

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

PROPOSITION D'UNE GRILLE D'ÉVALUATION DES FACTEURS DE RISQUE DES
PROGRAMMES D'AMÉLIORATION DU PROCESSUS LOGICIEL POUR LES
ORGANISATIONS DÉSIRANT PROGRESSER DU NIVEAU 1 AU NIVEAU 2 DU
MODÈLE CMM

PROPOSITION DE L'ACTIVITÉ DE SYNTHÈSE
PRÉSENTÉE
AU SOUS-COMITÉ D'ADMISSION ET D'ÉVALUATION
MAÎTRISE EN INFORMATIQUE DE GESTION

Par
VINH DUONG

OCTOBRE 2000

TABLE DES MATIÈRES

INTRODUCTION.....	1
1 MÉTHODE DE RECHERCHE.....	2
2 DÉFINITION DU PROJET.....	6
2.1 MOTIVATION.....	6
2.1.1 <i>Productivité en informatique</i> :.....	6
2.1.2 <i>Processus logiciel</i> :.....	6
2.1.3 <i>Gestion des risques</i> :.....	8
2.2 OBJET DE L'ÉTUDE.....	9
2.3 OBJECTIF DE L'ÉTUDE.....	11
2.4 UTILISATEURS.....	13
3 PLANIFICATION DU PROJET.....	14
3.1 ÉTAPE 1 : IDENTIFIER LES FACTEURS DE RISQUES GÉNÉRIQUES RELIÉS AU PROJET D' AMÉLIORATION DU PROCESSUS LOGICIEL EN S' APPUYANT SUR CMM.....	14
3.2 ÉTAPE 2 : IDENTIFIER LES FACTEURS DE RISQUES RELIÉS AUX SECTEURS CLÉS DU NIVEAU 2 DE CMM.....	16
3.3 ÉTAPE 3 : CONCEVOIR UNE GRILLE D' ÉVALUATION DES FACTEURS DE RISQUE DES PROGRAMMES D' AMÉLIORATION DU PROCESSUS LOGICIEL POUR LES ORGANISATIONS DÉSIRANT PROGRESSER DU NIVEAU 1 AU NIVEAU 2 DU MODÈLE CMM.....	18
3.4 ÉTAPE 4 : RÉVISION DE LA GRILLE PAR LES EXPERTS.....	20
3.3.1 <i>Lignes directrices relatives à la revue des documents</i>	21
4 TABLEAU RÉCAPITULATIF.....	21
5 ÉCHÉANCIER.....	24
RÉFÉRENCES.....	25

LISTE DES TABLEAUX

Tableau		Page
TABLEAU 1.1	STRUCTURE POUR LA RECHERCHE EN GÉNIE LOGICIEL (ADAPTÉ DE BASILI, SELBY ET HUTCHENS, 1986)	4
TABLEAU 2.2	STRUCTURE POUR LA RECHERCHE EN GÉNIE LOGICIEL – DÉFINITION DU PROJET	13
TABLEAU 3.3	PLANIFICATION DU PROJET, ÉTAPE 1	15
TABLEAU 3.4	PLANIFICATION DU PROJET, ÉTAPE 2	18
TABLEAU 3.5	PLANIFICATION DU PROJET, ÉTAPE 3	20
TABLEAU 5.6	ÉCHÉANCIER DE L' ACTIVITÉ DE SYNTHÈSE DE VINH DUONG.....	24

INTRODUCTION

Ce travail est une étude exploratoire ayant pour but de développer une grille d'évaluation des facteurs de risque des programmes d'amélioration du processus logiciel pour les organisations désirant progresser du niveau 1 au niveau 2 selon le modèle CMM (Paulk et al, 1993a, 1993b).

Une grille d'évaluation comprend un questionnaire et une méthode d'analyse de réponses composée de plusieurs questions conçues d'une façon structurée pour déterminer d'une manière objective les principaux facteurs de risque spécifiques à un projet. Le résultat de cette évaluation permet de formuler des recommandations afin de réduire les risques connus. La grille peut être divisée en plusieurs thèmes comprenant chacun un certain nombre de questions. La pondération de la grille peut être adaptée ou modifiée selon les besoins spécifiques (Laframboise, 1996, Girard, 1998).

Ce travail est destiné aux groupes d'ingénierie du processus logiciel (*Software engineering process group* ou SEPG) ou à ceux qui remplissent des fonctions similaires. Le SEPG est un groupe de spécialistes dont la tâche est de faciliter la définition, la maintenance et l'amélioration du processus logiciel dans l'organisation (Paulk et al, 1993a, 1993b). Le SEPG est composé de praticiens chevronnés possédant des compétences diverses en génie logiciel. Il est le carrefour des efforts de collaboration des membres de l'organisation qui sont impliqués dans l'amélioration du processus logiciel (Fowler et Rifkin, 1990).

1 Méthode de recherche

Pour mieux aborder les problèmes reliés à la recherche exploratoire, une adaptation du cadre de Basili (Basili, Selby et Hutchens, 1986), pour la recherche expérimentale en génie logiciel, est utilisée pour rendre plus structuré ce projet. Celui-ci fournit une méthode pour classer d'une manière systématique des étapes clés du projet aidant ainsi à mieux comprendre le travail et à faciliter l'évaluation des études expérimentales. Ce modèle comporte 4 phases principales : la définition, la planification, l'exécution et l'interprétation des résultats.

Le cadre de Basili est un excellent outil permettant l'élaboration de la synthèse des travaux. En effet, à la fin de chacune des quatre phases, le cadre de Basili est utilisé pour présenter le compte-rendu des résultats de recherche. La table des matières de cette proposition suit aussi le cadre de Basili. Elle n'inclut que les deux premières phases du projet qui sont la définition et la planification. La table des matières du mémoire suivra aussi le cadre de Basili. Elle inclura les quatre phases du projet qui sont la définition, la planification, l'exécution et l'interprétation des résultats. Depuis sa publication, le cadre de Basili a été adapté pour des études exploratoires notamment celles de (Abran, Laframboise et Bourque, 1999, Abran *et al.* , 1998, Wolff, 1999). Selon les mêmes auteurs, le cadre de Basili constitue un excellent outil pour éviter les embûches communes telles que : mal définir les objectifs du projet, commencer trop tôt la collecte des données sans se soucier de définir correctement l'objet devant être mesuré tout en cueillant des données inutilisables. Ces erreurs peuvent nul doute causer la mauvaise interprétation des résultats ou rendre difficile leur généralisation. Le cadre de Basili encourage les chercheurs à définir correctement le but et les objectifs du projet avant de s'aventurer plus loin ; la conception des échantillons et la définition des mesures doivent être aussi clairement indiquées. Ainsi, le cadre de Basili rend plus rigoureuse les recherches expérimentales au niveau du domaine du génie logiciel.

