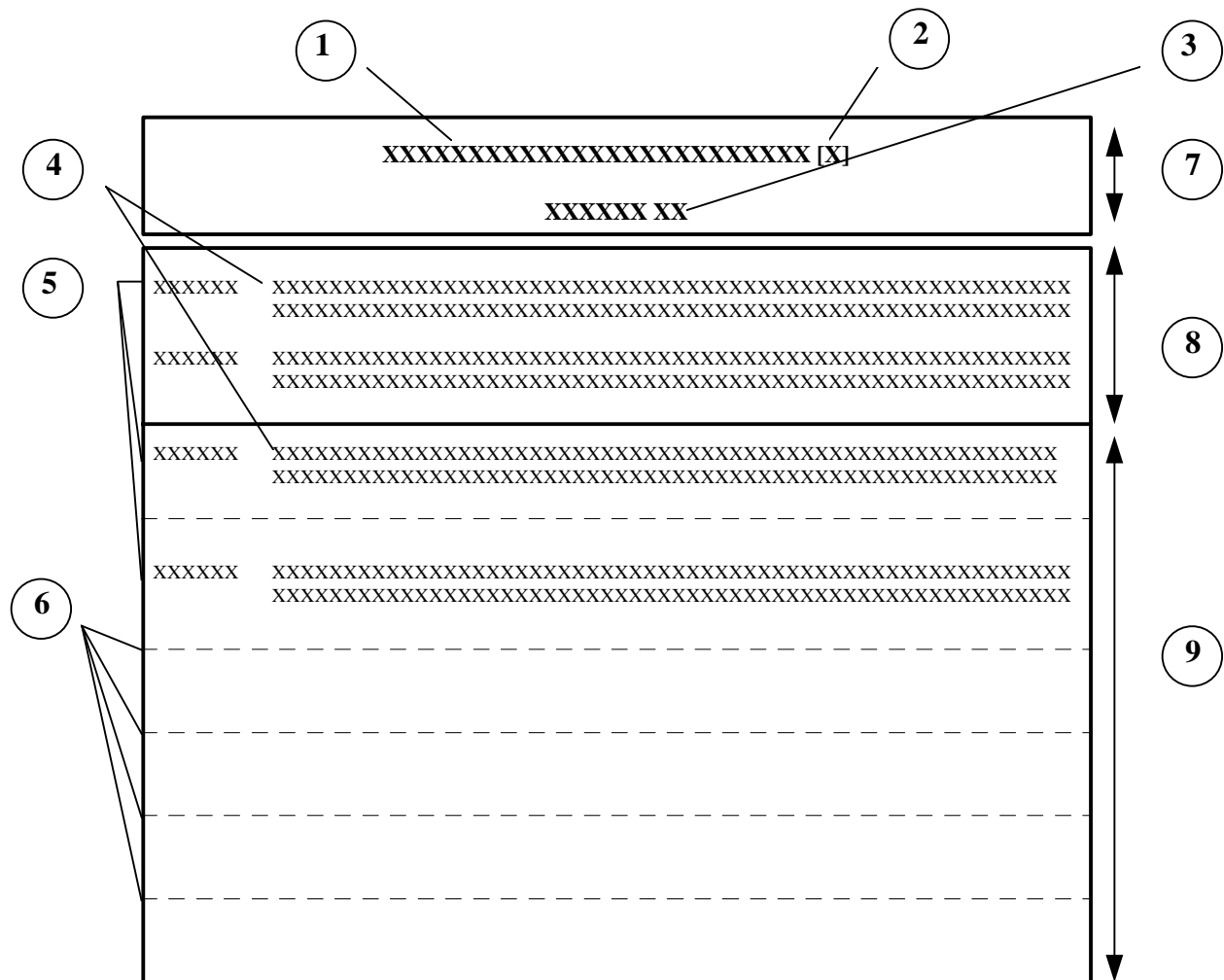


Figure 5.2 - Cadre standard de présentation du contenu du modèle de la maintenance version 1.0

La forme standard de présentation des secteurs-clés du modèle est décrite en annexe [annexe B] et est utilisée pour contenir d'une façon uniforme et homogène l'ensemble des secteurs-clés du modèle.

Le cadre de présentation est illustré par la figure 5.2.

1. Intitulé du secteur-clé.
2. Numéro du secteur-clé.
3. Référence du secteur-clé dans le glossaire des termes employés du modèle pour description sommaire (*cf.* glossaire en annexe C).
4. Description de l'objectif-clé à réaliser ou de la pratique-clé à mettre en oeuvre pour réaliser les objectifs-clés du secteur-clé *cf.* contenu du modèle en annexe B).
5. Code de l'objectif-clé ou de la pratique-clé comme référence (*cf.* guide des experts pour la structure de la codification en annexe A).
6. Séparateurs de niveaux de maturité pour les pratiques-clés (le niveau de maturité est contenu dans la codification de la pratique-clé).
7. Première partie du document : le titre du secteur-clé.
8. Deuxième partie du document : les objectifs-clés à réaliser par le secteur-clé.
9. Troisième partie du document : les pratiques-clés à réaliser pour atteindre les objectifs-clés du secteur-clé.

5.3.2 Contenu du secteur-clé "Organisation de la fonction de maintenance"

Le contenu détaillé du secteur-clé "Organisation de la fonction de maintenance" est décrit dans le tableau 5.2. Les pratiques-clés sont réparties par niveau de maturité (de 2 à 5) pour situer le degré d'influence de la pratique-clé dans le modèle.

Tous les objectifs-clés et les pratiques-clés sont codifiés pour établir un lien avec leur référence (ouvrage, auteur, *etc.*), comme indiqué dans la figure 5. 3.

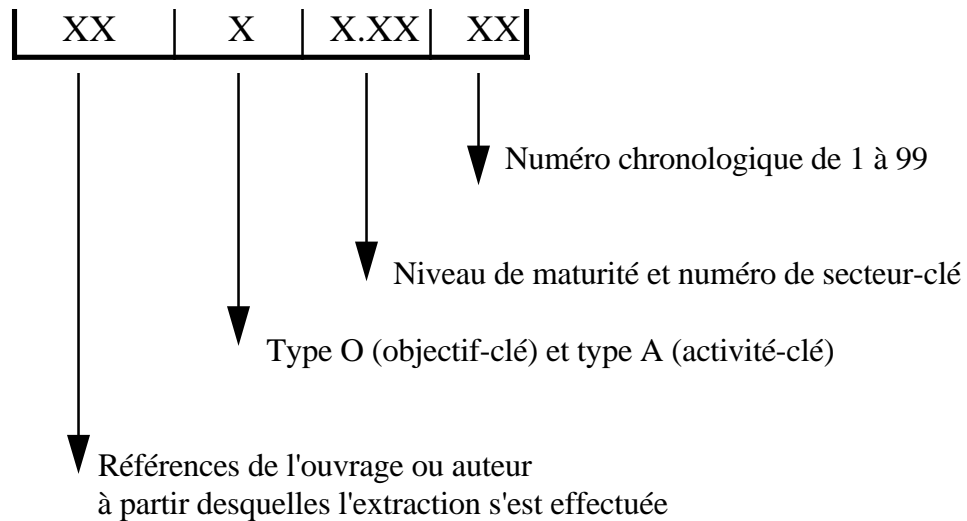


Figure 5.3 - Codification des objectifs-clés et des pratiques-clés

Les références d'ouvrages utilisés pour cette étude sont :

- MC Paulk, M.C., Weber, C.V., Garcia S., Chrissis, M.B et Bush, M. Pratiques clés du modèle d'évolution des capacités, version 1.1, Software Engineering Institute, CMU/SEI-93-TR-25 février 1993.
- SB Swanson, E.B., and Beath, C.M. (1989). Maintaining Information Systems in Organizations. John Wiley & Sons.
- RB Richard Ball (1990). Software Maintenance / Management. From The Seminar of Software Management Institute, North York, Ontario, April.
- GE Groupe d'experts formé dans le cadre de la présente étude.

Organisation de la fonction de maintenance logiciel [59]

(Secteur-clé 10)

-
- [SBO.10/01] La fonction de maintenance oeuvre pour rendre les portefeuilles d'applications plus performants.
-
- [GEA2.10/01] Tous les processus de gestion de maintenance sont détaillés en classe de maintenance [11].
- [RBA2.10/02] Des comités de gestion de logiciel [15] sont institués.
- [RBA2.10/03] Des réunions des comités de gestion de logiciel se tiennent dans le cadre du processus de gestion de logiciel.
- [RBA2.10/04] Un rapport de réunion de comité de gestion de logiciel [94] est obligatoirement diffusé.
- [RBA2.10/05] Un cycle de vie du logiciel [17] est considérée comme base de la gestion du logiciel.
- [SBA2.10/06] Les demandes de maintenance sont faites formellement, même après les faits, pour être enregistrées et rapportées.
- [SBA2.10/07] La responsabilité des tâches est partagée entre les utilisateurs et l'informatique.
-
- [GEA3.10/01] Le budget alloué à la maintenance du logiciel est réparti selon les processus de gestion de la maintenance et détaillé selon chaque classe de maintenance.
- [GEA3.10/02] Un niveau de service précis est établi pour chaque processus de gestion de la maintenance et chaque classe de maintenance;
- [GEA3.10/03] Les critères d'assignation à un processus de gestion de la maintenance est clairement défini.
- [SBA3.10/04] Des politiques spécifiques à la gestion des ressources humaines affectées à la maintenance sont établies.
-
- [SBA4.10/01] Un processus de mesures des tâches de maintenance logiciel [82] existe.

[SBA4.10/02]	La tâche de maintenance logiciel est perçue dans un contexte de portefeuille d'application pris dans son ensemble.
[SBA5.10/01]	Des techniques de gestion et modèles de recherche sont utilisés pour soutenir la planification et l'établissement de politiques spécifiques à la maintenance logiciel.
[SBA5.10/02]	Des approches d'innovations et de changement en matière organisationnelle sont encouragées.

Tableau 5.2 - Description des pratiques-clés du secteur-clé 10 du modèle de la maintenance

En résumé, au tableau 5.3, le secteur-clé Organisation de la fonction de maintenance est composé d'un objectif-clé et de 15 pratiques-clés réparties par niveau de maturité.

Nbre d'objectifs	Secteurs-clés du modèle de maintenance version 1.0	Nbre de pratiques	(1)	(2)	(3)	(4)	(5)
1	10. Organisation de la fonction de maintenance	15		7	4	2	2

Tableau 5.3 - Comptabilisation du secteur-clé 10 «Organisation de la fonction de maintenance logiciel»

Chapitre 6

Analyse des résultats

Le chapitre 6 présente une analyse des résultats présentés partiellement au chapitre précédent et complètement en annexe [annexe B]. Ce chapitre fait ressortir également les approches nouvelles que procure le nouveau modèle de maintenance ainsi que des pistes potentielles de recherche et un résumé sur les contributions dans le domaine de recherche (publications et communications).

6.1 Analyse quantitative du modèle

L'analyse des quantités d'objectifs-clés, de pratiques-clés et de secteurs-clés du modèle ainsi élaboré donnent des indications quant à la répartition des pratiques-clés par niveau de maturité et la répartition par ouvrage de référence à partir desquelles les extractions des pratiques-clés ont été effectuées.

6.1.1 Secteurs-clés du modèle de la maintenance

L'identification des pratiques-clés de maintenance, l'assignation de chaque pratique-clé à un secteur-clé donné et son appartenance à un niveau de maturité donné ont permis l'identification de domaines de gestion spécifiques à la maintenance du logiciel que l'on qualifiera de secteurs-clés du modèle adapté, comme indiqué par le tableau 6.1. Ce tableau présente également le nombre d'objectifs-clés par secteur-clé de maintenance ainsi que le nombre de pratiques-clés par niveau

de maturité. Les secteurs-clés du modèle de la maintenance sont au nombre de 21 représentant 63 objectifs-clés et 312 pratiques-clés.

Nbre d'objectifs	Secteurs-clés du modèle de maintenance version 1.0	Nbre de pratiques	(1)	(2)	(3)	(4)	(5)
2	01. Gestion des spécifications	4		2	2		
4	02. Planification de la maintenance logiciel	39		8	24	4	3
4	03. Suivi et supervision de la maintenance	21		12	7	2	
5	04. Gestion de la sous-traitance	14		11	3		
4	05. Assurance-qualité logiciel	15			14	1	
4	06. Gestion de la configuration	12		5	7		
3	07. Gestion de la transition vers la maintenance	22		14	5	2	1
2	08. Gestion des problèmes	11		2	8	1	
2	09. Gestion du support aux usagers	20		12	6	2	
1	10. Organisation de la fonction de maintenance	15		7	4	2	2
1	11. Rôles des intervenants autour de la maintenance	19		12	7		
3	12. Priorité organisationnelle sur les processus	7			6	1	
2	13. Définition du processus de l'organisation	6			6		
4	14. Programme de formation	25		4	18	3	
2	15. Gestion intégrée de la maintenance	11			10	1	
2	16. Ingénierie des produits	20		2	16	2	
3	17. Gestion quantitative du processus	13			1	12	
6	18. Gestion de la qualité	13		2	4	7	
3	19. Prévention des défauts	8		2	4		2
3	20. Gestion des changements technologiques	7		3	4		
3	21. Gestion des améliorations du processus	10		5	5		
63		312		103	161	30	8

Tableau 6.1 - Les secteurs-clés du modèle de maturité de la maintenance (version 1.0)

6.1.2 Pratiques-clés par niveau de maturité

La figure 6.1 indique également la répartition des pratiques-clés par niveau de maturité. On observe la concentration des pratiques-clés sur les deux niveaux de maturité qui se situent au milieu du cadre de maturité de processus de Watts Humphrey [Humphrey, 1988]. Les pratiques-clés de ces deux niveaux comptent pour 84% du modèle. Les capacités que représentent les deux niveaux de maturité appelés niveau répétable (niveau 2) et niveau défini (niveau 3) sont de

discipliner le processus, de le structurer et de le rendre visible. C'est à partir de ces deux niveaux que l'organisation peut disposer de bases solides pour procéder à des améliorations plus importantes et continues. C'est à ce niveau-là aussi (niveau 3) qu'elle commence à mesurer certaines tâches de façon efficace.

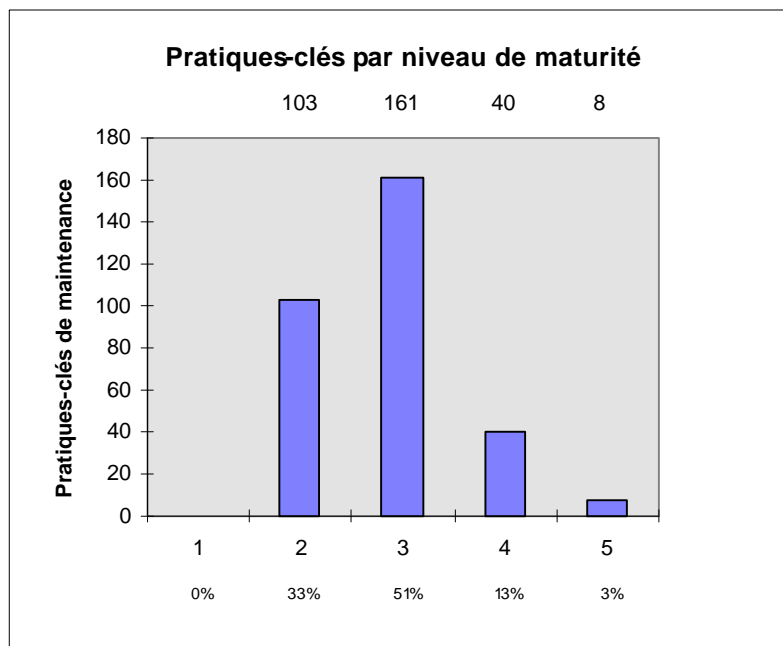


Figure 6.1 - Pratiques-clés par niveau de maturité du modèle de la maintenance version 1.0

6.1.3 Pratiques-clés par ouvrage de référence (auteur)

La figure 6.2 indique la répartition des pratiques-clés par ouvrage de référence (auteur) qui ont servi de support d'entrée à la présente étude (cf. flux d'entrée de modèle au chapitre 4). Cette répartition tient compte de l'ordre dans lequel l'étude de chaque ouvrage a été entreprise car d'un ouvrage à un autre, une pratique-clé similaire ne pouvait être retenue deux fois et se voyait comptabiliser au bénéfice du premier.

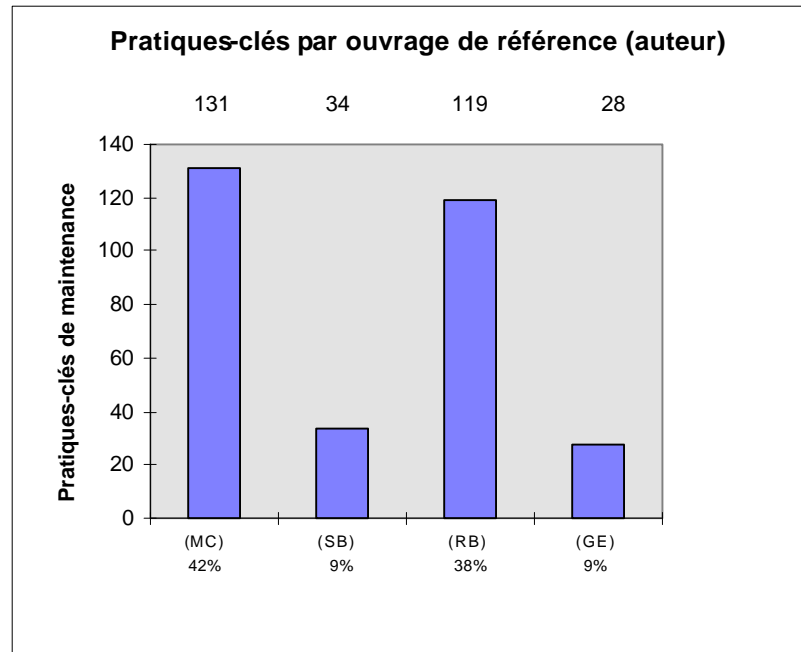


Figure 6.2 - Pratiques-clés par ouvrage de référence (auteur) du modèle de la maintenance version 1.0

Plusieurs observations retiennent l'attention :

- 1) Du contenu détaillé du modèle CMM du SEI ont été extraites 42% de pratiques-clés (modifiées ou non modifiées) qui correspondent à des activités de maintenance. Ces pratiques-clés sont des pratiques-clés de haut niveau du modèle de maturité du SEI (figure 6.2, colonne MC).
- 2) Les pratiques-clés extraites de l'ouvrage de Swanson & Beath représentent 9% des pratiques-clés du modèle (figure 6.2, colonne SB). Ces auteurs sont reconnus dans les domaines organisationnel et structurel de la maintenance du logiciel.
- 3) Un pourcentage substantiel de 38% caractérise les pratiques-clés extraites de l'ouvrage de Richard K. Ball (figure 6.2, colonne RB). C'est un ouvrage qui sert de support de cours et de présentation dans l'industrie dont l'auteur en est le fondateur et principal dirigeant. Le contenu de ces pratiques-clés reflètent la polyvalence des secteurs-clés qu'elles touchent. C'est aussi un

contenu tourné vers la pratique de la maintenance telle que vécue dans le milieu de la maintenance dans les organisations. Ce sont des pratiques courantes de bonne gestion de maintenance que l'on ne trouve pas actuellement en grand nombre dans la littérature consacrée à la littérature académique.

4) Les pratiques-clés provenant du dernier flux d'entrée du modèle, 9%, ont été apportées par le groupe d'experts chargé d'enrichir le modèle et de veiller à la cohérence de ses pratiques-clés (figure 6.2, colonne GE). Le faible pourcentage de ces pratiques-clés explique l'intervention des experts en dernier à l'étape de validation préliminaire du nouveau modèle.

6.2 Comparaison par rapport au modèle CMM du SEI

Le contenu du modèle de maturité du CMM du SEI a servi de flux d'entrée. De ce fait, les pratiques-clés et les secteurs-clés de la maintenance qui apparaissent dans le modèle adapté suite aux itérations de la phase de traitement sont de type modifié ou de type "nouveau" par rapport à ce contenu.

Le tableau 6.2 compare la source d'origine des secteurs-clés de maintenance aux secteurs-clés du modèle de maturité CMM du SEI consacré au développement. On y remarque que 6 secteurs-clés ont été extraits tels quels du modèle CMM du SEI, que 10 secteurs-clés ont été modifiés afin que leurs pratiques-clés correspondent plus à des pratiques-clés de maintenance et que 5 secteurs-clés de maintenance sont nouveaux (proviennent d'autres flux d'entrée) et sont basés sur des pratiques courantes de la maintenance logiciel et de la gestion de logiciel dans les organisations [Ball, 1990].

Secteurs-clés du modèle de maintenance version 1.0	Contenus des secteurs-clés comparés au modèle CMM du SEI		
	Non modifié	Modifié	Nouveau
01. Gestion des spécifications	X		
02. Planification de la maintenance logiciel		X	
03. Suivi et supervision de la maintenance		X	
04. Gestion de la sous-traitance		X	

05. Assurance-qualité logiciel		X	
06. Gestion de la configuration		X	
07. Gestion de la transition vers la maintenance			X
08. Gestion des problèmes			X
09. Gestion du support aux usagers			X
10. Organisation de la fonction de maintenance			X
11. Rôles des intervenants autour de la maintenance			X
12. Priorité organisationnelle sur les processus	X		
13. Définition du processus de l'organisation	X		
14. Programme de formation		X	
15. Gestion intégrée de la maintenance	X		
16. Ingénierie des produits		X	
17. Gestion quantitative du processus		X	
18. Gestion de la qualité		X	
19. Prévention des défauts		X	
20. Gestion des changements technologiques	X		
21. Gestion des améliorations du processus	X		

Tableau 6.2 - Contenus du modèle de maintenance *versus* contenus du modèle CMM du SEI

Il est à noter que pour une première version du modèle (version 1.0), les flux d'entrée qui servent de support d'extraction des pratiques-clés de maintenance sont limités (*cf.* envergure de l'étude au chapitre 4). L'élaboration des versions futures nécessitera des flux d'entrée plus importants.

Comparativement au modèle de maturité CMM du SEI, la maintenance du logiciel offre la possibilité d'être abordée d'une façon différente qui n'obéit pas à l'architecture interne du modèle CMM du SEI. Le modèle de maintenance logiciel pose le problème de la qualification du processus par niveau de maturité et par secteur-clé. Un secteur-clé de la maintenance a la possibilité de se situer graduellement à différents niveaux de maturité au fur et à mesure de son évolution dans le modèle.

Une des particularités de cette étude est que, comparativement au modèle CMM du SEI, les objectifs d'un secteur-clé, s'ils sont atteints, n'assurent pas obligatoirement la réussite du secteur-clé dans sa totalité car la version 1.0 est limitée volontairement aux seuls flux d'entrée retenus dans le cadre de cette étude. Ces flux d'entrée, rappelons-le, servent de support d'extraction des pratiques-clés et des objectifs-clés du modèle (*cf.* envergure de l'étude au chapitre 4).

6.3 Approche de secteurs multi-niveaux

L'observation des résultats obtenus et l'analyse des pratiques-clés par secteur-clé nous permet de constater une différence importante par rapport au modèle CMM du SEI : l'ensemble des pratiques-clés d'un secteur-clé donné n'est pas regroupé dans un seul niveau de maturité mais se trouve éparpillé dans différents niveaux de maturité selon la nature de la pratique-clé en question. Pour cela, faisons un rappel des capacités et des qualifications de processus par niveau de maturité et prenons un exemple de pratiques-clés d'un secteur-clé représentatif du modèle adapté.

Le tableau 6.3 nous indique comment évoluent les caractéristiques de capacité du processus d'un niveau à un autre niveau selon les règles, procédures et politiques édictées par le contenu global et détaillé du modèle de maturité CMM du SEI.

Niveaux	Caractéristiques de qualification de Processus
1	S'il y a existence de règles et de procédures, elles ne sont pas respectées. La capacité du processus est nulle.
2	Les règles et les procédures formelles existent. Le processus est discipliné. La capacité du processus à ce niveau-là est une caractéristique engendrée par les individus de l'organisation seulement.
3	Le processus est structuré et donne de la visibilité. L'aptitude du processus est considérée comme standard et consistante. Il est possible de commencer à mesurer certaines tâches de façon efficace. A ce niveau-là, l'organisation dispose d'une base solide pour entreprendre des changements importants.
4	L'organisation dispose de mesures de productivité et de qualité bien définies et consistantes. Le processus devient instrumenté.
5	Les données de mesure sont disponibles pour analyser et optimiser le processus par lui-même. Le processus est considéré comme optimisant. A ce stade, la capacité du processus peut être caractérisée l'amélioration continue.

Tableau 6.3 - Caractéristiques de qualification de processus par niveau de maturité

Pour l'exemple, on prend les pratiques-clés du secteur-clé "Planification de la maintenance" et on les compare avec les caractéristiques du tableau 6.3. On observe ainsi la nature de l'évolution des pratiques-clés de ce secteur à travers les différents niveaux de maturité du modèle, comme le montre le tableau 6.4. On se rend compte alors que c'est l'état d'avancement de la pratique-clé dans le modèle qui fixe son appartenance à un niveau de maturité et non pas la nature du secteur-clé qui le fait. Comparativement aux secteurs-clés qui appartiennent structurellement à un niveau de maturité donné dans le modèle de maturité CMM du SEI, cette approche de secteurs-clés multi-niveaux est inspirée de

l'approche développée dans le modèle TRILLIUM de Bell Canada, Northern Telecom et Research Bell Northern [April, 1995].

Niveaux de maturité	Pratiques-clés du secteur-clé : Planification de la maintenance
1	– Peu ou pas planifiée.
2	– Une procédure de planification de la maintenance logiciel est instituée. – <i>etc.</i>
3	– Un processus formel de réalisation des demandes de service en attente est utilisé pour résorber les files d'attente. – Les données de planification de la maintenance logiciel ainsi que les données de révision de la planification sont enregistrées à des fins de mesure. – La planification des RMPT et des versions de logiciel ainsi que leurs priorités sont flexibles et contrôlées par le client. – <i>etc.</i>
4	– La planification de la maintenance dans ses aspects de coûts, d'effort et de qualité applique rigoureusement les standards du développement. – <i>etc.</i>
5	– Les processus de planification de la maintenance s'appuient sur des modèles de productivité spécifiques à la maintenance logiciel. – <i>etc.</i>

Tableau 6.4 - Pratiques-clés du secteur-clé : Planification de la maintenance logiciel

Les résultats engendrés par une telle approche donnent une nouvelle approche : un secteur-clé peut se trouver à différents niveaux de maturité selon la nature et l'envergure de ses pratiques-clés. Le tableau 6.5 nous indique les secteurs-clés du modèle et la répartition de leurs pratiques-clés à travers les différents niveaux de maturité.

Secteurs-clés du modèle de maintenance version 1.0	Niveaux de maturité				
	(1)	(2)	(3)	(4)	(5)
01. Gestion des spécifications					
02. Planification de la maintenance logiciel					
03. Suivi et supervision de la maintenance					
04. Gestion de la sous-traitance					
05. Assurance-qualité logiciel					
06. Gestion de la configuration					
07. Gestion de la transition vers la maintenance					

08. Gestion des problèmes					
09. Gestion du support aux usagers					
10. Organisation de la fonction de maintenance					
11. Rôles des intervenants autour de la maintenance					
12. Priorité organisationnelle sur les processus					
13. Définition du processus de l'organisation					
14. Programme de formation					
15. Gestion intégrée de la maintenance					
16. Ingénierie des produits					
17. Gestion quantitative du processus					
18. Gestion de la qualité					
19. Prévention des défauts					
20. Gestion des changements technologiques					
21. Gestion des améliorations du processus					

Tableau 6.5 - Approche des secteurs multi-niveaux du modèle de la maintenance version 1.0

6.4 Pistes potentielles de recherche

Le modèle ainsi élaboré peut servir, à travers la mise en oeuvre de ses pratiques-clés, d'instrument et de cadre d'implantation de programmes d'amélioration de processus de la fonction de maintenance de logiciels dans les organisations.

Pour améliorer la version 1.0 et produire la version 2.0 du modèle, il faudrait réaliser les tâches suivantes :

- 1) Finaliser l'insertion et l'ordonnement des objectifs, pratiques clés, secteurs-clés , niveaux de maturité;
- 2) Procéder au raffinement de la position des pratiques-clés dans le modèle selon leurs caractéristiques propres;
- 3) Procéder à des itérations complémentaires des phases d'itération du modèle;
- 4) Élaborer le questionnaire de maintenance correspondant au modèle de maintenance pour proposer un support d'évaluation du processus de maintenance dans l'organisation.

Dans une perspective beaucoup plus large, il faudrait élargir les sources des flux d'entrée (autres ouvrages, enquêtes, articles, *etc.*). A cela, d'autres études seraient nécessaires pour compléter et ajuster le modèle, notamment des études empiriques pour :

- S'assurer que les pratiques-clés réalisées dans les organisations se trouvent bien dans le modèle et à la bonne place des niveaux de maturité, sinon procéder à des transferts et à des compléments;
- Étudier à travers le modèle adapté ainsi élaboré et les études empiriques à réaliser les différents contextes de gestion de maintenance de logiciels dans les organisations, leur faire subir des traitements statistiques et en tirer des enseignements pour donner plus de poids aux recommandations à faire dans le cadre d'implantations du modèle dans les organisations;
- Amorcer la recherche d'une approche organisationnelle cohérente et standard pour établir un cadre d'implantation de programmes d'amélioration de maintenance du logiciel dans les organisations.

Les études empiriques contribueront à une meilleure compréhension des problématiques de la maintenance du logiciel dans les organisations et aideront à trancher entre le choix de procéder par liens d'appartenance des secteurs-clés à un niveau de maturité donné, comme le suggère le modèle CMM du SEI pour le développement, ou le choix de procéder par passage des secteurs-clés dans les différents niveaux de maturité du modèle selon la nature et l'envergure de la pratique-clé du secteur-clé en question.

6.5 Diffusion des résultats

Nous avons eu l'occasion tout au long de la présente étude de faire connaître le sujet traité, le domaine de la maintenance logiciel et l'intérêt des pratiques de génie logiciel à la problématique de la maintenance logiciel.

D'abord, la publication d'un article intitulé "Élaboration d'un outil d'évaluation et d'amélioration du processus de maintenance de logiciels : une piste de recherche." dans les Actes des Huitièmes journées internationales du génie logiciel et de ses applications à Paris-La Défense [Zitouni, 1995]. Cet article [annexe D] est un résumé de la proposition de recherche présentée au sous-comité d'évaluation du programme de la maîtrise en informatique de gestion le 24 août 1995.

Ensuite, une proposition de publication d'un article intitulé "*A model to evaluate and improve the process quality of software maintenance*" a été soumise au comité organisateur de la Conférence annuelle internationale sur la qualité des logiciels qui se tient à Ottawa (Canada) en Novembre 1996.

Et finalement, une communication au 64ème congrès de l'Association Canadienne Française pour l'Avancement des Sciences (ACFAS) qui se tiendra du 13 au 17 mai 1996 à l'Université McGill (Montréal, Canada) a été acceptée.

Bibliographie

Abran, A. et Nguyenkim, H. (1991). Analysis of Maintenance Work Categories Through Measurement. *Proceedings of the IEEE Conference on Software Maintenance*, October, pp. 104-113.

Abran, A. et Nguyenkim, H. (1993). Measurement of the Maintenance Process from a Demand-based Perspective. *Software Maintenance : Research et Practice*, Vol. 5, pp.63-69.

ANSI / IEEE Standard 729-1983 (1983). *IEEE Standard Glossary of Software Engineering Terminology.*

ANSI / IEEE Standard 1074-1991 (1991). *IEEE Standard for Developing Software Life Cycle Processes.*

ANSI / IEEE Standard 1219-1992 (1992). *IEEE Standard for Software Maintenance.*

April, A. et Coallier, F. (1995). *TRILLIUM V3.0: A Model for the Assessment of Telecom Software System Development Capability.* Proceedings of the 2nd International SPICE Symposium, 1-2 June, Brisbane, Australia, pp. 79-88.

Arthur Lowel Jay (1988). *Software Evolution-The Software Maintenance Challenge.* John Wiley & Sons.

Ball, R.K. (1987). Voir Zvegintzov, N. (1991).

Ball, R.K. (1990). *Software Maintenance / Management.* From The Seminar of Software Management Institute, North York, Ontario, April.

Banker, R.D Datar, S.M & Kemerer, C.F (1991). A model to evaluate variables impacting the productivity of software maintenance projects. *Management Science*, Vol. 37 (1), pp. 1-18.

Benjamin, R.I., Rockart, J.F. (1984). Information Technology : A Strategic Opportunity. *Sloan Management Review*. Vol. 25 (3), Hiver, pp. 3-10.

Boehm, B.W. (1987). Improving Software Productivity. *IEEE Computer*, September, pp. 43-57.

Boynton, A.C., and Zmud, R.W. (1987). Information technology planning in the 1990's : Directions for Practice and Research. *MIS quarterly*. Vol. 11 (1), March, pp.59-71.

Cash, J.I., McFarlan, F.W., and McKenney, J.L. (1988). Corporate Information Systems Management : The Issue Facing Senior Executives. *Dow Jones-Irwin*, Homewood.

Desharnais, J.M. (1988). Analyse statistique de la productivité des projets de développement en informatique à partir de la technique des points de fonction. *Rapport d'activité de synthèse de la maîtrise en informatique de gestion*, Université du Québec à Montréal.

Dekleva, S. (1990). 1990 Annual Software Maintenance Survey. (*Survey conducted and compiled for the Software Maintenance Association*). P.O. Box 12004 no.297, Vallejo, CA, Voir Zvegintzov, N. (1991).

Fugetta, A. (1993). A Classification of CASE Technology. *Computer*, December, pp. 25-38.

Hale, D.P et Haworth, D.A. (1988). Software Maintenance : A Profile of Past Empirical Research. *Proceedings of the IEEE Maintenance Conference*, Washington DC. IEEE Computer Society Press.

Harrison, R. (1987). Maintenance Giant Sleeps Undisturbed in Federal Data Centers. *Computerworld*. Vol. 21 (10), 9 March, pp. 81-86.

Humphrey, W. S. (1988). Characterizing the Software Process : A Maturity Framework, *IEEE Software*, March, pp. 73-79.

Humphrey, W. S, T.R. Snyder, and R.R. Willis (1991). Software Process Improvement at Hugues Aircraft. *IEEE Software*. July, pp. 11-23.

Johnson, J.R. (1991). The Software Factory - Managing Software Development and Maintenance. 2nd edition, *QED Information Science*.

King, W.R. (1978). Strategic Planning for MIS. *MIS Quarterly*. Vol. 2 (1), March, pp. 27-37.

Lientz, B.P., and Swanson, E.B. (1980). Software Maintenance Management. Addison-Wesley, Reading, Mass.

Martin, J., and McClure, C.L. (1983). Software Maintenance : The Problem and its Solution. *Prentice-Hall*, Englewood Clifs, NJ.

Nolan, R.L. (1979). Managing the Crisis in Data Processing. *Harvard Business Review*. Vol. 57 (2), March-April, pp. 115-126.

Parikh, G. (1986). Handbook of Software Maintenance. John Wiley & Sons.

Parikh, G. (1986). Exploring the World of Software Maintenance. *ACM SIGSOFT software engineering notes*. Vol. 11(5), pp. 86-90.

Paulk, M. C. & Curtis, B. & Chrissis, M. B. (1991). Capability Maturity Model for Software. *Technical report CMU/SEI-91-TR-24 (ESD-TR-91-24)*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania. August.

Paulk, M. C. & Curtis, B. & Chrissis, M. B. & Weber, C. V. (1993a). Capability Maturity Model, Version 1.1, *IEEE Software*, July, pp. 18-27.

Paulk, M. C. et al. (1993b). Appendix C : Abridged Version of the Key Practices, Key Practices of the CMM, Version 1.1 Technical report. *CMU/SEI-93-TR-25*. Software Engineering Institute, Pittsburgh, pp. A27-A65.

Port, O. (1988). The Software Trap - Automate or Else. *Business Week. Special report*, 9 May, pp. 142-154

Porter, M.E., and Millar, V.E. (1985). How Information Gives you Competitive Advantage. *Harward Business Review*. Vol. 63 (4), July-August, pp. 149-160.

Sharpe, S., Haworth, D.A. et Hale, D. (1991). Characteristics of Empirical Software Maintenance Studies : 1980-1989. *Journal of Software Maintenance Research and practice*. Vol.3, pp. 1-5

Schneidewind, N. (1987). The State of Software Maintenance. *IEEE Transactions on Software Engineering*. Volume SE-13 (3), March, pp 303-310.

Swanson, E.B., and Beath, C.M. (1989). Maintaining Information Systems in Organizations. John Wiley & Sons.

Vennix, J.A.M et J.W. Gubbels (1991). Knowledge Elicitation in Conceptual Model Building. *European Journal of Operational Research, North-Holland.* Vol 59, pp. 85-101.

Zitouni, M., Abran, A., Bourque, P. (1995). Élaboration d'un outil d'évaluation et d'amélioration du processus de maintenance de logiciels : une piste de recherche. *Actes aux "Huitièmes journées internationales du génie logiciel et de ses applications"*, La Défense-Paris, Novembre 1995, pp. 727-739.

Zvegintzov N. (1991). Real Maintenance Statistics. *Software Maintenance News*, Vol. 9 (2), pp. 6-9

Les annexes

Annexe A :	Guide des experts	69
Annexe B :	Contenu du modèle de maturité de la maintenance logiciel (version 1.0)	77
Annexe C :	Glossaire des termes employés dans le modèle de maturité de la maintenance (version 1.0)	119
Annexe D :	Article ayant fait l'objet de publication	153

Annexe A : Guide des experts

Plan de présentation du document

	1. Courrier d'accompagnement
71	
	2. Activité de synthèse de l'étudiant
72	
	3. Élaboration du modèle (document préparatoire)
73	
	4. Tâche de l'étudiant
74	
	5. Rôle des experts
74	
	6. Codification des objectifs-clés et pratiques-clés
75	
	7. Limites du modèle à élaborer
76	

1. Courrier d'accompagnement

Montréal le 29 janvier 1996

A l'attention de Monsieur Alain April,
Conseiller en ingénierie de la qualité et approvisionnements,
Bell Canada,
Bureau 226,
2265 boulevard Roland Therrien
Longueuil, Québec
J4N 1C5

Objet : participation à un comité d'experts

Monsieur,

Dans le cadre du programme de la maîtrise en informatique de gestion que je poursuis actuellement sous la direction du professeur Alain Abran, du département d'informatique de l'UQAM, je vous demande de bien vouloir accepter de faire partie d'un comité d'experts chargé de réviser le modèle préliminaire de la maintenance des logiciels qui sert de base à mon activité de synthèse. Cette révision consiste en une validation informelle du contenu et d'un enrichissement que vous pouvez apporter au besoin.

Mon choix s'est porté sur votre personne pour plusieurs raisons dont l'éventail de vos connaissances acquises dans le domaine du génie logiciel ainsi que votre large expertise industrielle.

Vous trouverez ci-joint, en plus du rapport de mon activité de synthèse, les explications nécessaires pour vous guider à travers le document en question et un article de Zitouni, Abran et Bourque (Paris, novembre 95) vous décrivant les fondements conceptuels et méthodologiques sur lesquels reposent la conception et l'élaboration de ce travail.

Vous en remerciant par avance, veuillez agréer, Monsieur, mes salutations les meilleures.

Mohamed Zitouni,
étudiant à la maîtrise en informatique de gestion, UQAM
cc: Alain Abran, Professeur au département informatique, Directeur de recherche

2. Activité de synthèse de l'étudiant

Le programme MIG (maîtrise en informatique de gestion) de l'UQAM comporte deux parties. La première partie concerne la scolarité de maîtrise et est réalisée moyennant dix cours magistraux représentant trente crédits. La deuxième partie appelée aussi «activité de synthèse» se réalise sous la supervision d'un directeur de recherche, professeur du département informatique. Le succès à l'activité de synthèse représente quinze crédits. Cette activité de synthèse met à contribution l'ensemble des connaissances acquises pendant la première partie du programme par l'étudiant au service d'une problématique puisée dans le domaine de l'informatique.

Dans le cas présent l'activité de synthèse traite une des problématiques de la maintenance des logiciels dans les organisations, soit la difficulté de procéder à son évaluation et de lui apporter des pistes de solution pour son amélioration.

L'ordonnement de la réalisation de mon activité de synthèse est illustré dans le tableau 1.

Les étapes	Réalisation
1. Choix du directeur de recherche, accord sur le domaine, le sujet, l'envergure du projet, la méthode de recherche utilisée, l'échéancier, la qualité, <i>etc.</i>	oui
2. Réalisation du rapport de la proposition de recherche	oui
3. Présentation orale de la proposition devant le comité d'évaluation de la maîtrise	oui
4. Réalisation des livrables	oui
5. Enrichissement des livrables par un comité d'experts	en cours
6. Réalisation du rapport final de l'activité de synthèse	à faire
7. Présentation orale du rapport final devant le comité d'évaluation de la maîtrise	à faire

Tableau 1

3. Élaboration du modèle de la maintenance logiciel (document préparatoire)

Le document préparatoire soumis aux experts contient les secteurs et activités clés spécifiques à la maintenance du logiciel par niveau de maturité. La hiérarchie qui articule les niveaux de maturité, secteurs clés, activités clés, *etc.* dans le modèle CMM du SEI est schématisée dans la figure 1.