Selon Sekaran (Sekaran, 1992), une étude exploratoire est entreprise lorsque nous ne connaissons pas la problématique courante ou que nous n'avons pas d'information sur la façon de résoudre des problèmes similaires. Dans de tels cas, des études préliminaires approfondies doivent être entreprises pour se familiariser avec la problématique. Cette approche permet de

développer une solution initiale qui sera suivie plus tard par une plus rigoureuse investigation. Les études exploratoires sont donc initiées pour arriver à mieux comprendre la nature des problèmes du fait que très peu d'études ont été conduites sur la problématique à laquelle nous faisons face.

Le cadre de Basili a été conçu à l'origine pour des projets de recherche où le domaine de connaissance est déjà développé et structuré, et où la collecte approfondie des données est réalisable. Cependant, pour un contexte de recherche exploratoire, où les concepts ne sont pas encore clairement définis ou lorsque le domaine de connaissance n'est pas encore bien structuré, il est à souligner que quelques adaptations du cadre de Basili sont requises (Abran, Laframboise et Bourque, 1999).

La structure du cadre de Basili, reposant sur les 4 phases principales demeure inchangée. Quelques adaptations sont réalisées à l'intérieur de chaque phase. Étant donné qu'il n'y a pas de collecte formelle des données pour les analyses statistiques dans une telle recherche exploratoire, les étapes reliées à la conception des échantillons et aux analyses statistiques ont été enlevées du cadre. La phase de définition comporte 4 composantes : motivation, objet, objectif et utilisateurs. La motivation est à priori la raison qui pousse l'auteur à entreprendre le projet ; Elle identifie la grande question qu'on veut répondre. L'objet définit l'entité principale à être étudiée. L'objectif est le but précis du projet ; il constitue le véritable problème qu'on veut résoudre. Les utilisateurs sont les personnes susceptibles de mettre en pratique les résultats de la recherche.

Tableau 1.1 Structure pour la recherche en génie logiciel
(adapté de Basili, Selby et Hutchens, 1986)

<i>I. Définition du projet</i>			
Motivation	Objet	Objectif	Utilisateurs
<i>II. Planification du projet</i>			
Étapes du projet		Intrants	Biens Livrables
<i>III. Exécution</i>			
Développement des livrables		Révision par les experts	Amélioration du livrable
<i>IV. Interprétation</i>			
Contexte d'interprétation		Extrapolation	Travaux subséquents

Le projet de recherche exploratoire est planifié en détail dans la deuxième phase. Un plan de projet est développé avec l'identification des étapes. Les intrants du projet sont identifiés. Ils peuvent être composés de littérature scientifique ou de rapports sur les expériences connues dans l'industrie. Les personnes qui auront à réviser les biens livrables du projet sont aussi identifiées. Nous entrerons en communication avec eux pour obtenir une confirmation écrite sur leur volonté de participer au projet. Les biens livrables doivent être clairement définis afin de bien gérer les attentes des divers intervenants.

Les étapes du projet identifiées dans la phase de planification sont réalisées dans la phase d'exécution où les livrables proposés sont développés en se basant sur une revue approfondie de la littérature. La qualité de ces livrables est ensuite améliorée grâce aux révisions des experts provenant du monde académique et de l'industrie.

L'interprétation des résultats du projet constitue la dernière phase du projet. Elle est subdivisée en trois parties : le contexte d'interprétation, l'extrapolation des résultats et les travaux subséquents. Dans le contexte d'interprétation, nous examinerons le résultat final de la recherche pour voir si l'objectif a été atteint ou non. Dans l'extrapolation, nous étudierons les résultats pour voir s'ils peuvent être utilisés dans d'autres circonstances. Dans les travaux subséquents, nous discuterons des recherches plus poussées dans le même domaine d'étude.

2 DÉFINITION DU PROJET

2.1 Motivation

2.1.1 Productivité en informatique :

Aux États-Unis, les entreprises dépensent plus de \$250 milliards par année pour le développement de 175,000 projets informatiques. Les recherches de *The Standish Group* (Standish Group, 1995) démontrent que 31% des projets sont annulés en cours de route. 52% des projets dépasseront leur budget pour coûter jusqu'à 189% du montant estimé initialement. Selon les mêmes études, ces échecs et ces cas de dépassement de budget ne représentent que le bout de l'iceberg. Les coûts représentant les pertes d'opportunité ne sont pas mesurables, lesquels sont estimés à plus de mille milliards de dollars. Se basant sur ces statistiques, *The Standish Group* estime qu'en 1995, les entreprises américaines et les organismes gouvernementaux américains dépensaient \$81 milliards pour des projets qui seraient annulés. Ces mêmes organisations ont déboursé un montant additionnel de \$59 milliards pour les projets complétés mais qui dépassaient leur estimé original. Les risques sont toujours un facteur important lorsque la nouvelle technologie est adoptée. Cependant, plusieurs des projets qui ont échoué, représentaient des systèmes de base pour toute entreprise comme la création d'une base de données des permis de conduire, l'implantation d'un progiciel de comptabilité ou la création d'un système de commande.