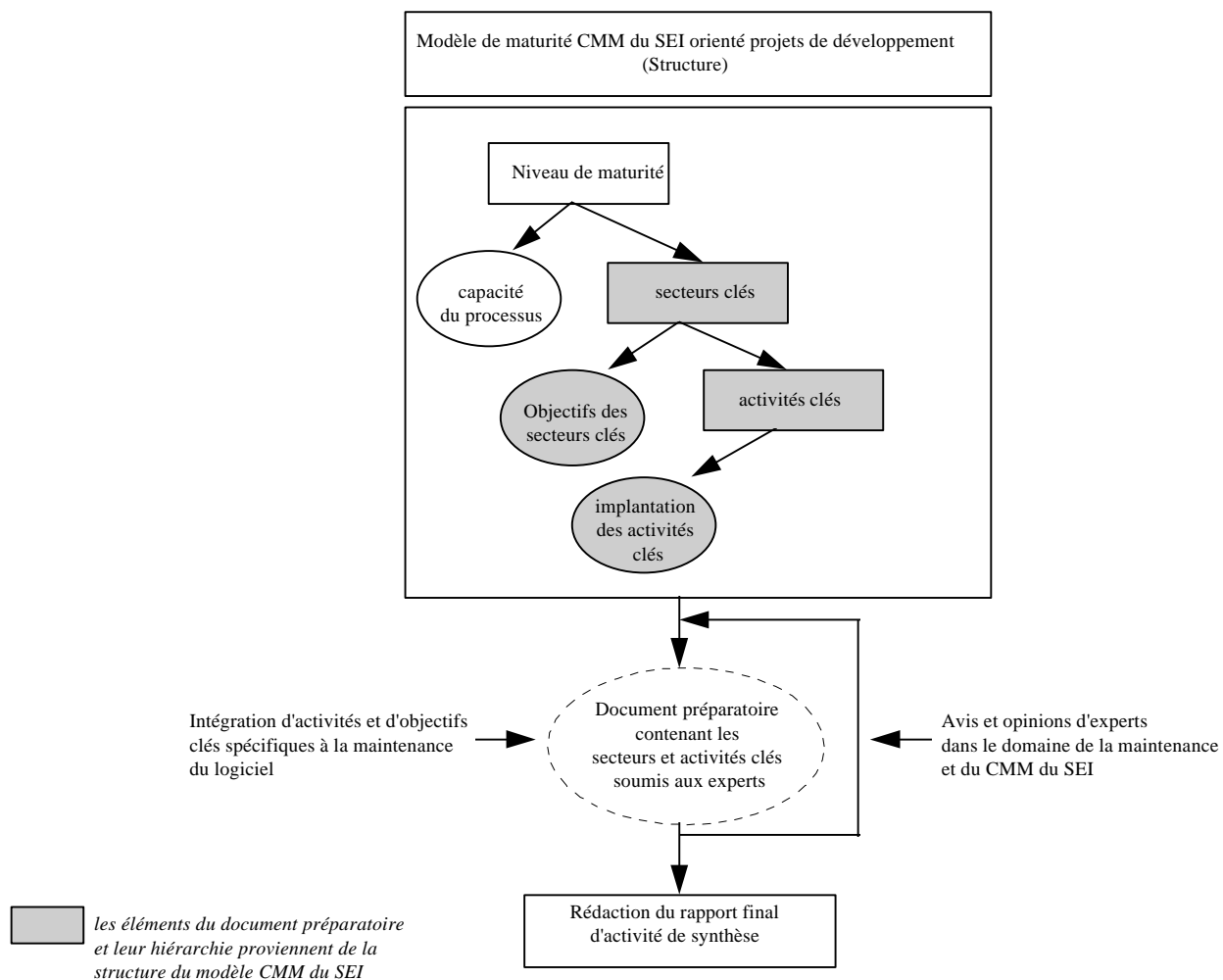


Figure 1

Les avis et opinions d'experts permettent d'apporter des suggestions et des améliorations pour enrichir ce qui est proposé dans le document préparatoire en plus de compléments de secteurs ou activités clés manquantes dans le document préparatoire.

Pour un bref rappel de la structure du modèle CMM du SEI, voir l'article publié par Zitouni, Abran, Bourque, novembre 95, se trouvant à la fin du document préparatoire.

4. Tâche de l'étudiant

D'abord identifier les problématiques auxquelles est confrontée la fonction de maintenance, inventorier les solutions correspondantes, choisir les composantes (objectifs et activités), les ordonner, les modéliser, les insérer à la bonne place dans le nouveau modèle, fournir ces composantes aux experts à des fins de d'amélioration et d'enrichissement. Ensuite définir des objectifs réalisables pour chaque secteur-clé et faire correspondre les pratiques-clés aux secteurs-clés. Et finalement élaborer les questions pertinentes touchant toutes les pratiques-clés et les secteurs-clés associés pour construire le questionnaire de maturité.

5. Rôle des experts

Le rôle des experts consiste à parcourir le document préparatoire afin de se prononcer sur:

- 1) Est-ce que les secteurs clés spécifiques à la maintenance du logiciel décrits dans le document se trouvent dans les niveaux de maturité appropriés?
- 2) Est-ce que les objectifs spécifiques à la maintenance du logiciel décrits dans le document se trouvent dans les secteurs clés appropriés?
- 3) Est-ce que les activités clés (pratiques clés) spécifiques à la maintenance du logiciel décrites dans le document se trouvent dans les secteurs clés appropriés?
- 4) Y a-t-il d'autres objectifs spécifiques à la maintenance du logiciel à rajouter et à relier aux secteurs clés appropriés?
- 5) Y a-t-il d'autres activités clés spécifiques à la maintenance du logiciel à rajouter dans les secteurs clés appropriés?
- 6) Y a-t-il d'autres secteurs clés spécifiques à la maintenance du logiciel à rajouter dans les niveaux de maturité appropriés?

Dans le contexte présent, l'expert se doit de signaler toute anomalie importante de cohérence ou anomalie de toute autre nature concernant le fond comme la forme. Les suggestions d'enrichissement de ce travail sont les bienvenues.

Pour faciliter la tâche de l'expert, la codification des objectifs, activités, secteurs clés et niveaux de maturité lui permet de porter ses réflexions directement sur le document préparatoire. Une fois cette tâche accomplie, il est souhaitable que l'expert reçoive l'étudiant pour une entrevue de l'ordre d'une demi journée (en une ou deux fois) afin de clarifier les différents points de vive voix. Un échéancier est proposé

6. Codification des objectifs-clés et pratiques-clés

Afin de faciliter la manipulation et la référencement des objectifs-clés, pratiques-clés et secteurs-clés, une codification simple est proposée.

Chaque pratique-clé est codifiée dans le format:

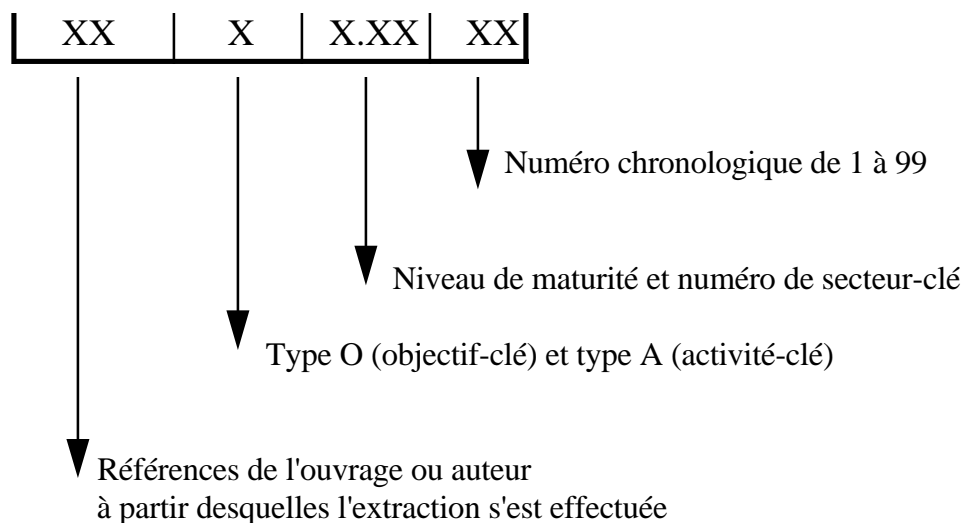


Figure 2

Les références d'ouvrages utilisés pour cette étude sont :

- MC Paulk, M.C., Weber, C.V., Garcia S., Chrissis, M.B et Bush, M. Pratiques clés du modèle d'évolution des capacités, version 1.1, Software Engineering Institute, CMU/SEI-93-TR-25 février 1993.

- SB Swanson, E.B., and Beath, C.M. (1989). *Maintaining Information Systems in Organizations*. John Wiley & Sons.
- RB Richard Ball (1990). *Software Maintenance / Management*. From The Seminar of Software Management Institute, North York, Ontario, April.
- GE Groupe d'experts formé dans le cadre de la présente étude.

7. Limites du modèle à élaborer

Dans la structure du CMM-SEI, les objectifs d'un secteur-clé sont atteints lorsque l'ensemble des pratiques clés du secteur clé soit implanté avec succès. Dans le modèle CMM du SEI, comme l'indique la figure 3, l'implantation des pratiques clés se réalise avec le concours d'autres pratiques-clés de support comme les engagements de réalisation, les capacités de réalisation, *etc.*, appelées caractéristiques communes de réalisation.

Dans le contenu du modèle à élaborer ci-joint, seuls les objectifs-clés et les pratiques-clés par secteur-clé et par niveau de maturité sont traités. Ces pratiques-clés sont inventoriées d'une façon globale par secteur-clé et par niveau de maturité. Les pratiques-clés du modèle contiennent déjà l'ensemble des caractéristiques communes de réalisation comme les engagements de réalisation, les capacités de réalisation, de vérification, *etc.*

Élaboration des pratiques clés selon la structure du CMM du SEI.
La structure réutilisée dans l'activité de synthèse ne touche que les parties Objectifs et pratiques clés.

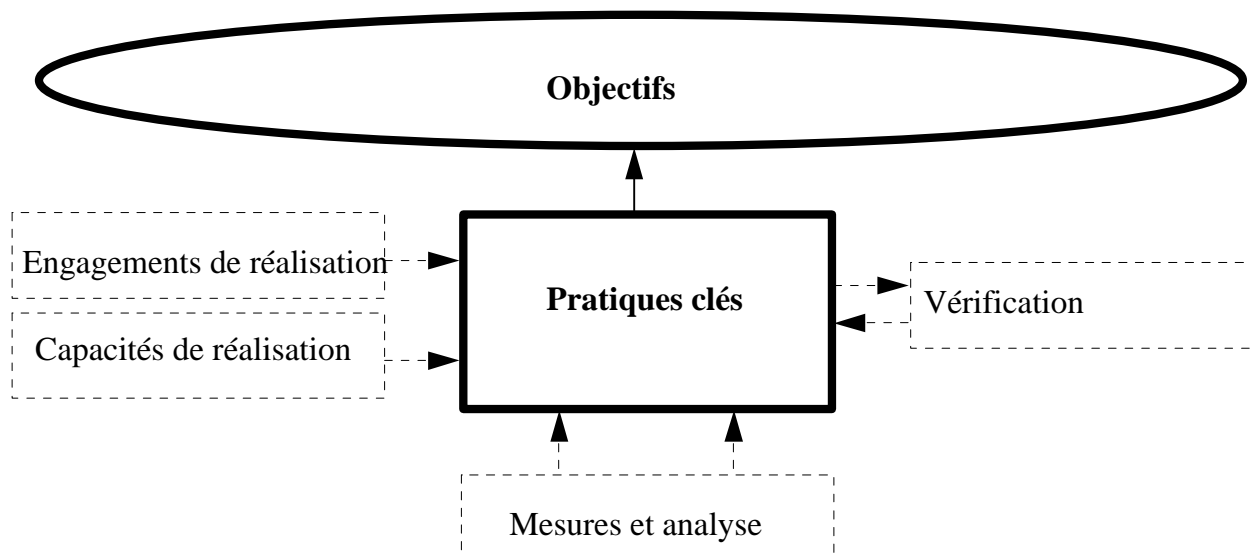


Figure 3

**Annexe B : Contenu du modèle de maturité de la maintenance
logiciel (version 1.0)**

4. Description de l'objectif-clé à réaliser ou de la pratique-clé à mettre en oeuvre pour réaliser les objectifs-clés du secteur-clé (*cf.* contenu du modèle en annexe B).
5. Code de l'objectif-clé ou de la pratique-clé comme référence (*cf.* guide des experts pour la structure de la codification en annexe A).
6. Séparateurs de niveaux de maturité pour les pratiques-clés (le niveau de maturité est contenu dans la codification de la pratique-clé).
7. Première partie du document : le titre du secteur-clé.
8. Deuxième partie du document : les objectifs-clés à réaliser par le secteur-clé.
9. Troisième partie du document : les pratiques-clés à réaliser pour atteindre les objectifs-clés du secteur-clé.

2) Contenu du modèle de maintenance version 1.0

Les secteurs-clés du modèle

01. Gestion des spécifications	81
02. Planification de la maintenance logiciel	82
03. Suivi et supervision de la maintenance logiciel	86
04. Gestion de la sous-traitance	88
05. Assurance-qualité logiciel	90
06. Gestion de la configuration	92
07. Gestion de la transition vers la maintenance	94
08. Gestion des problèmes	96
09. Gestion du support aux usagers	97
10. Organisation de la fonction de maintenance logiciel	99
11. Rôles des intervenants autour de la maintenance	101
12. Priorité organisationnelle sur les processus	103
13. Définition du processus de l'organisation	104
14. Gestion de la formation	105
15. Gestion intégrée de la maintenance	107
16. Ingénierie des produits	109
17. Gestion quantitative du processus	111
18. Gestion de la qualité	113
19. Prévention des défauts	115
20. Gestion des changements technologiques	116
21. Gestion des améliorations du processus	117

Gestion des spécifications [35]

(Secteur-clé 01)

[MCO.01/01] Une cohérence entre les nouvelles spécifications du client en matière de maintenance et le système existant est maintenue.

[MCO.01/02] Il est tenu compte des exigences client et des spécifications techniques dans l'étude des objectifs planifiés d'un projet de maintenance [91] ou d'une version planifiée de logiciel [112].

[GEA2.01/01] Un référentiel des spécifications logiciel existe.

[MCA2.01/02] Les changements apportés aux spécifications du client font l'objet d'une entente entre les gestionnaires de maintenance et le client.

[MCA3.01/01] Le personnel de maintenance passe en revue les exigences techniques du projet de maintenance ou de la version planifiée de logiciel à des fins de réalisation avant la prise en charge de la réalisation.

[MCA3.01/02] Les gestionnaires de la maintenance utilisent les spécifications du clients et les spécifications techniques comme base pour les plans, produits de travail et activités de maintenance à réaliser.

Planification de la maintenance logiciel [65]

(Secteur-clé 02)

-
- [GEO.02/01] La planification des projets de maintenance, des RMPT [98] et des versions planifiées de logiciel est organisée et stable.
- [MCO.02/02] La planification et la documentation des activités et engagements pris dans le cadre de projets de maintenance, RMPT et mises en oeuvre de versions planifiées de logiciel sont réalisées.
- [MCO.02/03] Les groupes et les personnes concernées par la planification de la maintenance logiciel acceptent leurs engagements relatifs aux projets de maintenance, RMPT et mises en oeuvre de versions planifiées de logiciel.
- [MCO.02/04] La documentation des estimations de projets de maintenance, RMPT et mises en oeuvre de versions planifiées de logiciel est réalisée pour des fins de planification et de suivi.
-
- [MCA2.02/01] La planification de projet de maintenance [66] ou de version planifiée de logiciel [68] est entreprise aux premières étapes de la planification globale des projets et des versions planifiées de logiciel.
- [MCA2.02/02] Une procédure de planification de RMPT [67] est instituée.
- [MCA2.02/03] La planification du projet de maintenance est documentée.
- [MCA2.02/04] Les produits de travail du projet de maintenance ou de la version planifiée de logiciel nécessaires au contrôle du projet de maintenance ou de la version planifiée de logiciel sont identifiés.
- [MCA2.02/05] Des estimations d'effort et de coût du projet de maintenance et de la version planifiée de logiciel sont préparées conformément à une procédure documentée.
- [RBA2.02/06] Un processus de mise en oeuvre des nouvelles versions planifiées de logiciel [83] est institué.
- [RBA2.02/07] Les demandes de service ou *work orders* sont approuvés par l'utilisateur et par l'informatique.
- [RBA2.02/08] Un processus de planification annuelle de la gestion du logiciel [85] est institué.

- [GEA3.02/01] Un processus formel de planification de projets de maintenance, RMPT et versions planifiées de logiciel est documenté et vérifié d'une façon indépendante [25].
- [GEA3.02/02] Les efforts du personnel affecté au projet de maintenance ou à la version planifiée de logiciel sont rapportés et enregistrés à des fins de mesure.
- [GEA3.02/03] Une estimation en temps et en effort du projet de maintenance est effectuée par une équipe compétente [21].
- [GEA3.02/04] La planification des RMPT donne des indicateurs sur la charge de travail et de coûts qu'engendre la réalisation des RMPT dans une période de temps déterminée (annuelle par exemple).
- [GEA3.02/05] Les priorités utilisées pour planifier les RMPT sont flexibles et contrôlées par le client.
- [RBA3.02/06] Un processus formel de réalisation des demandes de service en attente est utilisé pour résorber les files d'attentes.
- [RBA3.02/07] Un processus de mise en oeuvre de la conception et de la programmation des nouvelles versions de logiciel [84] est institué.
- [RBA3.02/08] Le cycle annuel de support aux usagers reçoit des indicateurs d'étapes significatives du support, de sa progression et marque ainsi la différence entre la maintenance et les opérations.
- [RBA3.02/09] Les demandes des utilisateurs et de l'informatique non urgentes sont satisfaites par la mise en oeuvre de nouvelles versions planifiées de logiciel.
- [RBA3.02/10] Un processus de traitement des demandes de services [88] est institué.
- [RBA3.02/11] La mise à disposition de ressources adéquates pour atteindre les objectifs est assurée par le processus de budgétisation annuelle.
- [MCA3.02/12] Une liste détaillant la réalisation de projets de maintenance, RMPT et versions planifiées de logiciel est préparée conformément à une procédure documentée.
- [MCA3.02/13] Les engagements pris, dans le cadre du projet de maintenance, envers des personnes et des groupes externes à l'organisation, sont passés en revue par la Direction conformément à une procédure documentée.
- [MCA3.02/14] Un cycle de vie de projet de maintenance, avec phases prédéfinies de taille maîtrisable, est identifié ou défini.

- [MCA3.02/15] La planification du projet de maintenance est élaborée conformément à une procédure documentée.
- [MCA3.02/16] Des estimations de la taille des produits de travail du projet de maintenance ou de la version planifiée de logiciel sont préparées conformément à une procédure documentée.
- [MCA3.02/17] Les estimations de ressources informatiques critiques du projet de maintenance ou de la version planifiée de logiciel sont décrites conformément à une procédure documentée.
- [MCA3.02/18] Dans les cas de sous-traitance, une planification documentée de projet de maintenance logiciel, RMPT ou de versions planifiées de logiciel du sous-traitant est passé en revue et approuvé par le maître d'oeuvre.
- [MCA3.02/19] Un calendrier formel de réalisation de projets de maintenance, RMPT et versions planifiées de logiciel par type de maintenance [111] et par priorité est préparé conformément à une procédure documentée.
- [MCA3.02/20] Un plan de réalisation prévisionnelle périodique des RMPT est élaboré conformément à une procédure documentée.
- [MCA3.02/21] Le plan de charge prévisionnel des RMPT est documenté.
- [MCA3.02/22] La planification concernant les installations et les outils de soutien d'ingénierie logiciel pour le projet de maintenance ou pour la version planifiée de logiciel est préparée.
- [MCA3.02/23] Les données de planification de projets de maintenance, RMPT et versions planifiées de logiciel sont enregistrées à des fins de mesure.
- [MCA3.02/24] Les données de planification prévisionnelle sont enregistrées à des fins de mesure.
-
- [MCA4.02/01] Les risques associés aux coûts, aux ressources, aux délais et aux aspects techniques des projets de maintenance, RMPT et versions planifiées de logiciel sont identifiés, évalués et documentés.
- [RBA4.02/02] Les phases de planification et de conception appliquent rigoureusement les standards de développement de l'informatique de l'organisation.
- [SBA4.02/03] La planification des projets de maintenance et des versions planifiées de logiciel est régulière et rigoureuse.
- [RBA4.02/04] Les outils logiciels d'aide au processus de la conception et de la programmation des nouvelles versions de logiciel [84] existent et sont mis en oeuvre.

- [GEA5.02/01] Les processus de planification s'appuient sur des modèles de productivité spécifiques à la maintenance.
- [GEA5.02/02] La qualité des modèles de productivité spécifiques à la maintenance est connue et documentée.
- [GEA5.02/03] Les modèles de productivité spécifiques à la maintenance sont mis à jour régulièrement à des fins d'amélioration.
-

Suivi et supervision de la maintenance logiciel [107]

(Secteur-clé 03)

-
- [MCO.03/01] Les résultats et les performances des projets de maintenance, RMPT et versions planifiées de logiciel sont comparés aux données de leur planification.
- [MCO.03/02] Des actions correctrices lorsque les résultats et les performances réels s'écartent de façon significative de la planification des projets de maintenance, RMPT et versions planifiées de logiciel sont réalisées.
- [MCO.03/03] Les groupes et les personnes concernés acceptent les changements apportés aux engagements du projet de maintenance, RMPT et versions planifiées de logiciel.
- [MCO.03/04] Les clients qui sollicitent des requêtes de maintenance sont satisfaits du suivi et de la réalisation de ces requêtes de maintenance.
-
- [MCA2.03/01] Les engagements de projet de maintenance, RMPT et versions planifiées de logiciel envers des personnes et des groupes externes à l'organisation ainsi que les changements apportés à ces engagements sont passés en revue avec la Direction conformément à une procédure documentée.
- [MCA2.03/02] Les changements approuvés aux engagements et affectant le projet de maintenance, RMPT et versions planifiées de logiciel sont communiqués à la Direction, aux gestionnaires de maintenance et aux équipes de réalisation de la maintenance.
- [MCA2.03/03] Les ressources informatiques critiques, en effort et en coûts, ainsi que les délais, des projets de maintenance, RMPT et versions planifiées de logiciel, font l'objet d'un suivi et des actions correctrices sont prises si nécessaires.
- [MCA2.03/04] Les gestionnaires de la maintenance supervisent les revues internes périodiques à des fins de suivi de l'avancement technique, de la performance par rapport à leur planification.
- [MCA2.03/05] Des revues formelles portant sur les réalisations et les résultats du projet de maintenance, RMPT et versions planifiées de logiciel sont effectuées à des jalons déterminés du projet de maintenance, RMPT et versions planifiées de logiciel, conformément à une procédure documentée.
- [MCA2.03/06] Une planification de projet de maintenance, RMPT et versions planifiées de logiciel, documentée est utilisée pour effectuer le suivi des activités de maintenance et

communiquer l'état des travaux.

[RBA2.03/07] L'informatique ou les utilisateurs sont autorisés à traiter une demande de modification de logiciel ou de services de support.

[RBA2.03/08] Les informations contenues dans un bon de travail [06] sont connues.

[RBA2.03/09] Les demandes en maintenance logiciel soumises par les utilisateurs sont classées par catégorie de maintenance [08].

[RBA2.03/10] Les critères de justification d'une demande de modification [14] sont bien connus.

[RBA2.03/11] Un statut du bon de travail [106] est institué pour situer le cheminement d'une requête de maintenance.

[RBA2.03/12] Un processus de réalisation de la modification est institué [87].

-

[MCA3.03/01] La planification de projet de maintenance, RMPT et versions planifiées de logiciel est révisée conformément à une procédure documentée.

[MCA3.03/02] Les données réelles de mesure ainsi que les données de révision de la planification pour le projet de maintenance, RMPT et versions planifiées de logiciel, sont enregistrées.

[RBA3.03/03] Un processus formel de bon de travail existe.

[RBA3.03/04] L'utilisation du bon de travail fait l'objet d'une vérification.

[RBA3.03/05] Le processus de traitement des demandes de service [88] est connu et suivi.

[RBA3.03/06] Le processus de traitement des demandes approuvées [89] existe et est formellement utilisé.

[RBA3.03/07] Une coordination de toutes les demandes de modification est instituée.

-

[GEA4.03/01] Une analyse quantitative périodique de l'évolution des charges de travail et des niveaux de performance s'effectue.

[MCA4.03/02] Les risques associés aux coûts, aux ressources, aux délais et aux aspects techniques du projet de maintenance, RMPT et versions planifiées de logiciel, sont identifiés et font l'objet d'un suivi.

Gestion de la sous-traitance [27]

(Secteur-clé 04)

-
- [GEO.04/01] Le maître d'oeuvre gère la performance du sous-traitant.
- [GEO.04/02] Optimiser le service à la clientèle de maintenance aux meilleurs coûts et niveaux de service.
- [MCO.04/03] Le choix de sous-traitants [10] qualifiés de maintenance logiciel est de la responsabilité du maître d'oeuvre [52].
- [MCO.04/04] Un accord existe entre le maître d'oeuvre et le sous-traitant sur leurs engagements respectifs réciproques
- [MCO.04/05] Une communication constante et de bonne qualité est maintenue entre le maître d'oeuvre et le sous-traitant en maintenance [103].
-
- [GEA2.04/01] Le gestionnaire de maintenance [28] du maître d'oeuvre met à la disposition du sous-traitant l'ensemble des informations et outils nécessaires à la réalisation du projet de maintenance, RMPT ou versions planifiées de logiciel.
- [MCA2.04/02] Le travail à sous-traiter est défini et planifié conformément à une procédure documentée.
- [MCA2.04/03] Le sous-traitant en matière de maintenance de logiciel est sélectionné en fonction des résultats de l'évaluation de sa capacité de réalisation des travaux pour le contrat de sous-traitance conformément à une procédure documentée .
- [MCA2.04/04] L'entente contractuelle entre le maître d'oeuvre et le sous-traitant de maintenance logiciel sert de base à la gestion du contrat de sous-traitance.
- [MCA2.04/05] Les changements apportés à l'énoncé des travaux du sous-traitant de maintenance logiciel, aux clauses du contrat de sous-traitance et à tout autre engagement sont traités conformément à une procédure documentée.
- [MCA2.04/06] Le gestionnaire de maintenance du maître d'oeuvre effectuent des revues périodiques de l'état et de la coordination de travaux, en collaboration avec les gestionnaires du sous-traitant.
- [MCA2.04/07] Des revues et des échanges techniques périodiques ont lieu avec le sous-traitant de maintenance.

- [MCA2.04/08] Le gestionnaire de maintenance du maître d'oeuvre effectue le suivi des activités de gestion de configuration du sous-traitant conformément à une procédure documentée.
- [MCA2.04/09] Le gestionnaire de maintenance du maître d'oeuvre effectue des tests d'acceptation dans le cadre de la livraison de produits du sous-traitant conformément à une procédure documentée.
- [MCA2.04/10] La performance du sous-traitant est évaluée sur une base périodique et cette évaluation est passée en revue avec le sous-traitant.
- [RBA2.04/11] Les éléments clés des appels d'offre, les clauses de performance, les plans d'opérations et les plans de maintenance sont étudiés.
-
-
- [MCA3.04/01] Les revues formelles portant sur les réalisations du sous-traitant sont effectuées à certains jalons déterminés du projet de maintenance, RMPT ou versions planifiées de logiciel, conformément à une procédure documentée.
- [MCA3.04/02] L'équipe de maintenance logiciel du maître d'oeuvre effectue le suivi des activités d'assurance-qualité du sous-traitant conformément à une procédure documentée.
- [MCA3.04/03] Une planification documentée et approuvée de projet de maintenance, RMPT ou de versions planifiées de logiciel du sous-traitant est utilisée aux fins de suivi des activités de maintenance logiciel et de communication de l'état des travaux.

Assurance-qualité logiciel [04]*(Secteur-clé 05)*

-
- [MCO.05/01] Les activités d'assurance-qualité pour la maintenance logiciel sont planifiées.
- [MCO.05/02] Le respect des normes, les procédures et exigences par les produits de travail et les activités de maintenance logiciel est objectivement vérifié.
- [MCO.05/03] Les groupes et les personnes concernés sont informés des activités et des résultats de l'assurance-qualité pour la maintenance logiciel.
- [MCO.05/04] Les problèmes de non conformité ne pouvant être résolus dans le cadre du projet de maintenance logiciel, de RMPT ou de versions planifiées de logiciel sont pris en charge par la Direction.
-
- [MCA3.05/01] Un plan d'assurance-qualité pour la maintenance logiciel est préparé pour le projet de maintenance conformément à une procédure documentée.
- [MCA3.05/02] Les activités de maintenance concernant l'assurance-qualité pour la maintenance logiciel sont réalisées conformément au plan d'assurance-qualité pour la maintenance.
- [MCA3.05/03] Le personnel affecté à l'assurance-qualité pour la maintenance logiciel collabore à la préparation et à la revue du plan de maintenance logiciel, des normes et des procédures du projet de maintenance logiciel ou de la version planifiée de logiciel.
- [MCA3.05/04] Le personnel affecté à l'assurance-qualité pour la maintenance logiciel passe en revue les activités de maintenance logiciel afin de vérifier leur conformité.
- [MCA3.05/05] Le personnel affecté à l'assurance-qualité pour la maintenance logiciel effectue un audit des produits de travail désignés afin de vérifier leur conformité.
- [MCA3.05/06] Le personnel affecté à l'assurance-qualité pour la maintenance logiciel rend compte du résultat de ses activités aux gestionnaires de la maintenance des systèmes périodiquement.
- [MCA3.05/07] Les déviations repérées dans les activités de maintenance logiciel et dans les produits de travail de maintenance sont documentées et traitées conformément à une procédure documentée.
- [MCA3.05/08] Le personnel affecté à l'assurance-qualité pour la maintenance logiciel conduit des

revues périodiques de ses activités et de ses constats avec les gestionnaires de la maintenance et avec le client, au besoin.

- [RBA3.05/09] La direction s'engage à réaliser des résolutions étroitement liées à la gestion du personnel de maintenance.
- [RBA3.05/10] Les gestionnaires de maintenance expriment clairement que leur personnel n'est pas juste en charge de la «maintenance du code», mais en charge de responsabilités qui contribuent à la gestion d'actifs importants de l'organisation.
- [RBA3.05/11] Les facteurs clés de satisfaction du personnel de maintenance logiciel [24] sont identifiés.
- [SBA3.05/12] Un lien important d'ordre organisationnel, opérationnel, stratégique, relie la tâche de maintenance et le portefeuille d'applications.
- [SBA3.05/13] La fonction de maintenance logiciel et ses réalisations sont reconnues et valorisées.
- [SBA3.05/14] La promotion de l'interdépendance entre l'informatique et l'utilisateur est assurée.
-
-
- [SBA4.05/01] Des politiques spécifiques à la maintenance logiciel [70] dans l'organisation sont implantées et rigoureusement vérifiées.

Gestion de la configuration [26]

(Secteur-clé 06)

-
- [MCO.06/01] Les activités de gestion de configuration sont planifiées.
- [MCO.06/02] Les produits de travail [90] sélectionnés sont contrôlés et disponibles.
- [MCO.06/03] Les changements apportés aux produits de travail identifiés sont contrôlés
- [MCO.06/04] Les groupes et les personnes concernés par la gestion de configuration sont informés de l'état et du contenu des référentiels [96].
-
- [MCA2.06/01] Un plan de gestion de configuration est préparé pour chaque réalisation de projet de maintenance ou de versions planifiées de logiciel conformément à une procédure documentée.
- [MCA2.06/02] Un plan de gestion de configuration documenté et approuvé est utilisé comme base pour réaliser les activités de gestion de configuration .
- [MCA2.06/03] Les produits de travail de maintenance à placer sous gestion de configuration sont identifiés.
- [MCA2.06/04] Les changements apportés aux référentiels sont contrôlés conformément à une procédure documentée.
- [MCA2.06/05] Les produits sont créés à partir de la bibliothèque de référentiels et leur lancement est contrôlé conformément à une procédure documentée.
-
-
- [MCA3.06/01] Des rapports standards documentant les activités de gestion de configuration ainsi que le contenu du référentiel sont élaborés et rendus disponibles aux personnes et aux groupes concernés.
- [MCA3.06/02] Un système de bibliothèque de gestion des configurations est établi comme dépôt pour stocker les référentiels.
- [MCA3.06/03] Les demandes de changement ainsi que les rapports de problèmes de configuration sont déclenchés, enregistrés, passés en revue, approuvés et soumis à un suivi conformément à une procédure documentée.

[MCA3.06/04] Les audits des référentiels sont effectués conformément à une procédure documentée.

[RBA3.06/05] Les procédures de contrôle et de changement de configuration respectent des normes de configuration [55] et sont explicites.

[RBA3.06/06] Des outils de gestion de configuration [59] existent.

[RBA3.06/07] Les responsabilités de gestion de configuration sont attribuées de façon claire et connue de tous.

(Secteur-clé 07)

-
- [RBO.07/01] Un processus de transition du logiciel du développement initial vers la maintenance existe formellement.
- [RBO.07/02] Le logiciel transféré à la responsabilité de la maintenance est de qualité, complet et prêt selon les standards de l'acceptation du logiciel.
- [RBO.07/03] L'utilisateur est assuré que le logiciel qu'on lui livre répond bien à ses besoins et qu'il en est satisfait.
-
- [RBA2.07/01] Un processus formel d'acceptation de logiciel [76] est institué.
- [RBA2.07/02] Les activités de préparation de réception du logiciel [02] par la maintenance sont sous contrôle.
- [RBA2.07/03] Une transition "douce" mais rigoureuse du développement vers la maintenance est assurée.
- [RBA2.07/04] Le logiciel transféré vers la maintenance est soumis à l'analyse et à la vérification [03].
- [RBA2.07/05] L'objectif de la qualité doit être planifié par avance et est évalué tout au long du cycle de vie du développement.
- [RBA2.07/06] L'utilisateur, les opérations et le support sont bien inclus dans la réalisation des objectifs de l'acceptation du logiciel.
- [RBA2.07/07] Lors de l'acceptation de logiciel, les besoins pour le support [05] sont entièrement satisfaits.
- [RBA2.07/08] La disponibilité du personnel est assurée lors de l'acceptation du logiciel.
- [RBA2.07/09] La mise en oeuvre de règles et de procédures de fonctionnement du support est effective lors de l'acceptation de logiciel.
- [RBA2.07/10] Lors des tests d'acceptation de logiciel [108] par l'utilisateur, la fonction de support donne entièrement satisfaction.
- [RBA2.07/11] Une phase de post-acceptation de logiciel [71] est instituée pour préparer les plans d'amélioration de logiciel.

[RBA2.07/12] Le plan d'amélioration de logiciel [64] est documenté et suivi.

[RBA2.07/13] Un processus de mise en oeuvre de l'acceptation de logiciel existe.

[RBA2.07/14] Les gestionnaires, les utilisateurs et le personnel de maintenance reçoivent régulièrement et à temps les rapports de post-audit [92].

-

[RBA3.07/01] Une phase de post-audit est instituée.

[RBA3.07/02] Un rapport de post-audit est obligatoire.

[RBA3.07/03] Le personnel impliqué dans la phase de post-audit ne doit pas et n'a pas été membre du développement.

[RBA3.07/04] Le personnel de support possède les critères d'acceptation du logiciel et l'autorité finale de l'acceptation du produit vers la maintenance.

[RBA3.07/05] Dans le processus d'acceptation de logiciel, une phase de pré-acceptation de logiciel [72] est instituée.

-

[RBA4.07/01] Lors de l'acceptation du logiciel, les produits livrés sont validés selon les standards du développement [104].

[RBA4.07/02] La facilité de maintenance logiciel [23] ou *maintenability* est mesurée et contrôlée.

-

[RBA5.07/01] La satisfaction de l'utilisateur [101] après la phase d'acceptation de logiciel est mesurée.

Gestion des problèmes [34]

(Secteur-clé 08)

-
- [GEO.08/01] Un processus formel encadre la gestion de problèmes.
- [GEO.08/02] Les temps d'interruption de systèmes sont réduits au minimum pour réduire les coûts.
-
- [RBA2.08/01] Des solutions aux problèmes provenant d'opérations ou d'interruptions de système sont signalées à temps et efficacement.
- [RBA2.08/02] Un processus de gestion des problèmes [81] est institué pour assurer des solutions rapides et efficaces aux interruptions.
-
- [GEA3.08/01] Les impacts sur la productivité du logiciel [40] sont analysés.
- [RBA3.08/02] Des règles et procédures de gestion des problèmes [97] existent et sont rigoureusement appliquées.
- [RBA3.08/03] L'information nécessaire pour prévenir la récurrence et pour gérer les tendances est fournie efficacement par le processus de gestion des problèmes.
- [RBA3.08/04] Le processus de gestion des problèmes est classé en catégories qui sont diffusées et déployées.
- [RBA3.08/05] Les mesures et le *reporting* des problèmes [53] dans le processus de gestion des problèmes sont diffusées et déployées.
- [RBA4.08/06] Un processus de *monitoring* des événements critiques [55] est institué.
- [RBA3.08/07] La priorité de la résolution des problèmes est dépendante du *monitoring* des événements critiques.
- [RBA3.08/08] Les résultats du *monitoring* de la performance basés sur l'analyse des événements critiques sont fournis par le personnel affecté à la gestion du logiciel sur une base périodique.
-

[RBA4.08/01] La corrélation entre la performance et les bénéfices du système est assurée par le processus de gestion des problèmes.

Gestion du support aux usagers [36]

(Secteur-clé 09)

[SBO.09/01] Un processus de gestion du support fourni aux usagers est mis en place.

[RBO.09/02] Un niveau de service satisfaisant en matière de support aux usagers est fourni.

[RBA2.09/01] La fonction de support dispose de toutes les ressources nécessaires pour accomplir sa tâche.

[RBA2.09/02] Les mesures et le *reporting* de la fonction de support aux usagers [54] sont utilisés efficacement pour les besoins de la gestion.

[RBA2.09/03] La révision des activités de support dans le processus de planification annuelle est implantée et bien suivie.

[RBA2.09/04] Un processus de budgétisation de la fonction de support [79] est institué.

[RBA2.09/05] Le budget de la fonction de support aux usagers [07] fait l'objet d'une étude.

[RBA2.09/06] Les budgets de la fonction de support aux usagers sont classés par catégories.

[RBA2.09/07] Un processus de gestion de la performance de la fonction de support aux usagers [80] existe.

[RBA2.09/08] Un rapport annuel de support aux usagers [93] est diffusé.

[RBA2.09/09] Dans le cadre de la gestion de la performance, les responsabilités sont clairement définies.

[RBA2.09/10] Les cibles de performance sont réalisées par une action commune des utilisateurs, des opérations, de l'informatique et éventuellement d'autres sources également.

[RBA2.09/11] Des rapports périodiques aux gestionnaires de l'informatique et aux utilisateurs-gestionnaires sont fournis par la gestion de la performance.

[RBA2.09/12] La révision de la performance fournit un rapport régulier aux comités de gestion de logiciel.

[GEA3.09/01] Le personnel de maintenance chargé d'estimer une tâche de support [22] dispose des compétences nécessaires pour une estimation optimale.

[RBA3.09/02] Le cycle annuel de support [16] dispose d'un budget et d'un plan annuel de support [63].

[RBA3.09/03] L'utilisateur et le gestionnaire de l'informatique révisent conjointement le rapport fourni par le cycle annuel de support pour préparer le plan de la prochaine année.

[RBA3.09/04] Le processus de planification annuelle du support [86] assure à l'utilisateur un support efficace.

[RBA3.09/05] Les changements apportés au système sont décrits avec détail dans un rapport périodique de support aux usagers.

[RBA3.09/06] Les services en support aux usagers [102] sont offerts selon des niveaux de services [56].

[RBA4.09/01] Un processus de facturation interne effective est à la base du choix des niveaux de services offerts aux usagers.

[RBA4.09/02] Les budgets de support aux usagers sont entre les mains des utilisateurs et négociés entre les utilisateurs et l'informatique.

Organisation de la fonction de maintenance logiciel [59]

(Secteur-clé 10)

[SBO.10/01] La fonction de maintenance oeuvre pour rendre les portefeuilles d'applications plus performants.

[GEA2.10/01] Tous les processus de gestion de maintenance sont détaillés en classe de maintenance [11].

[RBA2.10/02] Des comités de gestion de logiciel [15] sont institués.

[RBA2.10/03] Des réunions des comités de gestion de logiciel se tiennent dans le cadre du processus de gestion de logiciel.

[RBA2.10/04] Un rapport de réunion de comité de gestion de logiciel [94] est obligatoirement diffusé.

[RBA2.10/05] Un cycle de vie du logiciel [17] est considérée comme base de la gestion du logiciel.

[SBA2.10/06] Les demandes de maintenance sont faites formellement, même après les faits, pour être enregistrées et rapportées.

[SBA2.10/07] La responsabilité des tâches est partagée entre les utilisateurs et l'informatique.