2.1.2 Processus logiciel :

Confrontés à ces problèmes, les spécialistes suggèrent de calquer une approche qui a fait ses preuves dans le domaine manufacturier et qui passe par la définition et l'amélioration des processus pour accroître la qualité des produits, dans le respect des coûts et des délais (Basque, 1998). Selon le modèle *Capability Maturity Model* ou CMM (Paulk *et al.* , 1995), un processus est une séquence d'étapes exécutées pour atteindre un objectif donné. Le processus reflète ce que les gens font, utilisant les procédures, les méthodes, les outils, et les équipements, pour transformer les matières premières (intrants) en produits finis (extrants) dotés d'une valeur pour les clients. Dans une organisation considérée immature, les processus logiciels sont

improvisés par les informaticiens et les gestionnaires au cours du projet. Dans cette organisation, même si un processus logiciel est déjà mis en place, il ne sera pas respecté ni mis en vigueur. Cette organisation est toujours en mode de réaction ; ses gestionnaires sont souvent appelés à résoudre les crises immédiates. Ces organisations dépassent régulièrement les échéances et les budgets parce que ceux-ci ne sont pas basés sur des estimés réalistes. Quand une échéance serrée leur est imposée, ces organisations peuvent compromettre les fonctionnalités et la qualité du produit pour rencontrer leur date de livraison antérieurement convenue.

Toujours selon les mêmes auteurs, une organisation ayant un processus logiciel mature possède une bonne capacité à gérer les processus de développement et d'entretien des logiciels. Elle prescrit clairement le processus logiciel à suivre au personnel existant et aux nouveaux employés. Les activités sont ainsi entreprises selon le processus planifié. Toute l'organisation est impliquée dans l'amélioration du processus logiciel. Les rôles et les responsabilités sont clairement définis durant le projet. Dans une organisation mature, le gestionnaire fait le suivi de la qualité des logiciels et des processus qui les produisent. Il existe une base quantitative pour évaluer la qualité des produits et analyser les problèmes. Les échéances et les budgets sont réalistes du fait qu'ils sont basés sur des performances historiques. Les coûts, l'échéance, la fonctionnalité prévue des logiciels sont dans la plupart des cas respectés. En général, une organisation mature suit avec discipline et d'une manière consistante un processus logiciel parce que tous les participants connaissent les bénéfices de leurs actions. Il y existe une infrastructure pour supporter ce processus logiciel.

L'amélioration du processus logiciel est un projet difficile parce qu'elle requiert la mobilisation de toute l'entreprise. Comparé à un projet informatique qui implique en général un département ou un groupe d'utilisateurs et d'informaticiens reliés au projet, l'amélioration du processus logiciel est un projet d'envergure pouvant être aussi complexe qu'un projet informatique. Si les statistiques indiquent que 31% des projets informatiques sont annulés avant la date de fin et que 52% dépassent leur budget pour coûter jusqu'à 189% du budget estimé (Standish Group, 1995), nous pouvons raisonner que les projets d'amélioration du processus logiciel devraient aussi subir des taux élevés d'échec. Il est impossible d'obtenir les

statistiques sur les entreprises ayant échoué dans l'implantation du processus logiciel. Il s'en suit probablement que les organisations sont réticentes à diffuser les nouvelles de l'échec de leur projet d'implantation du processus logiciel pour la simple raison que cela pourrait nuire à l'image de l'entreprise. Par contre, celles qui réussissent à parvenir au niveau 2 de CMM et plus, participent activement aux sondages parce que ce succès au contraire projette hors de tout doute une image positive de l'entreprise. Il nous est permis de remarquer que les entreprises mentionnées dans la littérature sur le succès de l'amélioration du processus logiciel ne représentent probablement pas le profil moyen de la maturité dans l'industrie. Nous estimons que la réalité serait moins brillante comparée aux données des sondages parce que, tout comme les projets informatiques, beaucoup d'organisations abandonne probablement leur projet d'amélioration du processus logiciel.

2.1.3 Gestion des risques :

Selon les auteurs Sisti et Joseph (Sisti et Joseph, 1994), historiquement, les décideurs gestionnaires ont réussi à gérer les risques reliés à l'informatique en se basant sur leur propre compétence et expérience. Ce n'est plus le cas aujourd'hui. À cause de la complexité des systèmes informatiques, il est essentiel que ces derniers utilisent des méthodes plus disciplinées et systématiques pour gérer les risques informatiques. Pour gérer efficacement les risques, il est nécessaire d'utiliser les méthodes déjà définies dans le domaine de gestion de risque. Ces méthodes doivent faciliter la communication aux équipes appartenant à tous les niveaux de l'organisation. Elles doivent fournir aux décideurs assez d'information pour classer par ordre de priorité et minimiser les risques avec des ressources limitées. Les activités proposées par ces auteurs pour gérer les risques en génie logiciel sont :

- Identifier: Identifier d'avance les risques reliés aux projets logiciels.
- Analyser: Transformer les risques identifiés en informations utilisables par des décideurs.
- Planifier: Développer des plans d'actions pour minimiser les risques individuels ; Classer en ordre de priorité les actions et les intégrer dans un plan d'exécution de gestion des risques.
- Contrôler: Corriger les déviations de ces actions en ayant soin d'utiliser des outils de gestion de projet.
- Communiquer: Diffuser les informations sur la gestion des risques à travers l'organisation.

À titre de récapitulation, cette section présente la motivation de l'étude qui est l'examen des problèmes de productivité reliés aux projets en génie logiciel. Une des solutions proposée est la mise en place d'un processus logiciel plus mature. Son implantation constitue un projet d'envergure qui est couvert de risques. Ce projet peut subir le même sort d'échec qu'un grand nombre de projets informatiques. Il existe actuellement des méthodes pour gérer les risques des projets informatiques que nous pouvons adopter pour faciliter l'implantation du processus logiciel.

2.2 Objet de l'étude

En 1987, le Software Engineering Institute (SEI) de l'Université Carnegie Mellon a été mandatée par l'armée de l'air américaine pour développer une méthode ayant pour but de déterminer la compétence de ses fournisseurs informatiques. Un questionnaire sur la maturité (*Maturity Questionnaire* ou MQ) et l'évaluation de la capacité logiciel (*Software Capability Evaluation* ou SCE) ont été en conséquence développés. Une structure de cinq niveaux sur laquelle est basé le questionnaire, a été plus tard formalisée comme modèle appelé *Capability Maturity Model for Software* ou CMM. Le premier niveau est le moins mature et le cinquième niveau est le plus mature.

Les cinq niveaux de maturité sont les suivants (Paulk et al, 1993) :

1. Niveau Initial : Le processus logiciel est caractérisé par la prédominance d'interventions ponctuelles, voire chaotiques. Très peu de processus sont définis et la réussite dépend des efforts individuels.
2. Niveau Reproductible : Une gestion de projet élémentaire est définie pour assurer le suivi des coûts, des délais et de la fonctionnalité du produit. La discipline nécessaire au processus est en place pour reproduire la réussite de projets d'un même domaine d'application.
3. Niveau Défini : Le processus logiciel des activités de gestion et d'ingénierie est documenté, normalisé et intégré dans le processus logiciel standard de l'organisation. Tout nouveau projet de développement ou de maintenance logiciel fait intervenir une version adaptée et approuvée du processus logiciel standard de l'organisation.
4. Niveau Maîtrisé : Des mesures détaillées sont prises en ce qui concerne le déroulement du processus logiciel. Le processus logiciel et le niveau de qualité des produits sont connus et contrôlés quantitativement.
5. Niveau d'Optimisation : Une amélioration continue du processus logiciel est mise en œuvre par une rétroaction quantitative émanant du processus lui-même et par l'application d'idées et de technologies innovatrices.

Les organisations qui réussissent à améliorer leur processus logiciel, ont obtenu des bénéfices importants. Pour les entreprises impliquées dans les contrats avec le *Department of Defense* américain, le CMM devient un facteur significatif dans l'évaluation de la capacité des services informatiques (Thomson, 1998). L'octroi des contrats peut se baser sur la maturité du processus logiciel. Ainsi une organisation ayant une évaluation de CMM de niveau trois et plus a plus de chance d'être sélectionné pour un projet informatique qu'un compétiteur n'ayant qu'un niveau deux ou moins (McGarry, 1995). Voici un exemple illustrant l'enjeu important de posséder un processus logiciel mature. Au mois de mai 1999, *The Metropolitan Transportation Authority/Long Island Rail Road (LIRR)* a accordé à Bombardier Transport un contrat pour 192 voitures automotrices électriques. Ce contrat vise le remplacement de sa flotte de trains de banlieue. La valeur du contrat est de 655 millions de dollars canadiens et couvre la conception, la production et la livraison des nouvelles voitures. Le contrat inclut des options pour la production de 808 voitures supplémentaires. Si toutes les options étaient exercées, la valeur totale du contrat pourrait atteindre 2,7 milliards de dollars canadiens (\$1.850 milliards US) (Bombardier, 1999). Le LIRR stipule que, en ce qui concerne le processus de développement des logiciels dédiés aux voitures automotrices électriques, les fournisseurs de Bombardier et Bombardier Transport ont deux ans pour progresser du niveau 1 au niveau 2 de CMM. Après 2 ans, l'organisation qui ne réussit pas à gagner le niveau 2 de CMM verra son contrat de fabrication de trains révisé par le client (Basque, 1999).

L'amélioration du processus logiciel est une activité coûteuse qui demande un investissement très important en temps. En effet, dans une étude menée par le SEI auprès de 48 organisations (Hayes et Zubrow, 1995), on remarque que le période de temps nécessaire pour progresser du niveau 1 au niveau 2 de maturité peut varier en moyenne de 18 à 30 mois. Dans des cas extrêmement spéciaux, il existe des organisations qui prennent de 58 à 94 mois pour parvenir au niveau 2 de CMM. L'amélioration du processus logiciel est très complexe parce qu'elle entraîne des changements importants au niveau de la culture organisationnelle (Krasner, 1994). Elle exige l'engagement de la direction qui constitue un élément indispensable pour fournir le soutien nécessaire aux projets (Humphrey, 1990). Elle demande aussi la mise en place d'une infrastructure organisationnelle ayant pour but de supporter les équipes de génie logiciel dans

leur effort de produire du logiciel d'une manière efficace et systématique (Zahran, 1998). Elle nécessite la participation du personnel parce que ceux-ci ont une bonne connaissance et expérience des qualités et des défauts du processus de travail courant. Elle exige le maintien de la visibilité et de la clarté du projet ainsi que de ses activités qui sont des éléments cruciaux pour sa survie. Au début du projet, les résultats ne sont pas hautement visibles. Le personnel a tendance à perdre de vue les objectifs à long terme du projet d'amélioration du processus logiciel, particulièrement pendant les périodes de perturbation ou de crise (McFeeley, 1996). Elle demande le recrutement d'un champion qui est un gestionnaire technique ayant le respect des praticiens et en même temps l'écoute des gestionnaires d'affaires. Le champion a la responsabilité de réaliser l'implantation du processus logiciel (Pressman, 1988). Le champion maintient le cap sur l'objectif, surmonte les obstacles, et refuse d'abandonner au moment où le parcours devient difficile (Humphrey, 1990). Elle demande l'amélioration des capacités de travail des employés par les formations (Wiegers, 1996).

À titre de récapitulation, cette section présente l'objet de cette étude qui est l'amélioration du processus logiciel en se basant sur le modèle CMM. Un processus logiciel peut appartenir à l'un des cinq niveaux suivants classés par ordre croissant de maturité : initial (le moins mature), reproductible, défini, maîtrisé et finalement optimisé (le plus mature). L'amélioration du processus logiciel peut apporter des bénéfices importants à l'entreprise notamment au niveau de l'obtention des contrats d'affaires. Cependant elle est aussi une activité coûteuse et risquée qui demande un investissement à long terme. Le personnel ne devrait pas perdre de vue les objectifs à long terme de ce projet.