[GEA3.10/01] Le budget alloué à la maintenance du logiciel est réparti selon les processus de gestion de la maintenance et détaillé selon chaque classe de maintenance.

[GEA3.10/02] Un niveau de service précis est établi pour chaque processus de gestion de la maintenance et chaque classe de maintenance;

[GEA3.10/03] Les critères d'assignation à un processus de gestion de la maintenance est clairement défini.

[SBA3.10/04] Des politiques spécifiques à la gestion des ressources humaines affectées à la maintenance sont établies.

—

- [SBA4.10/01] Un processus de mesures des tâches de maintenance logiciel [82] existe.
- [SBA4.10/02] La tâche de maintenance logiciel est perçue dans un contexte de portefeuille d'application pris dans son ensemble.
- [SBA5.10/01] Des techniques de gestion et modèles de recherche sont utilisés pour soutenir la planification et l'établissement de politiques spécifiques à la maintenance logiciel.
- [SBA5.10/02] Des approches d'innovations et de changement en matière organisationnelle sont encouragées.
-

Rôles des intervenants autour de la maintenance [100]

(Secteur-clé 11)

-
- [GEO.11/01] Chaque intervenant dans les activités de maintenance connaît son rôle et ses responsabilités.
-
- [RBA2.11/01] Le personnel de maintenance communique l'état et les résultats des travaux à la supervision et aux utilisateurs.
- [RBA2.11/02] Le personnel de maintenance identifie et réalise les améliorations techniques.
- [RBA2.11/03] Le personnel de maintenance recommande, conseille et forme les utilisateurs dans l'utilisation optimale des logiciels.
- [RBA2.11/04] Le personnel de maintenance dispose de la compétence nécessaire dans le domaine d'application et en programmation.
- [RBA2.11/05] Le personnel de l'informatique fournit du service pour contribuer au succès de la fonction de support aux usagers.
- [RBA2.11/06] L'utilisateur participe obligatoirement aux spécifications des nouveau besoins.
- [RBA2.11/07] La requête de maintenance émise est approuvée obligatoirement par l'utilisateur.
- [RBA2.11/08] La priorité de la réalisation d'une requête de maintenance est approuvée par l'utilisateur.
- [RBA2.11/09] Les tests d'acceptation de la réalisation de la requête de maintenance nécessitent la participation de l'utilisateur.
- [RBA2.11/10] La mise à jour des manuels d'utilisateurs nécessite la participation de l'utilisateur.
- [RBA2.11/11] En matière d'activités de maintenance, le rôle de l'utilisateur [99] est clairement défini.
- [RBA2.11/12] Une grille des responsabilités [39] est mise en oeuvre pour déterminer le rôle des intervenants en matière de maintenance.
-
- [RBA3.11/01] Le personnel de maintenance élabore et maintient les standards techniques et de qualité.

- [RBA3.11/02] Le personnel de maintenance est responsable des demandes *ad hoc* de l'utilisateur, des corrections du systèmes nécessitant l'urgence, *etc.*
- [RBA3.11/03] Le personnel de maintenance fournit un dossier détaillé décrivant les traces du travail complété.
- [RBA3.11/04] Le personnel de maintenance communique avec diligence les problèmes et les résultats aux utilisateurs et aux gestionnaires de la maintenance.
- [RBA3.11/05] Le personnel de maintenance accepte les attributions de tâches et d'affectation de fonction basées sur le mérite en matière d'affaires et non pas sur le mérite en matière technique.
- [RBA3.11/06] Le personnel de maintenance dispose de connaissances appropriées dans le champ de compétence de l'utilisateur.
- [RBA3.11/07] Le personnel de maintenance est informé sur les fonctions d'affaires de l'utilisateur.
-

Priorité organisationnelle sur les processus [74]

(Secteur-clé 12)

-
- [MCO.12/01] Les activités de développement et d'amélioration du processus de la maintenance logiciel sont coordonnées à travers toute l'organisation.
- [MCO.12/02] Les forces et les faiblesses du processus de maintenance logiciel utilisées sont identifiées par rapport à un standard de processus.
- [MCO.12/03] Les activités de développement et d'amélioration du processus à l'échelle de l'organisation sont planifiées.
-
- [MCA3.12/01] Le processus de maintenance logiciel est évalué périodiquement et des plans d'action sont dressés en fonction des constats d'évaluation.
- [MCA3.12/02] L'organisation développe et maintient un plan pour le développement et l'amélioration de son processus de maintenance logiciel.
- [MCA3.12/03] Les activités de l'organisation et des projets pour développer et améliorer leurs processus de maintenance logiciel sont coordonnées au niveau de l'organisation.
- [MCA3.12/04] Les nouveaux processus, méthodes et outils utilisés de façon restreinte dans l'organisation font l'objet d'un suivi, sont évalués et, au besoin, sont transférés à d'autres secteurs de l'organisation.
- [MCA3.12/05] La formation couvrant les processus de maintenance logiciel de l'organisation est coordonnée à l'échelle de l'organisation.
- [MCA3.12/06] Les groupes impliqués dans la mise en oeuvre du processus de maintenance logiciel sont informés des activités de l'organisation concernant le développement et l'amélioration du processus de maintenance logiciel.
-
-
- [MCA4.12/01] L'utilisation des données historiques du processus de maintenance logiciel de l'organisation est coordonnée au niveau de l'organisation.
-

Définition du processus de l'organisation [18]*(Secteur-clé 13)*

-
- [MCO.13/01] Un processus standard pour l'organisation est développé et maintenu.
- [MCO.13/02] L'information sur l'application du processus standard de l'organisation dans les projets de maintenance ou les versions planifiées de logiciel est recueillie, passée en revue et rendue disponible.
-
- [MCA3.13/01] Le processus standard de l'organisation est développé et maintenu conformément à une procédure documentée.
- [MCA3.13/02] Le processus standard de l'organisation est documenté conformément aux normes organisationnelles établies.
- [MCA3.13/03] Les descriptions des cycles de vie dont l'utilisation a été approuvée pour les projets de maintenance ou pour les versions planifiées de logiciel sont documentées et maintenues.
- [MCA3.13/04] Des lignes directrices et des critères d'adaptation par la maintenance de type "projet de maintenance" et "versions planifiées de logiciel" du processus standard de l'organisation sont développés et maintenus.
- [MCA3.13/05] Les données historiques du processus de l'organisation sont traitées et maintenues.
- [MCA3.13/06] Une bibliothèque contenant la documentation se rapportant aux processus est mise en place et maintenue.
-

Gestion de la formation [30]

(Secteur-clé 14)

-
- [MCO.14/01] Les activités de formation sont planifiées.
- [MCO.14/02] Chaque type de maintenance logiciel développe et maintient un plan de formation qui précise ses besoins de formation.
- [MCO.14/03] La formation pour développer les connaissances et les compétences nécessaires pour jouer les rôles de gestion et de technique du logiciel est fournie.
- [MCO.14/04] Le personnel d'ingénierie et le personnel de maintenance reçoivent une formation leur permettant de jouer leurs rôles.
-
- [RBA2.14/01] Les gestionnaires de maintenance offrent à leur personnel une formation adéquate.
- [GEA2.14/02] Les gestionnaires de maintenance établissent des choix de gestion en matière de formation [09].
- [SBA2.14/03] Le personnel de maintenance détient de bonnes connaissances des systèmes, de la programmation et de l'environnement technique;
- [SBA2.14/04] Le personnel de maintenance a des connaissances variées et une expérience reliée au portefeuille d'applications;
-
-
- [MCA3.14/01] La formation au sein de l'organisation est dispensée conformément au plan de formation de l'organisation.
- [MCA3.14/02] Les cours de formation préparés au niveau de l'organisation sont développés et maintenus conformément aux normes de l'organisation.
- [MCA3.14/03] Les dossiers de formation sont maintenus.
- [MCA3.14/04] Le plan de formation de l'organisation est développé et révisé conformément à une procédure documentée.
- [RBA3.14/05] Un processus de test des nouvelles versions planifiées de logiciel ou d'un nouveau logiciel est institué.

- [RBA3.14/06] Des outils de tests des nouvelles versions planifiées de logiciel ou d'un nouveau logiciel [61] existent et sont utilisés.
- [RBA3.14/07] Un type de formation des utilisateurs [110] pour chaque type de maintenance est défini.
- [SBA3.14/08] Des systèmes d'information assurent le suivi de la formation.
- [SBA3.14/09] Une politique d'assignation de ressources aux tâches de maintenance [69] existe et est respectée.
- [SBA3.14/10] Les critères d'assignation d'une ressource à une tâche de maintenance [13] sont connus.
- [SBA3.14/11] La politique de spécialisation des ressources humaines en maintenance est recherchée.
- [SBA3.14/12] Il est procédé à l'embauche de consultants extérieurs pour diversifier les vues et élargir la perspective professionnelle.
- [SBA3.14/13] La Direction informatique assure un investissement important dans le développement de son nouveau personnel à travers une formation continue.
- [SBA3.14/14] La sélection du personnel de maintenance inclut le recrutement sur la base d'une formation large, orientée affaires plutôt qu'un profil de formation limitée orienté informatique.
- [SBA3.14/15] Il existe un «espace de l'échec» limité et encadré pour donner la possibilité d'expérimenter.
- [SBA3.14/16] L'informatique fournit des programmes d'expérimentation à l'adresse des utilisateurs.
- [SBA3.14/17] Les connaissances des applications par les utilisateurs sont périodiquement évaluées.
- [SBA3.14/18] Les données d'échecs de réalisation de la maintenance sont utilisées pour modifier le processus standard de la maintenance.
-
-
- [SBA4.14/01] Les besoins en formation sont clairement identifiés et entièrement satisfaits;
- [SBA4.14/02] Un entraînement formel pour le personnel de maintenance est institué. Il permet de partager la connaissance du système au delà des limites de la documentation.
- [SBA4.14/03] La responsabilité de la maintenance de systèmes stratégiques est une des stratégies suggérée pour la formation de l'équipe de maintenance

Gestion intégrée de la maintenance [37]*(Secteur-clé 15)*

-
- [MCO.15/01] Le processus défini de maintenance du projet de maintenance ou de la version planifiée de logiciel est une version adaptée du processus standard de l'organisation.
- [MCO.15/02] Le projet de maintenance ou la version planifiée de logiciel est planifié et géré conformément au processus défini de maintenance du projet de maintenance ou de la version planifiée de logiciel.
-
- [MCA3.15/01] Le processus défini de maintenance est développé en adaptant le processus standard de l'organisation conformément à une procédure documentée.
- [MCA3.15/02] Chaque processus défini de maintenance du projet de maintenance ou de la version planifiée de logiciel est révisé conformément à une procédure documentée.
- [MCA3.15/03] Le plan du projet de maintenance ou de la version planifiée de logiciel, qui décrit l'utilisation du processus défini de maintenance, est développé et révisé conformément à une procédure documentée.
- [MCA3.15/04] Le projet de maintenance ou la version planifiée de logiciel est géré conformément au processus défini de maintenance.
- [MCA3.15/05] Les données du processus de maintenance de l'organisation est utilisée pour la planification et l'estimation de la maintenance.
- [MCA3.15/06] L'effort et les coûts du projet de maintenance ou de la version planifiée de logiciel sont gérés conformément à une procédure documentée.
- [MCA3.15/07] Les ressources informatiques critiques du projet de maintenance ou de la version planifiée de logiciel sont gérées conformément à une procédure documentée.
- [MCA3.15/08] Les dépendances critiques ainsi que les chemins critiques du calendrier du projet de maintenance ou de la version planifiée de logiciel sont gérés conformément à une procédure documentée.
- [MCA3.15/09] Les risques du projet de maintenance ou de la version planifiée de logiciel sont identifiés, évalués, documentés et gérés conformément à une procédure documentée.

[MCA3.15/10] Le projet de maintenance ou la version planifiée de logiciel est périodiquement passé en revue afin de déterminer les actions nécessaires pour que la performance et les résultats

du projet de maintenance ou de la version planifiée de logiciel s'harmonisent aux besoins actuels et projetés de l'entreprise, du client et des utilisateurs finals, selon le cas.

[MCA4.15/01] La taille des produits de travail de maintenance ou l'ampleur des changements qui y ont été apportés est gérée conformément à une procédure documentée.

(Secteur-clé 16)

-
- [MCO.16/01] Les tâches d'ingénierie sont définies, intégrées et exécutées de façon cohérente pour maintenir le logiciel.
- [MCO.16/02] Les produits de travail sont maintenus cohérents les uns par rapport aux autres.
-
- [GEA2.16/01] La documentation de développement et d'essais est mise à la disposition de l'équipe de maintenance;
- [GEA2.16/02] L'utilisateur possède un environnement de validation parallèle qui lui permet d'éprouver les changements et nouvelles versions;
-
- [GEA3.16/01] L'environnement technique ainsi que les outils spécialisés pour la maintenance sont documentés et à la disposition de chaque application en maintenance pour utilisation.
- [MCA3.16/02] La maintenance, la documentation et la vérification du code de maintenance sont réalisées, conformément au processus défini de maintenance du projet de maintenance ou de la version planifiée de logiciel, pour mettre en oeuvre les exigences et la conception de la maintenance.
- [MCA3.16/03] Les tests de maintenance sont effectués conformément au plan de test du projet de maintenance ou de la version planifiée de logiciel.
- [MCA3.16/04] Les tests d'intégration du logiciel sont planifiés et effectués conformément au plan de test du projet de maintenance ou de la version planifiée de logiciel.
- [MCA3.16/05] Les tests système et les tests d'acceptation de logiciel sont planifiés et effectués afin de démontrer que le logiciel satisfait formellement aux exigences énoncées.
- [MCA3.16/06] La documentation qui sera utilisée dans le cadre de l'exploitation et de la maintenance du logiciel est développée et maintenue conformément au processus défini de maintenance du projet de maintenance ou de la version planifiée de logiciel.
- [MCA3.16/07] La *tracability* et la cohérence sont maintenues entre les produits de travail de maintenance, y compris les plans de maintenance, les descriptions de processus, les exigences système, les exigences de maintenance, la conception, le code, les plans de tests, les procédures de tests, *etc.*
- [RBA3.16/08] Des logiciels d'outils de configuration [42] existent et sont utilisés.

- [RBA3.16/09] Des logiciels d'outils de gestion de données [44] existent et sont utilisés.
- [RBA3.16/10] Des logiciels d'outils de gestion de documentation [43] existent et sont utilisés.
- [RBA3.16/11] Des logiciels d'outils de test et de *debugging* [46] existent et sont utilisés.
- [RBA3.16/12] Des logiciels d'outils de gestion de code [45] existent et sont utilisés.
- [SBA3.16/13] Des outils standards et "calibrés" ainsi que des méthodes d'ingénierie logiciel sont utilisés d'une façon effective pour la maintenance des systèmes.
- [SBA3.16/14] Une politique de standard technologique d'ingénierie [105] de produits existe et est vérifiée.
- [SBA3.16/15] Des outils standards et "calibrés" d'analyse de programmes permettent de trouver les différentes pièces des systèmes, leur interconnexion, ou encore la structure et le flot des procédures, du contrôle et des données.
- [SBA3.16/16] La gestion des outils de maintenance est centralisée
-
-
- [MCA4.16/01] Des méthodes et outils d'ingénierie appropriés sont intégrés au processus défini de maintenance du projet de maintenance ou de la version planifiée de logiciel.
- [MCA4.16/02] Les données sur les défauts identifiés au cours des revues par les pairs et des tests sont recueillis et analysés conformément au processus défini de maintenance du projet de maintenance ou de la version planifiée de logiciel.

(Secteur-clé 17)

-
- [MCO.17/01] Les activités de gestion quantitative du processus sont planifiées.
- [MCO.17/02] La performance du processus défini de maintenance du projet de maintenance ou de la version planifiée de logiciel est contrôlée quantitativement.
- [MCO.17/03] La capacité du processus standard de maintenance de l'organisation est connue en termes quantitatifs.
-
- [GEA3.17/01] Les données concernant les efforts de maintenance sont accumulées d'une façon détaillée, validées et stockées;
-
- [GEA4.17/01] Des programmes de mesure de productivité et de qualité de services de la maintenance sont formellement utilisés;
- [GEA4.17/02] Des analyses de tendances par processus de gestion et classes de maintenance sont effectuées et servent à des fins de planification et de budgets;
- [GEA4.17/03] Des outils de simulation d'impact sont disponibles, connus, configurés et à la disposition de chaque application en maintenance pour utilisation;
- [GEA4.17/04] Les outils de performance utilisés par le centre de données sont disponibles, connus, configurés et à la disposition de chaque application en maintenance pour utilisation.
- [MCA4.17/05] Le plan de gestion quantitative de processus du projet de maintenance ou de la version planifiée de logiciel est élaboré conformément à une procédure documentée.
- [MCA4.17/06] Les activités de gestion quantitative de processus du projet de maintenance ou de la version planifiée de logiciel sont réalisées conformément au plan de gestion quantitative de processus du projet de maintenance ou de la version planifiée de logiciel.
- [MCA4.17/07] La stratégie de collecte de données et les analyses quantitatives à effectuer sont déterminées en fonction du processus défini de maintenance du projet de maintenance ou de la version planifiée de logiciel.
- [MCA4.17/08] Les données de mesure utilisées pour contrôler quantitativement le processus défini de maintenance du projet de maintenance ou de la version planifiée de logiciel sont recueillies conformément à une procédure documentée.

- [MCA4.17/09] Le processus défini de maintenance du projet de maintenance ou de la version planifiée de logiciel est analysé et placé sous contrôle quantitatif conformément à une procédure documentée.
- [MCA4.17/10] Des rapports documentant les résultats des activités de gestion quantitative de processus du projet de maintenance ou de la version planifiée de logiciel sont préparés et distribués.
- [MCA4.17/11] Le référentiel des capacités du processus standard de maintenance de l'organisation est établi et maintenu conformément à une procédure documentée.
- [SBA4.17/12] Une utilisation standard pour toutes les applications du portefeuille d'applications des dictionnaires de données, des analyseurs, des «restructeurs», des convertisseurs de codes automatisés, *etc.* est assurée.
-

(Secteur-clé 18)

-
- [MCO.18/01] Les activités de gestion de la qualité de la maintenance sont planifiées.
- [MCO.18/02] Des objectifs mesurables en termes de qualité des produits et la priorité de ces objectifs sont définis.
- [MCO.18/03] L'avancement réel vers l'atteinte de ces objectifs de qualité pour les produits est quantifié et géré.
- [RBO.18/04] L'état du logiciel et la progression de son amélioration sont clairement visibles et compréhensibles.
- [RBO.18/05] Publication des forces et faiblesses du logiciel à l'attention des gestionnaires de l'informatique et des utilisateurs.
- [RBO.18/06] Création d'un environnement proactif et créatif.
-
- [RBA2.18/01] Un outil de communication de l'état de la qualité des logiciels informe le management sur l'état de la qualité du logiciel et la progression de son amélioration.
- [RBA2.02/02] La stratégie d'implantations périodiques de nouvelles versions de systèmes est favorisée par l'organisation.
-
-
- [RBA3.18/01] Des plans d'amélioration de logiciel sont acceptés, réalisés et donnent des résultats satisfaisants.
- [RBA3.18/02] Les plans d'amélioration de logiciel sont toujours présentés en termes de bénéfices.
- [RBA3.18/03] Un processus d'amélioration de logiciel [77] est implanté.
- [RBA3.07/04] Un engagement formel de la Direction est nécessaire pour réaliser l'acceptation du logiciel.
-
-
- [GEA4.18/01] Les analyses de tendance de maintenance sont utilisées à des fins de sélection d'applications candidates à la refonte et/ou à la réingénierie, appelée aussi "maintenance ciblée".

- [MCA4.18/02] Le plan qualité du portefeuille d'applications en maintenance est développé et maintenu conformément à une procédure documentée.
- [MCA4.18/03] Le plan qualité du portefeuille d'applications en maintenance est la base des activités de gestion de la qualité du portefeuille d'applications.
- [MCA4.18/04] Les objectifs quantitatifs de qualité des produits pour le portefeuille d'applications en maintenance sont définis, font l'objet d'un suivi et sont révisés tout au long des activités.
- [MCA4.18/05] La qualité des produits issus du portefeuille d'applications en maintenance est mesurée, analysée et comparée aux objectifs quantitatifs de qualité des produits sur "événement".
- [MCA4.18/06] La réalisation des objectifs de qualité du projet de maintenance ou de la version planifiée de logiciel pour les produits est communiquée, de manière appropriée, aux sous-traitants qui fournissent des produits de maintenance au projet de maintenance ou à la version planifiée de logiciel.
- [RBA4.18/07] Une procédure d'analyse de situation de logiciel [75] existe.

(Secteur-clé 19)

-
- [MCO.19/01] Les activités de prévention des défauts sont planifiées.
- [MCO.19/02] Les causes communes des défauts sont recherchées et identifiées
- [MCO.19/03] Les causes communes des défauts sont classées selon leur priorité et systématiquement éliminées.
-
- [MCA2.19/01] Au début d'un projet de maintenance ou d'une version planifiée de logiciel, les membres de l'équipe de maintenance devant effectuer cette tâche se réunissent afin de préparer les activités de cette tâche ainsi que les activités de prévention des défauts correspondantes.
- [MCA2.19/02] Chaque équipe chargée de la coordination des activités de prévention des défauts se réunit sur une base périodique afin de passer en revue et de coordonner la mise en oeuvre des propositions d'action issues des réunions d'analyse causale.
-
-
- [MCA3.19/01] Le projet de maintenance ou la version planifiée de logiciel développe et maintient un plan de ses activités de prévention des défauts.
- [MCA3.19/02] Les réunions d'analyse causale se déroulent conformément à une procédure documentée.
- [MCA3.19/03] Des données de prévention des défauts sont documentées et font l'objet d'un suivi pour toutes les équipes de coordination des activités de prévention des défauts.
- [MCA3.19/04] Les révisions du processus standard de maintenance de l'organisation, découlant des actions de prévention des défauts, sont intégrées conformément à une procédure documentée.
-
-
- [MCA5.19/01] Les révisions du processus défini de maintenance, découlant des actions de prévention des défauts, sont intégrées conformément à une procédure documentée.
- [MCA5.19/02] Le personnel affecté à l'ingénierie et le personnel affecté à la maintenance logiciel reçoivent, sur une base périodique, des comptes rendus sur l'état et les résultats des activités de prévention des défauts réalisées dans le projet de maintenance ou dans la version planifiée de logiciel au sein de l'organisation.

Gestion des changements technologiques [33]

(Secteur-clé 20)

-
- [MCO.20/01] L'intégration des changements technologiques est planifiée.
- [MCO.20/02] Les nouvelles technologies sont évaluées pour déterminer leur impact sur la qualité et la productivité.
- [MCO.20/03] Les nouvelles technologies qui conviennent à l'organisation sont intégrées dans les pratiques courantes de l'organisation.
-
- [MCA2.20/01] L'organisation développe et maintient un plan de gestion des changements technologiques.
- [MCA2.20/02] Les responsables logiciel et le personnel technique sont tenus au courant des nouvelles technologies.
- [MCA2.20/03] Des projets pilote d'amélioration technologique sont entrepris au besoin avant l'introduction d'une nouvelle technologie dans les pratiques courantes.
-
-
- [MCA3.20/01] La fonction de gestion des changements technologiques de l'organisation identifie avec les activités de maintenance les domaines se prêtant à de tels changements technologiques.
- [MCA3.20/02] La fonction de gestion des changements technologiques pour l'organisation analyse systématiquement le processus standard de maintenance de l'organisation pour identifier les domaines nécessitant une nouvelle technologie ou susceptibles d'en tirer parti.
- [MCA3.20/03] Les technologies sont choisies et acquises pour l'organisation et les projets de maintenance conformément à une procédure documentée.
- [MCA3.20/04] Les nouvelles technologies qui conviennent à l'organisation sont intégrées au processus standard de maintenance de l'organisation conformément à une procédure documentée.
-

Gestion des améliorations du processus [32]

(Secteur-clé 21)

-
- [MCO.21/01] L'amélioration continue de processus est planifiée.
- [MCO.21/02] Les activités d'amélioration du processus de l'organisation impliquent une participation généralisée.
- [MCO.21/03] Le processus standard de maintenance de l'organisation et les processus définis de maintenance des projets de maintenance ou des versions planifiées de logiciel font l'objet d'une amélioration continue.
-
- [MCA2.21/01] L'organisation développe et maintient un plan d'amélioration du processus de maintenance conformément à une procédure documentée.
- [MCA2.21/02] Les activités d'amélioration du processus de maintenance sont réalisées conformément au plan d'amélioration du processus de maintenance.
- [MCA2.21/03] Les membres de l'organisation participent activement, en équipes, au développement d'améliorations du processus pour les domaines qui leur ont été confiés.
- [MCA2.21/04] Des dossiers d'activités d'amélioration du processus de maintenance sont maintenus.
- [MCA2.21/05] Les responsables logiciel et le personnel technique reçoivent, sur événement, des comptes rendus sur l'état et les résultats des activités d'amélioration de processus de maintenance.
-
-
- [MCA3.21/01] Un programme d'amélioration du processus est établi et accorde aux membres de l'organisation le pouvoir d'améliorer les processus de l'organisation.
- [MCA3.21/02] La fonction d'amélioration des processus de l'organisation coordonne les activités d'amélioration de processus.
- [MCA3.21/03] Les propositions d'amélioration du processus de maintenance sont traitées conformément à une procédure documentée.
- [MCA3.21/04] Les améliorations du processus sont mises en oeuvre dans le cadre d'un projet pilote en vue de déterminer leur efficacité et les avantages qu'elles comportent avant leur introduction dans les pratiques normales.

[MCA3.21/05] Lorsqu'il a été décidé de transférer, dans les pratiques normales, une amélioration de processus de maintenance, l'amélioration est mise en oeuvre conformément à une procédure documentée.

**Annexe C : Glossaire des termes employés dans le modèle de
maturité de la maintenance (version 1.0)**

A	
[01]	Acceptation de logiciel 124
[02]	Activités de préparation de réception du logiciel 124
[03]	Analyse et vérification du logiciel transféré vers la maintenance 124
[04]	Assurance-qualité logiciel (secteur-clé 05) 125
B	
[05]	Besoins pour le support 125
[06]	Bon de travail de maintenance 125
[07]	Budget de la fonction de support aux usagers 126
C	
[08]	Catégories de maintenance logiciel 126
[09]	Choix de gestion en matière de formation 127
[10]	Choix de sous-traitants 127
[11]	Classes de maintenance 127
[12]	Classes de pénalisation 127
[13]	Critères d'assignation d'une ressource à une tâche de maintenance . 128
[14]	Critères de justification d'une demande de modification 128
[15]	Comités de gestion de logiciel 128
[16]	Cycle annuel de support 128
[17]	Cycle de vie du logiciel 129
D	
[18]	Définition du processus de l'organisation (secteur-clé 13) 129
[19]	Demande de service (cf. bon de travail) 129
E	
[20]	Éléments communs du bon de travail 129
[21]	Équipe compétente 130
[22]	Estimation d'une tâche de support 130
F	
[23]	Facilité de maintenance 130
[24]	Facteurs clés de satisfaction du personnel de maintenance logiciel 130

	[25]	Façon indépendante	131
G	[26]	Gestion de la configuration (secteur-clé 06)	131
	[27]	Gestion de la sous-traitance (secteur-clé 04)	131
	[28]	Gestionnaire de maintenance logiciel	132
	[29]	Gestion de la qualité (secteur-clé 18)	132
	[30]	Gestion de la formation (secteur-clé 14)	132
	[31]	Gestion de la transition vers la maintenance (secteur-clé 07)	132
	[32]	Gestion des améliorations du processus (secteur-clé 21)	133
	[33]	Gestion des changements technologiques (secteur-clé 20)	133
	[34]	Gestion des problèmes (secteur-clé 08)	133
	[35]	Gestion des spécifications (secteur-clé 01)	133
	[36]	Gestion du support aux usagers (secteur-clé 09)	133
	[37]	Gestion intégrée de la maintenance (secteur-clé 15)	134
	[38]	Gestion quantitative du processus (secteur-clé 17)	134
	[39]	Grille des responsabilités	134
I	[40]	Impacts sur la productivité du logiciel	134
	[41]	Ingénierie des produits (secteur-clé 16)	135
L	[42]	Logiciels d'outils de configuration	135
	[43]	Logiciels d'outils de gestion de documentation	135
	[44]	Logiciels d'outils de gestion de données	135
	[45]	Logiciels d'outils de gestion de code	135
	[46]	Logiciels d'outils de test et de <i>debugging</i>	135
M	[47]	Maintenance adaptative	136
	[48]	Maintenance corrective	136
	[49]	Maintenance logiciel	136
	[50]	Maintenance perfective	136
	[51]	Maintenance de support aux usagers	136
	[52]	Maître d'oeuvre	137
	[53]	Mesures et <i>reporting</i> de la gestion des problèmes	137
	[54]	Mesures et <i>reporting</i> de la fonction de support aux usagers	137
	[55]	<i>Monitoring</i> des événements critiques	137
N	[56]	Niveaux de services	138

	[57]	Normes de configuration	138
	[58]	Nouvelles versions de logiciel (cf. versions planifiées de logiciel)	138
O	[59]	Organisation de la fonction de maintenance logiciel (secteur-clé 10)	
		138	
	[60]	Outils de gestion de configuration (cf. logiciels d'outils de gestion de configuration)	
		138	
	[61]	Outils de test de version planifiée de logiciel ou de nouveau logiciel	
		138	
	[62]	Outils logiciels d'aide au processus de la conception et de la programmation des nouvelles versions de logiciel	139
P	[63]	Plan annuel de support	
		139	
	[64]	Plan d'amélioration de logiciel	
		139	
	[65]	Planification de la maintenance logiciel (secteur-clé 02)	139
	[66]	Planification du projet de maintenance	139
	[67]	Planification des RMPT	
		140	
	[68]	Planification de versions de logiciel	140
	[69]	Politiques d'assignation de ressources aux tâches de maintenance ..	140
	[70]	Politiques spécifiques à la maintenance logiciel	141
	[71]	Post-acceptation de logiciel	
		141	
	[72]	Pré-acceptation de logiciel	
		141	
	[73]	Prévention des défauts (secteur-clé 19)	141
	[74]	Priorité organisationnelle sur les processus (secteur-clé 12)	142
	[75]	Procédure d'analyse de situation de logiciel	142
	[76]	Processus d'acceptation de logiciel	
		142	
	[77]	Processus d'amélioration de logiciel	142
	[78]	Processus de bon de travail	
		142	
	[79]	Processus de budgétisation de la fonction de support aux usagers	
		143	
	[80]	Processus de gestion de la performance de la fonction de	

	support aux usagers	
143		
[81]	Processus de gestion des problèmes	143
[82]	Processus de mesure des tâches de maintenance logiciel	143
[83]	Processus de mise en oeuvre des nouvelles versions planifiées de logiciel	
144		
[84]	Processus de mise en oeuvre de la conception et de la programmation des versions planifiées de logiciel	144
[85]	Processus de la planification annuelle de la gestion du logiciel	144
[86]	Processus de planification annuelle du support	144
[87]	Processus de réalisation de la modification	145
[88]	Processus de traitement des demandes de service	145
[89]	Processus de traitement des demandes de service approuvées	145
[90]	Produits de travail (<i>cf.</i> produits logiciel)	146
[91]	Projet de maintenance	
146		
R		
[92]	Rapport de post-audit	
146		
[93]	Rapport annuel de support aux usagers	146
[94]	Rapport de réunion de comité de gestion de logiciel	146
[95]	Requête de maintenance	147
[96]	Référentiels	
147		
[97]	Règles et procédures de gestion des problèmes	147
[98]	RMPT	
147		
[99]	Rôle de l'utilisateur	
147		
[100]	Rôles des intervenants autour de la maintenance (secteur-clé 11) ...	148
S		
[101]	Satisfaction de l'utilisateur après la phase d'acceptation de logiciel	
148		
[102]	Services en maintenance offerts aux utilisateurs	148
[103]	Sous-traitants en maintenance logiciel	148
[104]	Standards de développement	
149		
[105]	Standards technologiques d'ingénierie	149
[106]	Statut du bon de travail	149
[107]	Suivi et supervision de la maintenance logiciel (secteur-clé 03)	149
T		

[108]	Test d'acceptation de logiciel	
	150	
[109]	Traitement des demandes de service	150
[110]	Types de formation des utilisateurs	150
[111]	Types de maintenance logiciel	
	151	

V

[112]	Version planifiée de logiciel	151
-------	--	-----

[01] **Acceptation de logiciel**

Dans le cadre d'implantation d'un logiciel et à travers le cycle de vie du logiciel, l'acceptation du logiciel est la phase qui se situe après la phase de développement et avant la phase de mise en production. La transition du développement vers la mise en production (opérations, maintenance, *etc.*) est une phase critique du cycle de vie du logiciel. C'est aussi la phase la moins gérée. Le gestionnaire joue un rôle important dans le contrôle de la qualité du logiciel qui vient du développement. Il doit s'assurer d'une transition douce vers le support.

La phase d'acceptation du logiciel poursuit plusieurs objectifs dont :

- le *design* correspond bien à aux intérêts du logiciel;
- le logiciel est complètement implanté;
- les opérations, les usagers et le personnel de maintenance (support) sont parfaitement équipées et entraînées pour assurer les opérations, l'utilisation et le support;
- *etc.*

[02] **Activités de préparation de réception du logiciel**

Les activités de préparation de réception du logiciel par la maintenance logiciel comprennent :

- la réalisation et la mise en oeuvre du plan de prise en charge du logiciel;
- le budget de prise en charge;
- l'acquisition du personnel de support et de formation;
- l'acquisition d'équipements et d'outils;
- la plate-forme d'accord pour la performance et les mesures de performance du support;
- la connaissance et la diffusion du cycle planifié de traitement;
- la documentation des activités de support pour les opérations;
- la documentation des activités de gestion;
- la documentation des activités d'amélioration du logiciel;
- la documentation des activités d'élaboration de mesures pour le support;
- *etc.*

[03] **Analyse et vérification du logiciel transféré vers la maintenance**

L'analyse et la vérification du logiciel transféré vers la maintenance est une activité du secteur-clé "Gestion de la transition". Cette activité se réalise dans la phase d'acceptation de logiciel. Elle consiste à veiller à ce que le logiciel transféré à la responsabilité de la maintenance soit analysé.

La vérification assure que le logiciel corresponde bien à aux caractéristiques énoncées du logiciel, c'est-à-dire que le logiciel est facile à maintenir, facile à opérer, facile à utiliser, *etc.*

[04] **Assurance-qualité logiciel (secteur-clé 05)**

Le secteur-clé "Assurance-qualité logiciel" regroupe l'ensemble des pratiques à réaliser pour assurer que les résultats obtenus après réalisation de la maintenance soient conformes aux exigences techniques de la requête de maintenance. L'assurance-qualité vise à fournir aux gestionnaires la vision appropriée sur le processus utilisé par le projet de maintenance, la RMPT ou les versions planifiées de logiciel et sur les produits en élaboration.

L'assurance qualité implique de passer en revue et d'auditer les produits et activités de maintenance pour vérifier s'ils sont conformes aux procédures et normes applicables et de fournir au personnel de maintenance et aux autres gestionnaires appropriés les résultats de ces revues et de ces audits.

[05] **Besoins pour le support**

Dans la phase d'acceptation de logiciel, les besoins en matière de support comprennent:

- la facilité de maintenance;
- la qualité du logiciel;
- la facilité de test;
- la compétence en *debugging*;
- la documentation;
- *etc.*

[06] **Bon de travail de maintenance**

Le bon de travail de maintenance est un formulaire de demande de service ou requête de maintenance à réaliser. Ce formulaire est appelé aussi *Work order*, demande de travail, *work request*, demande de service, *work service*, *etc.* C'est la base d'administration et de contrôle de la fonction de maintenance.

Le bon de travail de maintenance est utilisé pour administrer la planification des activités de support de logiciel qui comprennent :

- les demandes de modification de logiciel;
- les rapports *ad hoc*;
- les traitements spéciaux à réaliser;
- les services de support;
- *etc.*

Le bon de travail de maintenance contient des éléments communs [y] pour permettre l'analyse et l'approbation ou le rejet de la requête de maintenance.

[07] **Budget de la fonction de support aux usagers**

La fonction de support aux usagers dispose d'un budget. L'étude de ce budget tient compte de plusieurs points:

- les arriérés, c'est-à-dire les sommes correspondant à ce qui n'a pas été résolu ou ce qui n'a pas été dépensé;
- les demandes actuelles qui dépassent le budget;
- les ressources concurrentes, c'est-à-dire les ressources provenant du développement vs support;
- la satisfaction de l'utilisateur;
- la perception de l'informatique dans l'organisation;
- *etc.*

Le budget de support aux usagers est classé en trois catégories. Chaque catégorie offre une priorité:

- *première priorité* : dédiée aux dépenses en "support correctif". Cette catégorie de support obligatoire englobe le support à la production (résolution de problèmes, planification, support quotidien), le support aux modifications obligatoires (provenant de la maison mère, générés pour des besoins de réglementation, de système d'exploitation, de mise à jour de versions de langage de programmation, *etc.*), le support en consultation (études sur un système qui fonctionne) et le support en planification de recouvrement de désastres, de tests, de mise à jour, *etc.*
- *deuxième priorité*: dédiée aux dépenses en support administratif. Cette catégorie de support administratif englobe la gestion du logiciel (réunions, planification, budget, correspondance), le support en formation du personnel nouveau (utilisateurs et informaticien) et le support d'information (dossiers, rapports à la hiérarchie).
- *troisième priorité*: dédiée aux dépenses en support discrétionnaire. Cette catégorie de support discrétionnaire englobe le rehaussement ou le renchérissement des applications, le support en évaluation de systèmes, le support en demandes *ad hoc*, le support en amélioration de logiciel, le support en consultation pour des études de faisabilité, *etc.*

[08] **Catégories de maintenance logiciel**

La maintenance du logiciel est définie comme l'ensemble des travaux réalisés sur un logiciel après sa mise en opération. L'ensemble de ces tâches peuvent être regroupées en quatre catégories de maintenance:

- la maintenance corrective;
- la maintenance adaptative;
- la maintenance perfective ;
- la maintenance de support aux usagers.

[09] **Choix de gestion en matière de formation**

Les choix de gestion en matière de formation touchent la définition des tâches de maintenance, la gestion administrative du Département informatique et le développement du plan de carrière du personnel de maintenance.

[10] **Choix de sous-traitants**

Le choix du sous-traitant en matière de maintenance logiciel par le maître d'oeuvre est établi en fonction de la capacité du sous-traitant d'effectuer les travaux de maintenance logiciel. De nombreux facteurs interviennent dans la décision d'attribuer à un sous-traitant une portion ou la totalité des travaux confiés par le maître d'oeuvre. Les sous-traitants peuvent être sélectionnés en fonction d'alliances commerciales stratégiques ou encore de considérations techniques spécifiques.

[11] **Classes de maintenance**

La maintenance logiciel est différenciée par la nature de la tâche qu'elle réalise. Cette nature appelée classe de maintenance peut-être:

- corrective;
- adaptative;
- perfective;
- activités de support aux usagers.

[12] **Classes de pénalisation**

Dans le *monitorage* des événements critiques, il existe des classes de pénalisations (1 à 3) ou *business penalties* pour aider le gestionnaire à faire des choix et prendre une décision.

L'évaluation de la pénalité est fonction de la classe:

- *classe 1 de pénalisation*: entraîne des pénalités sévères. Elle englobe les dérapages en temps pour remettre la paie, des erreurs de calcul de la paie, un retard dans la clôture de fin de mois financière, des modifications exigées par des règles législatives nouvelles ou toute exigence de cet ordre.
- *classe 2 de pénalisation*: entraîne des pénalités modérées mais importantes. Elle englobe des erreurs de programme pour une sortie de rapport, une sortie de rapport qui prend du retard, le temps de réponse se dégrade.
- *classe 3 de pénalisation*: entraîne de possibles pénalités mais pas certaines. Elle englobe la sortie retardée d'un rapport comptable, la charge du système en composants critiques trop importante.

[13] **Critères d'assignation d'une ressource à une tâche de maintenance**

L'assignation d'une ressource à une tâche de maintenance nécessite de remplir des conditions comme:

- la connaissance du domaine de l'application;
- les capacités techniques;
- l'expérience en maintenance de l'application;
- l'expérience et le lien avec l'utilisateur;
- *etc.*

[14] **Critères de justification d'une demande de modification**

Les critères de justification d'une demande de modification d'un logiciel à la base d'une requête de maintenance englobent:

- des modifications d'ordre juridique imposées à l'organisation,
- des règles ou procédures internes à l'organisation dans le but d'éviter des coûts;
- réduction ou élimination de coûts dans le but de produire des revenus;
- *etc.*

[15] **Comités de gestion de logiciel**

Les comités de gestion de logiciel offrent aux gestionnaires de logiciel un mécanisme efficace pour gérer la maintenance. Les responsabilités du comité de gestion de logiciel sont :

- attribution des priorités aux demandes de travail;
- jugement sur la planification et le contenu des nouvelles versions planifiées de logiciel;
- autorisation et déclenchement d'études spéciales de post-audit;
- évaluations de logiciel;
- ré-ingénierie;
- conversions et migrations de logiciel;
- résolution de problèmes prenant des proportions importantes;
- *etc.*

[16] **Cycle annuel de support**

Le cycle annuel de support fait partie de la gestion de la fonction de support aux usagers. Ce cycle fournit un rapport annuel de gestion des systèmes. La réalisation de la clôture du cycle annuel de support procède à la consolidation des indicateurs analysés et des résultats obtenus dans le rapport annuel de support.