2.3 Objectif de l'étude

Ce projet est un travail exploratoire parce que, à notre meilleure connaissance, il n'existe pas encore de grille qui permet à une entreprise d'identifier les risques pendant le projet d'implantation du processus logiciel basé sur le modèle CMM. Cette grille évitera à la compagnie à faire des recherches exhaustives dans la littérature ou d'embaucher des consultants externes pour réaliser les mêmes objectifs. En effet, il existe des articles qui discutent des avantages reliés à l'amélioration du processus logiciel de l'entreprise. On peut trouver des livres

qui décrivent le modèle CMM. Il existe aussi des sondages menés par le groupe Software Process Improvement Capability dEtermination, plus connu sous l'acronyme SPICE (SPICE, 1998), et par le SEI sur les facteurs de succès reliés à l'implantation du processus logiciel. Il y a beaucoup d'articles sur le développement des mesures du logiciel, sur la gestion des risques des projets informatiques. Par contre, il n'existe pas de littérature sur une méthode objective et reproductible pour évaluer les risques reliés à l'implantation du processus logiciel au début du projet. Cette grille facilitera l'évaluation des risques. Les sources des risques seront identifiées et leurs effets potentiels évalués. Un processus de contrôle du risque pourra être ensuite développé par l'utilisateur pour mettre en place un plan de résolution du risque (Hall, 1998). Cette grille répondra aussi à un des besoins identifiés dans le sondage de la S.E.I. sur les entreprises qui utilisent le modèle CMM (Goldenson et Herbsled, 1995). En effet, un grand nombre de répondants affirme avoir besoin de l'aide pour pouvoir réussir à améliorer leur processus logiciel.

Selon une évaluation de 734 organisations qui s'est terminée en août 1999 (CMM, 1999) : 43.2% des sites évalués démontraient un niveau de maturité **1**, 34.2% un niveau **2**, 17.3% un niveau **3**, 4% un niveau **4** et 1.4% un niveau **5**. Il est à noter que ce n'est pas nécessairement un échantillon représentatif de l'industrie, les compagnies qui ont accepté de faire des évaluations officielles sont probablement des leaders ou sont contraintes à le faire pour des raisons contractuelles. Le passage du niveau 1 à 2 est choisi étant donné qu'un pourcentage important d'organisations se trouve encore au niveau 1. L'impact de la grille d'évaluation des risques proposée sera donc important parce qu'elle visera un nombre important d'utilisateurs potentiels qui font partie des organisations appartenant au niveau de maturité 1 de CMM. Cette grille pourra être utilisée une fois que toute l'infrastructure est mise en place pour démarrer le projet d'amélioration du processus logiciel. Elle permettra à l'organisation de surveiller de manière proactive le projet d'amélioration du processus logiciel sur une base préventive afin d'identifier les zones de problème éventuel et les mesures à prendre et au besoin pour assurer l'avancement du projet.

À titre de récapitulation, cette section présente l'objectif du projet qui est le développement d'une grille d'évaluation des facteurs de risque des programmes d'amélioration du processus

logiciel pour les organisations désirant progresser du niveau 1 au niveau 2 selon le modèle CMM. Le passage du niveau 1 à 2 est choisi parce que selon les études, un grand nombre d'entreprises se trouvent encore au niveau 1 de maturité.

2.4 Utilisateurs

Ce travail est destiné aux équipes de coordination de l'implantation du processus logiciel (*software engineering process groups* ou SEPG) ou à ceux et celles qui ont pour mission la mise en place d'un processus logiciel basé sur CMM.

Le tableau 2.2 ci-dessous présente la synthèse de la définition du projet incluant la motivation poussant l'auteur à entreprendre le projet, l'objet du projet, l'objectif du projet et les utilisateurs qui vont utiliser le bien livrable du projet.

Tableau 2.2 Structure pour la recherche en génie logiciel – Définition du projet

I. Définition du projet			
Motivation	Objet	Objectif	Utilisateurs
<ul style="list-style-type: none"> Réduire les échecs des projets informatiques. 	<ul style="list-style-type: none"> Amélioration du processus logiciel en se basant sur le modèle CMM. 	<ul style="list-style-type: none"> Développer une grille d'évaluation des facteurs de risque des programmes d'amélioration du processus logiciel pour les organisations désirant progresser du niveau 1 au niveau 2 selon le modèle CMM 	<ul style="list-style-type: none"> Équipes de coordination de l'implantation du processus logiciel (<i>software engineering process groups</i>)

3 PLANIFICATION DU PROJET

Ce chapitre est divisé en quatre sections représentant les quatre étapes de la phase d'exécution du projet. L'étape 1 constitue une revue de la littérature sur les risques génériques reliés au projet d'implantation de CMM. L'étape 2 est une revue de la littérature sur les risques reliés aux secteurs clés du niveau 2 de CMM. L'étape 3 constitue une revue de la littérature sur la conception d'une grille de gestion des risques en informatique. Et finalement, l'étape 4 présente les experts qui feront la révision de la grille, le processus qui sera suivi par ces experts pour fournir leurs commentaires ainsi que comment leur résolution sera documenté.

3.1 Étape 1 : Identifier les facteurs de risques génériques reliés au projet d'amélioration du processus logiciel en s'appuyant sur CMM.

I.D.E.A.L. (McFeeley, 1996) constitue un excellent outil pour guider le développement et la gestion d'un programme d'amélioration du processus logiciel. Il est choisi principalement parce qu'il est développé par le S.E.I. pour faciliter l'amélioration du processus logiciel en s'appuyant sur CMM. Nous l'avons adopté comme modèle de base pour identifier les risques génériques reliés à l'implantation du processus logiciel et pour connaître en même temps les bonnes approches à suivre dans l'amélioration du processus logiciel. Le modèle I.D.E.A.L. est composé de cinq phases qui constituent une boucle continue dans l'amélioration du processus logiciel :

- Dans la phase initiale, sont définis les objectifs du programme d'amélioration du processus logiciel, est établie l'infrastructure initiale pour supporter et faciliter le programme d'amélioration du processus logiciel.
- Dans la phase du diagnostic, le plan d'action est élaboré en se basant sur la vision de l'organisation et sur le plan stratégique d'affaires. Une évaluation de la maturité du processus logiciel est entreprise.
- Dans la phase de mise en place, les problèmes que l'organisation veut résoudre par l'amélioration du processus logiciel sont classés par ordre de priorité. Les stratégies sont développées. Les objectifs mesurables sont ainsi déterminés.
- Dans la phase d'action, les activités d'amélioration planifiées précédemment sont déployées à travers l'organisation.
- La dernière phase de la boucle, la révision, vise à de rendre le prochain passage à travers le modèle I.D.E.A.L. plus efficace. Arrivé à cette étape, les solutions sont développées, les leçons sont apprises, les mesures sur la performance et la réussite des objectifs sont collectées. Les corrections et les ajustements de la stratégie, des méthodes et de

l'infrastructure se font en conséquence avant de recommencer de nouveau la boucle du modèle I.D.E.A.L.