[17] **Cycle de vie du logiciel**

Le cycle de vie du logiciel dans la gestion du logiciel est composé de trois phases:

- *phase d'acceptation de logiciel*: phase de transition entre la fin du développement et la mise en production.
- *phase d'opérations et de support aux usagers* : phase de stabilisation et d'optimisation du système.
- *phase de retrait et de remplacement*: phase d'abandon et de retrait progressif du logiciel. C'est l'accompagnement du logiciel vers sa fin.

[18] **Définition du processus de l'organisation (secteur-clé 13)**

Le secteur-clé "Définition du processus de l'organisation" vise à développer et à maintenir un ensemble utilisable d'acquis processus améliorant la performance du processus d'un projet à l'autre et constituant le fondement pour des bénéfices cumulatifs et sur le long terme pour l'organisation.

La définition du processus de l'organisation comprend le développement et la maintenance du processus standard de l'organisation, de même que des acquis processus qui s'y rapportent , notamment les descriptions des cycles de vie, les lignes directrices et les critères d'adaptation des processus, la base de données du processus de l'organisation et une bibliothèque contenant la documentation se rapportant au processus .

[19] **Demande de service** (cf. bon de travail)

[20] **Éléments communs du bon de travail**

Les éléments communs du bon de travail ou *work order* sur lesquels reposent l'analyse de la requête de maintenance sont :

- demandeur;
- date de la demande;
- type de service demandé;
- justification de la demande;
- bénéfices attendus;
- obligatoire ou discrétionnaire;
- approbation de l'utilisateur;
- approbation de l'informatique;
- date actuelle de démarrage de la réalisation;
- date actuelle de fin de la réalisation;
- coûts d'estimation actuelle;
- statut actuel de la demande;

– *etc.*

[21] **Équipe compétente**

Une équipe de maintenance est considérée comme compétente lorsqu'elle maîtrise des connaissances basées sur des données historiques de l'application en maintenance. Ces connaissances peuvent être des connaissances:

- du domaine;
- de l'application;
- de l'environnement technique,
- *etc.*

[22] **Estimation d'une tâche de support**

L'estimation d'une tâche de support se base sur:

- l'intuition;
- l'analyse des échecs précédents;
- la comparaison de l'historique de tâches similaires;
- la comparaison de mesure (exemple des points de fonction basés sur une moyenne des coûts pour ajouter, modifier ou supprimer un point de fonction);
- l'estimation de logiciel façon système expert;
- l'approche simplifiée d'entreprise pour la meilleure estimation possible;
- *etc.*

[23] **Facilité de maintenance**

Appelée aussi *maintenability*, la facilité de maintenance du logiciel est déterminée par des caractéristiques de qualité comme :

- la modularité;
- la descriptivité;
- la consistance;
- la simplicité;
- l'extensibilité;
- la testabilité;
- *etc.*

[24] **Facteurs clés de satisfaction du personnel de maintenance logiciel**

Les facteurs-clés de satisfaction du personnel de maintenance encouragés par la direction sont:

- le respect;

- la reconnaissance "égale" à celle du développement;
- la position "positive" dans l'échelle des positions (titre, échelle, *etc.*);
- l'importance de la fonction de maintenance (le pouvoir de l'acceptation de logiciel par exemple institué comme veto de mise en production pour les projets de développement);
- la paternité des standards (pour les processus de gestion de logiciel);
- *etc.*

[25] **Façon indépendante**

La vérification exécutée par une personne ou par un groupe d'une "façon indépendante" consiste à faire exécuter la tâche de vérification par une autre personne ou un autre groupe que la personne ou le groupe qui a exécuté la tâche de réalisation.

[26] **Gestion de la configuration (secteur-clé 06)**

Le secteur-clé "Gestion de la configuration" vise à établir et à maintenir l'intégrité des produits du projet de maintenance ou des versions planifiées de logiciel tout au long du cycle de vie du projet de maintenance ou des versions planifiées de logiciel.

La gestion de configuration consiste à identifier la configuration du logiciel (c'est-à-dire les produits de travail sélectionnés et leur description) à des points déterminés du temps, à contrôler systématiquement les changements apportés à la configuration, et à maintenir son intégrité et sa *traçabilité* tout au long du cycle de vie.

Parmi les produits de travail placés sous gestion de configuration se trouvent les produits livrés au client (document des exigences et code par exemple) ainsi que les éléments nécessaires à la création de ces produits (compilateur, par exemple) ou identifiés avec ceux-ci.

[27] **Gestion de la sous-traitance (secteur-clé 04)**

Le secteur-clé "Gestion de la sous-traitance" regroupe l'ensemble des pratiques à réaliser pour sélectionner des sous-traitants qualifiés et à les gérer efficacement.

La gestion de la sous-traitance de la maintenance consiste à:

- choisir un sous-traitant;
- définir les engagements avec le sous-traitant;
- effectuer le suivi;
- passer en revue les résultats produits et la performance de ses livrables;
- *etc.*

Ces pratiques couvrent la gestion d'un contrat de sous-traitance de maintenance logiciel. Les travaux de

maintenance logiciel sous-traités sont réalisés par le sous-traitant. Le maître d'oeuvre veille à ce que ces activités soient réalisées de façon appropriée et à ce que les produits livrés par le sous-traitant satisfassent aux critères d'acceptation. Les normes que le sous-traitant doit respecter sont compatibles avec celles du maître d'oeuvre.

[28] **Gestionnaire de maintenance logiciel**

Le ou les gestionnaires de maintenance logiciel assurent la réalisation, le suivi et la supervision de la fonction de maintenance dans l'organisation.

[29] **Gestion de la qualité (secteur-clé 18)**

Le secteur-clé "Gestion de la qualité" du logiciel vise à développer une compréhension quantitative de la qualité des produits issus des activités de maintenance et à favoriser la réalisation d'objectifs de qualité spécifiques.

La gestion de la qualité englobe la définition des objectifs de qualité pour les produits, la mise en place de plans permettant d'atteindre ces objectifs et le suivi et l'ajustement des plans, produits de travail, activités et objectifs de qualité afin de satisfaire les besoins et les attentes du client et de l'utilisateur final en termes de produits de haute qualité.

[30] **Gestion de la formation (secteur-clé 14)**

Le secteur-clé "Gestion de la formation" vise à développer les compétences et les connaissances du personnel de maintenance pour qu'il puisse jouer pleinement son rôle, c'est-à-dire de façon efficace et efficiente.

[31] **Gestion de la transition vers la maintenance (secteur-clé 07)**

Le secteur-clé "Gestion de la transition vers la maintenance" regroupe l'ensemble des pratiques à réaliser pour couvrir la période qui se situe entre la fin de la phase de développement initial du logiciel et le début de la phase de support et des opérations. La transition se fait dans la phase d'acceptation de logiciel.

Les éléments d'une bonne gestion de transition vers la maintenance comprennent:

- la définition des rôles et des responsabilités;
- la mise en oeuvre d'un plan de transition avec choix des dates;
- la mise en oeuvre du suivi du plan de transition;
- *etc.*

[32] **Gestion des améliorations du processus (secteur-clé 21)**

Le secteur-clé "Gestion des améliorations de processus" vise l'amélioration continue du processus de la maintenance logiciel utilisé par l'organisation en vue d'améliorer la qualité logiciel, d'augmenter la productivité et de diminuer le cycle de vie de la maintenance.

La gestion des améliorations du processus comprend la définition des objectifs d'amélioration du processus et leur mise en oeuvre.

[33] **Gestion des changements technologiques (secteur-clé 20)**

Le secteur-clé "Gestion des changements technologiques" vise à identifier les nouvelles technologies avantageuses pour les activités de maintenance logiciel (outils, méthodes et processus) et à en effectuer le suivi de façon ordonnée au sein de l'organisation.

La gestion des changements technologiques comprend l'identification, le choix et l'évaluation de nouvelles technologies et l'intégration dans l'organisation de celles qui sont efficaces. Ce secteur a pour objectif l'amélioration de la qualité, l'augmentation de la productivité et la diminution de la durée du cycle de la maintenance.

[34] **Gestion des problèmes (secteur-clé 08)**

Le secteur-clé "Gestion des problèmes" regroupe l'ensemble des pratiques à réaliser pour coordonner et solutionner les problèmes qui ont un lien direct ou indirect avec le personnel de maintenance.

[35] **Gestion des spécifications (secteur-clé 01)**

Le secteur-clé "Gestion des spécifications" regroupe l'ensemble des pratiques à réaliser pour couvrir deux aspects majeurs des spécifications en maintenance:

- les spécifications vis à vis du client pour mettre en oeuvre ses spécifications;
- les spécifications techniques vis à vis de la technologie pour réaliser la tâche à effectuer.

Les spécifications vis à vis du client visent à établir entre le client et le gestionnaire de maintenance logiciel [y] une compréhension commune des spécifications du client que l'équipe de réalisation se propose de satisfaire.

[36] **Gestion du support aux usagers (secteur-clé 09)**

Le secteur-clé "Gestion du support aux usagers" vise à assurer aux usagers une qualité de service au meilleur coût et au meilleur délai.

[37] **Gestion intégrée de la maintenance (secteur-clé 15)**

Le secteur-clé "Gestion intégrée de la maintenance" sur le plan logiciel vise à intégrer les activités d'ingénierie et de gestion logiciel sous forme d'un processus défini et cohérent qui est adapté à partir du processus logiciel de maintenance standard de l'organisation.

[38] **Gestion quantitative du processus (secteur-clé 17)**

Le secteur-clé "Gestion quantitative du processus" vise à contrôler quantitativement la performance du processus de la maintenance. La performance du processus de la maintenance correspond aux résultats réels obtenus en respectant un processus de maintenance donné.

[39] **Grille des responsabilités**

En matière de rôles des intervenants autour de la maintenance, une grille des responsabilités permet d'identifier les tâches à réaliser et d'assigner ces tâches à des ressources. La grille des responsabilités permet également de :

- clarifier le rôle et le degré d'implication de chacun;
- identifier formellement les responsabilités;
- réalise une gestion de logiciel plus aisée;
- *etc.*

[40] **Impacts sur la productivité du logiciel**

Les impacts sur la productivité du logiciel sont classés par les gestionnaires de maintenance par niveau:

– *priorité numéro un ou niveau "essentiel"*: inclut des spécifications précises de l'utilisateur et une bonne définition du problème, la participation de l'utilisateur aux tests ainsi que l'obligation de donner des informations de suivi ou *feedback* à l'utilisateur.

– *priorité numéro deux ou niveau "très souhaitable"*: inclut l'utilisation d'un code structuré en développement, l'existence d'une tribune pour s'informer des personnes s'occupant d'un même système, une excellente connaissance du domaine de l'utilisateur, la compétence en programmation modulaire (structurée), l'utilisation de documentation éducative d'actualité, la disponibilité de ressources d'expérience, la définition standardisée des données (dictionnaires de données), la documentation de l'historique des modifications, l'élaboration de fichiers de tests, *etc.*

– *priorité numéro trois ou niveau "utile"*: inclut une documentation compréhensible et actualisée de l'utilisateur, l'approche de versions planifiées, les modifications non urgentes, la définition claire des responsabilités, la participation au développement de représentants de la maintenance et la révision de ce

qui fonctionne bien, l'obligation des procédures d'acceptation de systèmes, l'utilisation d'éditeurs techniques pour maintenir la documentation, l'utilisation de générateurs de tests, *etc.*

[41] **Ingénierie des produits (secteur-clé 16)**

Le secteur-clé "Ingénierie de produits" vise à exécuter systématiquement un processus d'ingénierie bien défini intégrant toutes les activités d'ingénierie en vue de produire de façon efficace et efficiente des produits cohérents et corrects.

[42] **Logiciels d'outils de configuration**

Les logiciels d'outils de configuration aident à une gestion plus efficace du secteur-clé "gestion de configuration". Les logiciels d'outils de configuration englobent des outils pour coordonner le code Source vs code objet, comparer le code source, fusionner le code source, *etc.*

Les outils de gestion de configuration permettent de suivre l'ensemble des changements apportés sur les systèmes et les applications ainsi que leurs raisons, les dates, les personnes responsables des changements, *etc.*

[43] **Logiciels d'outils de gestion de documentation**

Les logiciels d'outils de gestion de documentation aident à une gestion plus efficace du secteur-clé Gestion de configuration. Ils sont utilisés aussi pour d'autres tâches de développement, de maintenance, *etc.* Les logiciels d'outils de gestion de documentation englobent des outils pour générer des formes graphiques, traitement de textes, croiser des listes de données ou de code ainsi que corriger des textes, données, code source, *etc.*

[44] **Logiciels d'outils de gestion de données**

Les logiciels d'outils de gestion de données sont utilisés pour des tâches de développement, de maintenance, *etc.* Les logiciels d'outils de gestion de données englobent des outils pour comparer, standardiser et manipuler des données.

[45] **Logiciels d'outils de gestion de code**

Les logiciels d'outils de gestion de code sont utilisés pour des tâches de développement, de maintenance, *etc.* Les logiciels d'outils de gestion de code aident à analyser le code statique et le code dynamique, reformer le code (alignement et indentation du code), restructurer le code (transformation de code non structuré en code structuré, identifier le code supplémentaire et inutile), convertir des langages, *etc.*

[46] **Logiciels d'outils de test et de *debugging***

Les logiciels d'outils de test et de *debugging* sont utilisés pour des tâches de développement, de maintenance, *etc.* Les logiciels d'outils de test et de *debugging* aident gérer les problèmes, analyser les interruptions, comparer le code source, tester et «debugger» en mode interactif, «monitorer» le test, générer

des données de test (par extraction des fichiers de données de la production ou par analyse des descriptions de fichiers), émuler des scénarios de tests.

[47] **Maintenance adaptative**

Activité de maintenance qui consiste à apporter des compléments ou des modifications aux fonctionnalités du système pour le rendre conforme aux besoins nouveaux de l'organisation.

[48] **Maintenance corrective**

Activité de maintenance qui consiste à apporter des corrections à un programme dans l'immédiat ou à court terme dans le but de garder les systèmes en opération.

[49] **Maintenance logiciel**

L'activité de maintenance logiciel n'a pas un contenu et un statut normalisés. Tant au niveau de la tâche qu'au niveau de l'environnement de la tâche, l'exécution de la maintenance est souvent différente d'une organisation à une autre. Ces différences reposent sur plusieurs aspects dont la nature des tâches de maintenance, le processus d'exécution de la maintenance, l'emplacement de la fonction de maintenance dans l'organigramme structurel d'une organisation, sa mission, ses ressources, *etc.*

Il faut adjoindre à cette définition la caractéristique suivante pour en limiter l'envergure : « *toutes les activités de maintenance qui se trouvent dans les mains de programmeurs, d'analystes et de gestionnaires de logiciels d'applications* ». Ce qui exclut de ce champ notamment les équipements, l'installation et l'exploitation des ordinateurs et des réseaux.

Dans le cadre de cette étude, la maintenance logiciel se compose de plusieurs catégories de maintenance.

[50] **Maintenance perfective**

Activité de maintenance qui consiste à apporter des modifications d'ordre technique sans remettre en cause les fonctionnalités du système. Cette amélioration génère souvent une meilleure performance.

[51] **Maintenance de support aux usagers**

Activité de maintenance qui englobe plusieurs autres activités comme :

- les demandes de rapports *ad hoc* par les usagers;
- le temps consacré à fournir des explications aux usagers sur le fonctionnement des applications et des différents cadres et règles dans lesquels s'exécutent et s'opèrent ces applications;
- les demandes d'études d'impacts en cas de changements envisagés dans l'avenir;
- *etc.*

[52] **Maître d'oeuvre**

Le maître d'oeuvre en matière contractuelle est la partie qui fait appel à la sous-traitance pour réaliser des travaux. Le maître d'oeuvre confie des travaux de maintenance logiciel à des sous-traitants en maintenance logiciel.

[53] **Mesures et reporting de la gestion des problèmes**

Les mesures et les données de *reporting* des problèmes dans le secteur-clé de la "gestion des problèmes" comprennent:

- un numéro affecté au problème;
- un code pour distinguer la sévérité du problème;
- un code pour distinguer le type de problème;
- un code pour distinguer le module associé au problème;
- le temps dépensé pour solutionner le problème;
- les coûts en termes d'affaires comparativement aux pénalités dues au problème;
- le ratio des coûts occasionnés par le problème comparativement aux coûts occasionnés par la formation de l'utilisateur;
- le ratio des coûts occasionnés par le problème comparativement aux coûts occasionnés par la prévention de la maintenance hardware;
- le ratio des coûts occasionnés par le problème comparativement aux coûts occasionnés par la modification du système;
- *etc.*

[54] **Mesures et reporting de la fonction du support aux usagers**

Les mesures et les données du *reporting* de la gestion du support sont basés sur:

- des critères de productivité du personnel utilisateur et informatique;
- la qualité de service;
- la performance du support;
- les niveaux de service (estimation vs actuels);
- la satisfaction des utilisateurs;
- les facteurs de motivation du personnel (rotation du personnel, absentéisme, *etc.*);
- *etc.*

[55] **Monitoring des événements critiques**

Le monitoring des événements critiques est divisé en classes de pénalisation [12] pour aider les gestionnaires de support à établir des choix éclairés.

[56] **Niveaux de services**

La notion de "niveau de service" en matière de maintenance logiciel consiste à classer la qualité et l'efficacité du service fourni par la fonction de support aux usagers en niveaux. Les niveaux de service sont définis d'un commun accord entre l'informatique et l'utilisateur moyennant des coûts associés aux critères de qualité et d'efficacité.

[57] **Normes de configuration**

Les normes de configuration sont un ensemble de règles, procédures et politiques à appliquer pour une gestion de configuration optimale

Exemples :

- ne gérer qu'une seule configuration à la fois;
- ne maintenir qu'une seule version de logiciel à la fois;
- les mises à jour de logiciel système et de logiciel d'application sont effectuées dans des versions séparées;
- l'utilisateur dispose en permanence de deux ou trois niveaux de versions de système (courante, release, patch);
- *etc.*

[58] **Nouvelles versions de logiciel** (cf. versions planifiées de logiciel)

[59] **Organisation de la fonction de maintenance logiciel (secteur-clé 10)**

Le secteur-clé "Organisation de la fonction de maintenance" logiciel vise à rendre performant la fonction de maintenance logiciel dans l'organisation en mettant en place une structure organisationnelle adéquate, des processus de gestion de performance, des comités de gestion de logiciel, *etc.*

[60] **Outils de gestion de configuration** (cf. logiciels d'outils de gestion de configuration)

[61] **Outils de test de version planifiée de logiciel ou de nouveau logiciel**

Des outils de tests de la nouvelle version planifiée ou d'un nouveau logiciel englobent :

- les tests et debugging interactifs;
- le monitoring des tests;
- les générateurs de données de test, les tests online, les outils de gestion de tests, *etc.*;
- *etc.*

[62] **Outils logiciels d'aide au processus de la conception et de la programmation des nouvelles versions de logiciel**

Les outils logiciels d'aide au processus de la conception et la programmation englobent :

- des outils de développement comme les "Case Tools";
- des analyseurs statiques comme "Inspector";
- des vérificateurs de code source;
- des analyseurs d'impacts;
- des concepteurs d'échantillons de données;
- *etc.*

[63] **Plan annuel de support**

L'exécution du plan annuel de support met en oeuvre:

- les mesures d'évaluation du succès;
- la documentation par produit;
- le rapport des résultats sur une base périodique (semaine, mois, trimestre);
- *etc.*

[64] **Plan d'amélioration de logiciel**

Les plans d'amélioration du logiciel sont établis par les gestionnaires de logiciel. Les fonctionnalités du système ne sont pas le point le plus important du plan d'amélioration mais peuvent être

incluses pour bénéficier d'une vue complète du système. Un des points contenus dans un plan d'amélioration de logiciel est de garder le personnel de maintenance et de gestion du logiciel attentifs et de s'assurer pour contribuer au succès de l'implantation du logiciel que le personnel de support suscite une égale reconnaissance par rapport au personnel de développement.

[65] **Planification de la maintenance logiciel (secteur-clé 02)**

Le secteur-clé "Planification de la maintenance logiciel" couvre trois types de maintenance : planification de projet de maintenance, planification des RMPT et planification de versions planifiées de logiciel.

[66] **Planification du projet de maintenance**

La planification de projet de maintenance vise à établir des prévisions raisonnables pour la mise en oeuvre des travaux de réalisation et de gestion du projet de maintenance. Elle comprend l'estimation des travaux à effectuer, l'établissement d'engagements vis à vis du client et l'élaboration du plan pour effectuer les travaux de réalisation.

La planification de projet de maintenance vise à établir des prévisions raisonnables pour la mise en oeuvre des travaux de réalisation et de gestion du projet de maintenance.

La planification de projet de maintenance comprend l'estimation des travaux à effectuer, l'établissement d'engagements vis à vis du client et l'élaboration du plan pour effectuer les travaux de réalisation.

Le processus de planification de projet de maintenance comprend des étapes permettant d'estimer la taille des produits de travail et les ressources nécessaires, de produire un calendrier de mise en oeuvre, d'identifier et d'évaluer les risques et de négocier les engagements avec le client. A ce stade s'établit le plan du projet de maintenance .

[67] **Planification des RMPT**

La planification des RMPT se fait sur une base prévisionnelle de charge de travail, définie sur une période déterminée de réalisation des RMPT (annuelle par exemple). Elle comprend l'estimation de la charge de travail à effectuer, l'établissement des engagements nécessaires auprès des clients, l'affectation des ressources nécessaires à la réalisation des tâches et l'élaboration d'un plan global pour réaliser les travaux. Elle dispose d'un historique représentant les estimations et les réalisations des RMPT.

Il n'y a pas de plan de réalisation fait au préalable pour chacune des RMPT avant qu'elle ne soit connue du groupe de maintenance et rendue formelle mais il existe un plan global de charge des RMPT. Chaque RMPT fait l'objet d'une étude préliminaire rapide pour se voir attribuer un statut dans la file d'attente des RMPT en instance de réalisation.

Dans la gestion des files d'attente, les urgences sont relativement prévisibles en terme de charge de travail et dominées par la planification. Un des rôles que joue le groupe de maintenance est d'aller au devant des besoins des clients qui sont basés plus sur une utilisation anticipée des systèmes que sur l'utilisation actuelle par la planification prévisionnelle et la réingénierie.

[68] **Planification de versions de logiciel**

La planification de versions de logiciel vise à implanter périodiquement de nouvelles versions de logiciels pour réaliser les changements exigés par les besoins d'affaires et par les besoins techniques à partir d'applications existantes. Les changements en question proviennent des demandes de services approuvées et non encore réalisées. Mais le processus de planification de nouvelles versions de logiciel est globalement identique à celui du développement.

[69] **Politique d'assignation de ressources aux tâches de maintenance**

La politique suivie en matière de maintenance est composée de plusieurs pratiques :

- la pratique d'assignation de ressources aux tâches de maintenance qui consiste à affecter les nouveaux employés aux systèmes de moindre importance stratégique, que ces systèmes soient récents ou non;
- la pratique de rotation de poste qui consiste à familiariser le personnel affecté à la maintenance et les systèmes pour des besoins de polyvalence;

- La pratique du poste-type qui consiste à répondre aux besoins d'avancement du personnel IS. Cette politique est critique pour la motivation, la performance et la satisfaction de l'employé dans sa tâche de maintenance et englobe les préoccupations suivantes: variété des aptitudes, identité de la tâche, portée de la tâche, autonomie, feed-back;
- *etc.*

[70] **Politiques spécifiques à la maintenance logiciel**

Dans le secteur-clé "Assurance-qualité logiciel", les politiques spécifiques à la maintenance logiciel dont le fonctionnement est régulièrement contrôlé englobent:

- les critères d'acceptation;
- les revues de qualité;
- l'inspection visuelle ou automatisée du code;
- la revue de la documentation;
- l'analyse de complexité;
- *etc.*

[71] **Post-acceptation de logiciel**

La phase de post-acceptation de logiciel, ayant comme prérogative d'éliminer les failles et déficiences du logiciel, met en oeuvre les plans d'amélioration de logiciel.

[72] **Pré-acceptation de logiciel**

La phase de pré-acceptation de logiciel assure la conception d'une nouvelle formule de gestion de logiciel en formant des groupes d'utilisateurs, d'opérations et de support, met en place les standards d'acceptation basés sur les standards de développement, communique les attentes aux «développeurs» et met en place le plan d'acceptation de logiciel.

[73] **Prévention des défauts (secteur-clé 19)**

Le secteur-clé "Prévention des défauts" vise à identifier les causes de défauts et à empêcher qu'ils ne se reproduisent.

Le secteur-clé La prévention des défauts couvre l'analyse des défauts identifiés dans le passé et le déclenchement d'actions spécifiques afin d'empêcher que des défauts du même type ne se reproduisent. Il peut s'agir de défauts identifiés dans d'autres projets de maintenance, versions planifiées de logiciel ou lors des premières phases ou tâches de projet de maintenance en cours. Les activités de prévention des

défauts servent également de mécanisme de transmission des leçons retenues d'un projet de maintenance à l'autre.

[74] **Priorité organisationnelle sur les processus (secteur-clé 12)**

Le secteur-clé "Priorité organisationnelle sur les processus" vise à établir une responsabilité organisationnelle quant aux activités reliées au processus de la maintenance en vue d'améliorer l'ensemble de la capacité du processus de maintenance de l'organisation.

La priorité organisationnelle sur les processus comprend le développement et le maintien de la bonne compréhension du processus de maintenance de l'organisation et des projets et la coordination des activités d'évaluation, de maintenance et d'amélioration de ces processus.

[75] **Procédure d'analyse de situation de logiciel**

La procédure d'analyse de situation de logiciel déclenche une proposition de plan d'amélioration de logiciel. Elle est basée sur l'excès de maintenance qui entraîne une détérioration du logiciel et sur l'existence de problèmes dus à la qualité technique du logiciel.

[76] **Processus d'acceptation de logiciel**

Le processus formel d'acceptation de logiciel englobe des caractéristiques spécifiques de définition d'objectifs et de critères de révisions. L'acceptation de logiciel par le support est formalisée par écrit et prévoit des rapports d'information à la Direction. Le processus dispose de ressources suffisantes en personnel expérimenté et d'un plan d'acceptation et de mise en oeuvre de la planification des activités d'acceptation de logiciel.

[77] **Processus d'amélioration de logiciel**

Le processus d'amélioration de logiciel consiste à sélectionner les logiciels pour évaluation, définir des catégories d'évaluation, définir les critères d'évaluation de chaque catégorie, réaliser l'évaluation de logiciel, identifier les améliorations potentielles, réaliser les analyses coûts - bénéfiques, donner des priorités d'amélioration, documenter le plan d'amélioration et le rédiger, obtenir les approbations, implanter et faire le suivi, évaluer et communiquer les résultats, mesurer et refaire le processus.

[78] **Processus de bon de travail**

Le processus du bon de travail ou *Work order* assure que la modification est autorisée par le client, facilite l'administration de la modification de logiciel demandée, fournit une trace des modifications de logiciel, garde une trace pour justifier des dépenses en temps et aide à prendre des décisions d'acceptation ou de rejet des demandes.

[79] **Processus de budgétisation de la fonction de support aux usagers**

Le budget de la fonction de support dispose d'un budget qui se fait sur une base annuelle.

[80] **Processus de gestion de la performance de la fonction de support aux usagers**

La mise en oeuvre du processus de gestion de la performance nécessite d'identifier les facteurs d'influence sur le succès comme:

- l'établissement et la clarification des responsabilités;
- L'établissement des révisions de critères de la performance;
- Fixer des objectifs raisonnables de performance;
- utiliser des critères de mesure de la performance qui sont basés sur des fonctions de l'application et sur l'environnement technique où s'exécute cette application;
- *etc.*

[81] **Processus de gestion des problèmes**

Le processus de gestion des problèmes fait appel au support aux usagers et au support aux opérations. Ce processus est classé en catégories: préventive, conditionnelle, temporaire, adaptative et corrective.

Il existe des standards du processus de gestion des problèmes:

- la hiérarchie des procédures à suivre peut être définie à l'avance entre l'informatique, les utilisateurs et les fournisseurs;
- l'utilisateur identifie le problème, collecte "l'évidence" (sentiment logique de l'événement qui a causé le problème ainsi que le détail de l'opération habituelle), informe l'informatique du problème, autorise
 - les retraitements (*reruns*), corrige les données erronées, détecte et informe sur les *outputs* défectueux, assure la continuité des activités par un système (manuel ou autre) pendant le temps de l'interruption.
 - l'informatique coordonne la résolution du problème, identifie les causes du problème, corrige l'erreur, obtient de l'utilisateur l'autorisation de procéder à des retraitements (*reruns*).
- *etc.*

[82] **Processus de mesure des tâches de maintenance logiciel**

Le processus de mesures des tâches de maintenance logiciel aide à :

- évaluer la taille;
- évaluer le coût;
- évaluer les délais de réalisation;
- choisir le type de ressources correspondantes;
- *etc.*

[83] **Processus de mise en oeuvre des nouvelles versions planifiées de logiciel**

Le processus de mise en oeuvre de nouvelles versions planifiées est composé de deux phases: la planification et l'approbation du contenu de la version planifiée du logiciel ainsi que la conception et la programmation de la version planifiée du logiciel.

[84] **Processus de mise en oeuvre de la conception et de la programmation des versions planifiées de logiciel**

Le processus de mise en oeuvre de la conception et de la programmation englobe:

- l'analyse;
- la conception;
- la programmation;
- la documentation,;
- le contrôle de la configuration;
- les tests;
- les mises en production;
- *etc.*

[85] **Processus de la planification annuelle de la gestion du logiciel**

Le processus de la planification annuelle de la gestion du logiciel comprend l'identification des activités majeures attendues pour l'année à venir.

L'identification se fait à partir de l'évaluation des résultats de l'année en cours (performance du support et niveau de service fournis, demandes encore en attente, performance du système, résultats des évaluations de systèmes, *etc.* L'identification se fait aussi à partir des besoins planifiés et guidés par des impératifs juridiques, techniques, d'affaires, *etc.*

[86] **Processus de planification annuelle du support**

Dans le processus de la planification annuelle du support aux usagers sont identifiées les activités majeures pour parvenir à un niveau de portefeuille visé, appelé aussi niveau d'applications. La révision des activités de support comprend:

- les accords sur les responsabilités;
- les accords sur la performance;
- les niveaux de service visés par l'informatique;
- le cycle des versions planifiées;
- les plans de recouvrement en cas de désastre;
- les efforts visés pour le support;
- *etc.*

Dans ce cadre là, un montant forfaitaire, en terme d'effort de support par catégorie, est institué comme technique pour la formation d'équipes et pour l'amélioration de la productivité (le coût de support par catégorie comparé à ce qui se fait en général dans l'industrie par exemple).

[87] **Processus de réalisation de la modification**

Le processus de réalisation de la modification comprend:

- la compréhension de la demande;
- la compréhension du système;
- la localisation des zones ou des éléments touchés par le changement à apporter;
- l'identification de tous les impacts possibles en amont et en aval;
- la codification et la compilation;
- les tests;
- la mise au courant du coordinateur utilisateur du logiciel;
- l'implantation suivant les conventions et standards de l'informatique;
- la vérification des résultats et des garanties;
- *etc.*

[88] **Processus de traitement des demandes de service**

Le processus de traitement des demandes de service exige pour chaque demande de:

- réceptionner la demande;
- identifier le besoin;
- analyser la demande;
- estimer la demande;
- approuver ou rejeter la demande;
- informer le demandeur;
- *etc.*

[89] **Processus de traitement des demandes de service approuvées**

Le processus de traitement des demandes approuvées exige pour chaque demande approuvée de:

- donner une priorité à la demande approuvée;
- attribuer des ressources pour réaliser la demande approuvée;
- fixer des délais de réalisation;
- préparer le suivi;
- clôturer le dossier une fois la demande réalisée.

[90] **Produits de travail** (*cf.* produits logiciel)

[91] **Projet de maintenance**

Un projet de maintenance est un projet qui remplit toutes les caractéristiques d'un projet de développement avec la particularité que l'application centrale (autour de laquelle se construira le projet de maintenance) existe. L'application centrale détient les spécifications techniques et clients de base.

[92] **Rapport de post-audit**

Le rapport de post-audit fournit:

- un sommaire de gestion;
- une description des objectifs;
- les résultats majeurs constatés: (fonctionnalités planifiées vs actuelles, performance planifiée vs actuelle, contrôle et sécurité planifiés vs actuels, coûts et bénéfices planifiés vs actuels, points de désaccord sur l'implantation, *etc.*
- *etc.*

Ce rapport fournit en outre des recommandations d'amélioration de processus de développement, d'améliorations de logiciel ainsi que la possibilité de parfaire les habiletés et la compétence du personnel chargé de l'audit.

Un examen de post-audit ne peut être réalisé qu'avec l'autorisation mutuelle de l'informatique et des utilisateurs impliqués dans la phase d'acceptation de logiciel. Il ne peut être réalisé que dans les deux années suivant l'implantation du logiciel et une fois que toutes les phases d'acceptation de logiciel sont réalisées.

[93] **Rapport annuel de support aux usagers**

Le rapport annuel de support aux usagers fournit un sommaire des résultats de l'année et du plan de l'année prochaine ainsi que la documentation de gestion de logiciel collectée durant l'année. Les changements apportés au système sont décrits dans le rapport annuel de support. Cela contribue à valoriser auprès du personnel le sens de la réalisation et de l'accomplissement de soi en plus de fournir un dossier de l'historique des conditions d'affaires et des décisions qui en ont découlé.

[94] **Rapport de réunion de comité de gestion de logiciel**

Les réunions des comités de gestion de logiciel fournissent le cadre essentiel de la prise de décision et des déclenchements de réalisations d'actions. Leur fréquence est mensuelle ou annuelle. Il est souhaitable de choisir la fréquence opportune en fonction du contexte.

[95] **Requête de maintenance**

Une requête de maintenance est une demande de réalisation d'un besoin basé sur des raisons techniques provenant de l'informatique ou sur des raisons applicatives provenant du client. La réalisation

peut prendre la forme d'un projet de maintenance, d'une nouvelle version planifiée de logiciel ou d'une ou de plusieurs requêtes de maintenance de petite taille.

[96] **Référentiels**

Spécification ou produit qui sert de base à un développement. Dans le cadre de la gestion de configuration, cette spécification ou produit ne peut être modifié qu'à travers des procédures de contrôle de changement.

[97] **Règles et procédures de gestion des problèmes**

Lors d'un problème ou d'une interruption de systèmes, des règles et des procédures sont utilisées:

- les opérations fournissent la première ligne de défense en matière de hardware, communications, système et problèmes d'application;
- l'analyste de maintenance contacté devient le point central pour la résolution du problème et l'utilisateur coordinateur devient le point central de la communication;
- les standards existent et sont appliqués formellement;
- la responsabilité de l'informatique est entière;
- le rôle de l'utilisateur est important;
- *etc.*

[98] **RMPT**

Les RMPT ou requêtes de maintenance de petite taille correspondent à des besoins en maintenance qui n'entrent ni dans la catégorie des projets de maintenance ni dans la catégorie des versions planifiées de logiciel parce que les ressources et les délais qu'elles nécessitent pour leur réalisation sont réduits.

[99] **Rôle de l'utilisateur**

Le rôle de l'utilisateur est de:

- mettre en oeuvre les directives et les plans;
- coordonner les changements apportés au système avec les autres unités;
- mesurer les besoins vs les fonctionnalités actuelles;
- mesurer la performance et les bénéfices;
- s'assurer d'une utilisation efficace du système par les utilisateurs;
- communiquer les besoins en changement;
- communiquer les problèmes de performance;
- *etc.*

[100] **Rôles des intervenants autour de la maintenance (secteur-clé 11)**

Le secteur-clé "Rôles des intervenants autour de la maintenance" vise à identifier le rôle et les responsabilités de chacun: l'informaticien de maintenance, l'utilisateur, l'opérateur des systèmes, *etc.*

[101] **Satisfaction de l'utilisateur après la phase d'acceptation de logiciel**

La satisfaction de l'utilisateur comporte des critères précis de mesure:

- fonctionnalités complètes;
- exactitude;
- performance;
- facilité d'utilisation;
- documentation;
- formation;
- *etc.*

[102] **Services en maintenance offerts aux utilisateurs**

Les services offerts aux utilisateurs sont classés par catégorie de maintenance. Ces services englobent:

- le support urgent;
- la résolution d'erreurs planifiées;
- les modifications;
- les améliorations;
- les demandes ad hoc;
- la consultation;
- la formation;
- *etc.*

[103] **Sous-traitants en maintenance logiciel**

En matière contractuelle, le sous-traitant en maintenance logiciel est la partie qui exécute des travaux de maintenance à la demande du maître d'oeuvre.

[104] **Standards de développement**

Les standards de développement comprennent:

- la sélection d'un langage de programmation;
- la conception orientée sur le critère de facilité de maintenance maximale;

- la codification correspondant à une facilité de maintenance maximale;
- la documentation;
- la performance;
- la sécurité;
- le contrôle;
- *etc.*

D'autres critères y contribuent aussi comme:

- la facilité des tests et debugging;
- la mise en oeuvre des standards d'acceptation de logiciel;
- la participation du support aux revues de développement et d'acceptation de logiciel;
- *etc.*

[105] **Standards technologiques d'ingénierie**

Une politique de standards technologiques d'ingénierie vise l'obtention de facilité de modifier, de tester, *etc.* avec les outils d'installation de tests et de supports de tests.

106] **Statut du bon de travail**

Le statut du bon de travail ou *work order* a pour rôle de situer et d'informer sur la progression de la réalisation:

- travail commencé?
- réalisé?
- suspendu?
- en attente de réalisation?
- annulé?
- *etc..*

[107] **Suivi et supervision de la maintenance logiciel (secteur-clé 03)**

Le secteur-clé "Suivi et la supervision de la maintenance logiciel" vise à donner une bonne perspective de l'avancement réel des travaux de façon à ce que les gestionnaires puissent intervenir efficacement lorsque la performance du projet de maintenance s'écarte de façon significative des prévisions. le suivi et la supervision de projets de maintenance couvrent:

- le suivi des réalisations et des résultats;
- leur revue par rapport aux estimations, engagements et plans documentés;
- l'adaptation de ces plans en fonction des réalisations et des résultats réels;
- *etc.*

Une planification documentée pour le projet de maintenance sert de base au suivi des activités de maintenance, aux communications sur l'état des travaux et à la révision de la planification.

Dans le cas des RMPT, l'analyse et la comparaison des résultats se font sur une base globale et périodique comprenant l'ensemble des RMPT réalisées.

Dans le cas des versions planifiées de logiciel, le suivi et la supervision est identique au suivi et à la supervision des projets de maintenance.

Le suivi et la supervision des activités de maintenance sont assurés par les gestionnaires.

[108] **Test d'acceptation de logiciel**

Les tests d'acceptation de logiciel sont réalisés:

– *Par l'utilisateur* : test de fonctionnalités, performance, documentation, formation, évaluation de la facilité d'utilisation ou *usability*, etc.

– *Par le support* : tests de conformité avec les standards, documentation, caractéristiques des opérations, résultats de test, évaluation de la facilité d'opérer ou *operability*, etc.

– *Par les opérations* : tests de conformité avec les standards, documentation, qualité de logiciel, résultats de tests, tests de facilité de maintenance ou *maintenability*, évaluation de la facilité de donner du support ou *supportability*, etc.

[109] **Traitement des demandes de service**

Le processus de traitement des demandes est basé sur trois approches:

- le bon de travail ou *work order* individuel;
- les versions planifiées de logiciel (*sheduled releases*);
- les projets de maintenance.

[110] **Types de formation des utilisateurs**

Le type de formation des utilisateurs pour chaque type de maintenance peut être centralisé : une personne ou un groupe de personnes se forme et formera à son tour l'ensemble du personnel ou bien

l'équipe d'utilisateurs dans sa totalité (en rotation) va sur le site se former au support à la formation, au support à l'implantation et au support à la post-implantation

[111] **Types de maintenance logiciel**

Dans le cadre de cette étude, les types de maintenance logiciel sont:

- projet de maintenance;
- RMPT ou requête de maintenance de petite taille;
- versions planifiées de logiciel.

[112] **Version planifiée de logiciel**

L'approche de l'implantation de versions planifiées de logiciel contribue à:

- concentrer les efforts;
- rendre les équipes stables;
- donner plus d'efficacité;
- améliorer les tests;
- donner de meilleures analyses des problèmes;
- pousser les utilisateurs à être sélectifs;
- traduire un bon esprit d'équipe;
- améliorer le profil du gestionnaire;
- aider à une meilleure formation de l'utilisateur;
- faciliter la participation de l'utilisateur aux tests;
- accroître le moral du programmeur;
- fournir un meilleur contrôle;
- procurer une meilleure connaissance des attentes des gestionnaires;
- offrir un traitement efficace aux problèmes;
- faciliter le *monitoring* de la performance informatique;
- servir de base de justification de réécriture de certaines parties du système;
- *etc.*

Annexe D : Article ayant fait l'objet de publication