En se basant sur le modèle I.D.E.A.L., nous avons répertorié les quatre catégories de risques génériques suivantes :

- Les risques liés à l'engagement de la direction.
- Les risques liés à la communication avec les participants du projet d'amélioration du processus logiciel.
- Les risques liés à l'établissement des objectifs d'affaires.
- Les risques liés à l'infrastructure supportant le nouveau processus.

Le tableau ci-dessous représente le sommaire des activités de recherche pour identifier les facteurs de risques génériques liés au modèle I.D.E.A.L.

Tableau 3.3 Planification du projet, étape 1

<i>II. Planification du projet</i>		
Étape 1	Intrants	Livrables
<ul style="list-style-type: none"> • Identifier les risques génériques liés au projet d'amélioration du processus logiciel en s'appuyant sur CMM 	<ul style="list-style-type: none"> • (McFeeley, 1996) I.D.E.A.L., Software Engineering Process Group Guide 	Facteurs de risques génériques relatifs au modèle CMM répartis dans les catégories suivantes : <ul style="list-style-type: none"> • Risques liés à l'engagement de la direction. • Risques liés à la communication avec les participants du projet d'amélioration du processus logiciel • Les risques liés à l'établissement des objectifs d'affaires • Les risques liés à l'infrastructure.

3.2 Étape 2 : Identifier les facteurs de risques reliés aux secteurs clés du niveau 2 de CMM

Pour parvenir au niveau 2 de CMM, l'organisation doit maîtriser les six secteurs clés suivants (Paulk et al, 1993a, 1993b) : la gestion des exigences, la planification des projets informatiques, le suivi et la supervision de projet logiciel, la gestion de la sous-traitance logiciel, l'assurance-qualité logiciel et la gestion de la configuration logiciel. Parmi ces six secteurs clés, nous avons choisi d'étudier les trois secteurs qui nous intéressent les plus. Ils sont :

- Gestion des exigences : Elle vise à établir une compréhension commune, entre le client et les informaticiens, des exigences du client que ce projet se propose de satisfaire.
- Suivi et supervision de projet logiciel : Ces activités visent à donner une bonne perspective de l'avancement réel des travaux de façon que les gestionnaires puissent intervenir efficacement lorsque la performance du projet logiciel s'écarte de façon significative des prévisions.
- Gestion de la configuration logiciel : Elle vise à établir et à maintenir l'intégrité des produits du projet logiciel tout au long du cycle de vie logiciel du projet.

Chaque secteur clé est décrit en termes de pratiques clés contribuant à en satisfaire les objectifs. Les pratiques clés décrivent l'infrastructure et les activités qui contribuent le plus à la mise en œuvre et à l'institutionnalisation efficace du secteur clé. Les principales pratiques de chaque secteur clé sont réparties parmi les cinq caractéristiques communes qui sont des attributs indiquant si la mise en œuvre et l'institutionnalisation d'un secteur clé sont efficaces, reproductibles et durables (Paulk *et al.* , 1993a, 1993b). Pour identifier les risques reliés à chacun des trois secteurs clés, nous analyserons leur fonctionnement à travers les caractéristiques communes identifiées ci-dessous :

- Engagement de réalisation (*Commitment to perform*) : Mesures que l'organisation doit prendre pour veiller à ce que le processus soit mis en œuvre de façon durable.
- Capacité d'exécution (*Ability to perform*) : Conditions préalables devant être présentes dans le projet ou dans l'organisation pour mettre en œuvre, de façon compétente, le processus logiciel.
- Activités réalisées (*Activity to perform*) : Description des rôles et des procédures nécessaires à la mise en œuvre d'un secteur clé.
- Mesures et analyse (*Measurement and analysis*) : Description du besoin de mesurer le processus logiciel et d'analyser le résultat de ces mesures.
- Vérification de mise en œuvre (*Verifying implementation*) : Actions qui permettent d'assurer que les activités seraient réalisées conformément au processus établi.

Le guide SWEBOK (IEEE Computer Society et ACM Committee, 2000) et ses références bibliographiques seront utilisés comme outil de base pour identifier les risques reliés à la mise en place des secteurs clés du modèle CMM. SWEBOK est conçu pour cerner les caractéristiques de la discipline du génie logiciel. Il décrit les dix domaines de connaissance (*Knowledge Area* ou *KA*) qui font partie des connaissances reliées au domaine du génie logiciel (*Body of Knowledge*). Ces dix domaines sont : les exigences des logiciels, la conception des logiciels, la construction des logiciels, le test des logiciels, l'entretien des logiciels, la gestion de la configuration, la gestion du génie logiciel, les outils et méthodes en génie logiciel, le processus logiciel et finalement la qualité du logiciel. Le tableau ci-dessous représente le sommaire des activités de recherche pour identifier les facteurs de risques appartenant aux trois secteurs clés de CMM.

Tableau 3.4 **Planification du projet, étape 2**

II. Planification du projet		
Étape 2	Intrants	Livrables
Identifier les facteurs de risques appartenant aux trois secteurs clés du niveau 2 de CMM suivants : <ul style="list-style-type: none"> • Gestion des exigences • Suivi et supervision de projet logiciel • Gestion de la configuration logiciel 	<ul style="list-style-type: none"> • (Paulk et al, 1995) The Capability Maturity Model : Guidelines for Improving the Software Process • (IEEE Computer Society et ACM Committee, 2000) Guide to the Software Engineering Body of Knowledge, A Stone Man Version (Version 0.7)² 	Facteurs de risques relatifs à l'avancement au niveau 2 de CMM des trois secteurs clés (gestion des exigences, suivi et supervision du projet logiciel et gestion de la configuration logiciel) répartis chacun dans les catégories suivantes : <ul style="list-style-type: none"> • Risques reliés à l'engagement de réalisation • Risques reliés à la capacité d'exécution • Risques reliés aux activités réalisées • Risques reliés aux métriques et analyse • Risques reliés à vérification de mise en œuvre

3.3 Étape 3 : Concevoir une grille d'évaluation des facteurs de risque des programmes d'amélioration du processus logiciel pour les organisations désirant progresser du niveau 1 au niveau 2 du modèle CMM

Les deux documents intitulés “*A Risk Assessment Method and Grid for Software Measurement Programs*” (Abran, Laframboise et Bourque, 1999) et “*Une méthode d'analyse du risque de projets de nouvelle technologie informatique*” (Girard, 1998) seront utilisés dans la phase de construction de la grille de risques.

² La version 0.7 représente la version la plus récente de SWEBOK au moment même de la rédaction de ce document.

Dans (Abran, Laframboise et Bourque, 1999), les auteurs sont motivés par le fait que malgré la mise en place des programmes de mesure en génie logiciel, les risques d'échec en informatique demeurent particulièrement élevé dans les entreprises. Selon eux, il n'existe aucun outil d'évaluation du risque pour assister les entreprises lors de la mise en place d'un programme de mesures. Par conséquent, il est nécessaire de mener une étude exploratoire afin de développer une méthode visant à évaluer les facteurs de risques reliés à la mise en place d'un programme de mesure en génie logiciel. Les résultats sont potentiellement utiles aux entreprises désirant implanter un programme de mesures en génie logiciel, plus particulièrement les groupes d'assurance de qualité ou de support méthodologique des grandes entreprises.

Dans la deuxième étude, l'auteur Girard (Girard, 1998) s'intéresse particulièrement à la gestion de risque des projets de nouvelle technologie informatique. Selon lui, les principes de gestion de risques des projets de développement en informatique ne sont pas tout à fait applicables à des projets d'innovation technologique. Il n'en demeure pas moins que les risques d'échec sont encore beaucoup plus élevés pour les entreprises qui s'orientent vers une technologie innovatrice. L'auteur mène une étude exploratoire pour développer une méthode d'évaluation des risques associés à un projet d'innovation technologique dans une entreprise. Le but de cette recherche est de structurer une méthode d'évaluation du risque de manière à la rendre flexible et capable d'intégrer de nouvelles caractéristiques et des contraintes spécifiques aux nouvelles technologies.

Selon l'auteur, la gestion du risque ne se limite pas uniquement à l'utilisation de la grille d'évaluation du risque. Elle fait partie d'un processus plus complet qui inclut les activités suivantes : la validation, la sélection, la grille d'évaluation elle-même et le contrôle.

Le tableau suivant représente le sommaire des activités de recherche pour concevoir une grille de gestion de risques.

Tableau 3.5 Planification du projet, étape 3

Étape 3	Intrants: revue de littérature	Livrables
<ul style="list-style-type: none"> • Concevoir une grille d'évaluation des facteurs de risque des programmes d'amélioration du processus logiciel pour les organisations désirant progresser du niveau 1 au niveau 2 du modèle CMM 	<ul style="list-style-type: none"> • Une Méthode d'Analyse du Risque de Projets de Nouvelle Technologie Informatique (Girard, 1998) • (Abran, Laframboise et Bourque, 1999) A Risk Assessment Method and Grid for Software Measurement Programs • Facteurs de risques génériques relatifs au modèle CMM à l'étape 1. • Facteurs de risques relatifs à l'avancement au niveau 2 de CMM des trois secteurs clés à l'étape 2. 	<ul style="list-style-type: none"> • Grille d'évaluation des facteurs de risque des programmes d'amélioration du processus logiciel pour les organisations désirant progresser du niveau 1 au niveau 2 du modèle CMM (Version initiale)

3.4. Étape 4 : Révision de la grille par les experts

Les experts seront tous invités à une présentation initiale du projet. Nous leur remettrons la grille pour fin de révision accompagnée du document de recherche. Les réviseurs nous retourneront la grille par courrier électronique avec les commentaires contenus dans un document en format Word.

La grille de gestion de risque sera révisée par les trois experts suivants :

- Richard Basque intervient depuis plus de 25 ans en informatique, à la suite d'une formation spécialisée complétée en 1973 à l'Université de Montréal. Il est accrédité par le *Software Engineering Institute* (SEI) comme chef évaluateur et comme instructeur pour le cours sur le modèle CMM.
- Claude Y. Laporte a complété, en 1973, un baccalauréat en physique et mathématiques au Collège militaire royal de Saint-Jean. En 1980, il compléta une maîtrise en physique à l'université de Montréal puis, en 1986, une maîtrise en sciences appliquées au département de génie électrique et génie informatique de l'école Polytechnique de Montréal. Il a été officier au sein des Forces armées canadiennes et professeur pendant plus de 10 ans au Collège militaire royal de Saint-Jean. De 1988 à 1992, il a travaillé à la mise sur pied du Centre de génie logiciel appliqué situé au CRIM. En 1992, il a joint Oerlikon Aérospatiale où il s'occupait du développement et de la mise en œuvre des processus de génie logiciel, de génie des systèmes et de gestion. En 1999 il a formé sa propre société conseil. En plus d'offrir de l'expertise en ingénierie des processus, il offre plusieurs ateliers de formation d'équipe et de démarrage de projets. Il préside le groupe d'intérêt en amélioration des processus le SPIN de Montréal.

- Daniel Dutil ??????????????????????

3.3.1 Lignes directrices relatives à la revue des documents

Pour la phase de révision de la grille, nous adopterons la même approche utilisée lors de la révision de SWEBOK. Voici les instructions qui seront adressées aux réviseurs :

1. Cette révision est publique signifiant que les identités des réviseurs, leurs commentaires ainsi que les résolutions de ces commentaires seront publiés dans la version finale de l'activité de synthèse.
2. La formulaire de révision ayant les changements requis doit contenir des informations adéquates pour pouvoir les implanter. Cela signifie que les commentaires doivent être formulés clairement pour indiquer la façon à aborder le changement en question. Les commentaires bien formulés génèrent plus d'attention et de crédibilité que ceux qui sont vagues et imprécis. Ces derniers seront rejetés.

Le réviseur est prié de lister les demandes de changement ou les commentaires d'une manière séquentielle en se basant sur le numéro du paragraphe pour faciliter leur classification et leur

fusion. Les paragraphes de la grille destinée aux réviseurs seront numérotés dans un ordre

3. séquentiel croissant.
4. Pour chaque commentaire ou demande de changement, le réviseur doit indiquer le numéro de paragraphe correspondant ou le numéro de la figure ou du tableau si c'est le cas.
5. Quoique les critiques du contenu de la grille constituent l'objectif principal de la révision, tout autre commentaire qui pourra rendre le produit final plus facile à lire et toute autre suggestion ou demande de changement demandant des clarifications supplémentaires seront aussi considérés.
6. Le réviseur est prié de rapporter les inconsistances à l'intérieur de la grille ou les incohérences au niveau de la terminologie.

Tableau 3.6 Planification du projet, étape 4

Étape 4	Intrants	Livrables
<ul style="list-style-type: none"> • Révision de la grille par les experts 	<ul style="list-style-type: none"> • Grille d'évaluation des facteurs de risque des programmes d'amélioration du processus logiciel pour les organisations désirant progresser du niveau 1 au niveau 2 du modèle CMM (Version initiale) 	<ul style="list-style-type: none"> • Grille d'évaluation des facteurs de risque des programmes d'amélioration du processus logiciel pour les organisations désirant progresser du niveau 1 au niveau 2 du modèle CMM (Version révisée)

4 TABLEAU RÉCAPITULATIF

Le tableau suivant représente la récapitulation des activités de recherche de la conception d'une grille d'évaluation des facteurs de risque des programmes d'amélioration du processus logiciel pour les organisations désirant progresser du niveau 1 au niveau 2 selon le modèle CMM

Tableau 4.7 Planification du projet, récapitulatif

I. Définition du projet			
Motivation	Objet	Objectif	Utilisateurs
<ul style="list-style-type: none"> Réduire les échecs des projets informatiques 	<ul style="list-style-type: none"> Amélioration du processus logiciel en se basant sur le modèle CMM. 	<ul style="list-style-type: none"> Développer une grille d'évaluation des facteurs de risque des programmes d'amélioration du processus logiciel pour les organisations désirant progresser du niveau 1 au niveau 2 selon le modèle CMM 	<ul style="list-style-type: none"> Équipes de coordination de l'implantation du processus logiciel (<i>software engineering process groups</i>)
II. Planification du projet			
Étape 1	Intrants	Livrables	
Modèle I.D.E.A.L : <ul style="list-style-type: none"> Déterminer les risques génériques reliés au projet d'implantation de CMM 	<ul style="list-style-type: none"> (McFeeley, 1996) I.D.E.A.L., Software Engineering Process Group Guide 	Facteurs de risques génériques relatifs au modèle CMM répartis dans les catégories suivantes : <ul style="list-style-type: none"> Risques reliés à l'engagement de la direction. Risques reliés à la communication avec les participants du projet d'amélioration du processus logiciel Les risques reliés à l'établissement des objectifs d'affaires Les risques reliés à l'infrastructure. 	
Étape 2	Intrants	Livrables	
Déterminer les facteurs de	<ul style="list-style-type: none"> (Paulk et al, 1995) The 	Facteurs de risques relatifs à	

<p>risques appartenant aux trois secteurs clés du niveau 2 de CMM suivants :</p> <ul style="list-style-type: none"> • Gestion des exigences • Suivi et supervision de projet logiciel • Gestion de la configuration logiciel 	<p>Capability Maturity Model : Guidelines for Improving the Software Process</p> <ul style="list-style-type: none"> • (IEEE Computer Society et ACM Committee, 2000) Guide to the Software Engineering Body of Knowledge, A Stone Man Version (Version 0.7)³ 	<p>l'avancement au niveau 2 de CMM des trois secteurs clés choisis, réparti chacun dans les catégories suivantes :</p> <ul style="list-style-type: none"> • Risques reliés à l'engagement de réalisation • Risques reliés à la capacité d'exécution • Risques reliés aux activités réalisées • Risques reliés aux métriques et analyse • Risques reliés à vérification de mise en œuvre
Étape 3	Intrants	Livrables
<ul style="list-style-type: none"> • Concevoir une grille d'évaluation des facteurs de risque des programmes d'amélioration du processus logiciel pour les organisations désirant progresser du niveau 1 au niveau 2 du modèle CMM 	<ul style="list-style-type: none"> • Une Méthode d'Analyse du Risque de Projets de Nouvelle Technologie Informatique (Girard, 1998) • (Abran, Laframboise et Bourque, 1999) A Risk Assessment Method and Grid for Software Measurement Programs • Facteurs de risques génériques relatifs au modèle CMM à l'étape 1. • Facteurs de risques relatifs à l'avancement au niveau 2 de CMM des trois secteurs clés à l'étape 2. 	<ul style="list-style-type: none"> • Grille d'évaluation des facteurs de risque des programmes d'amélioration du processus logiciel pour les organisations désirant progresser du niveau 1 au niveau 2 du modèle CMM (Version initiale)
Étape 4	Intrants	Livrables
<ul style="list-style-type: none"> • Révision de la grille par les experts 	<ul style="list-style-type: none"> • Grille d'évaluation des facteurs de risque des programmes d'amélioration du processus logiciel pour les organisations désirant progresser du niveau 1 au niveau 2 du modèle CMM (Version initiale) 	<ul style="list-style-type: none"> • Grille d'évaluation des facteurs de risque des programmes d'amélioration du processus logiciel pour les organisations désirant progresser du niveau 1 au niveau 2 du modèle CMM (Version révisée)

³ La version 0.7 représente la version la plus récente de SWEBOK au moment même de la rédaction de ce document.

5 ÉCHÉANCIER

La présente recherche, se faisant sous la supervision du professeur Pierre Bourque, a débuté en septembre 1999. Dans le tableau suivant, se présentent les différentes étapes du projet avec leur échéance respective.

Tableau 5.6 Échéancier de l'activité de synthèse de Vinh Duong

ID	Activité	Date de début aaaa-mm-jj	Date de fin aaaa-mm-jj	Durée (heures)
1	Présentation de l'activité de synthèse	2000-11-15	2000-11-15	1
2	Déterminer les facteurs de risques génériques	2000-11-16	2000-12-01	36
3	Déterminer les facteurs de risques reliés aux secteurs clés	2000-12-02	2000-12-31	27
4	Concevoir la grille d'évaluation	2001-01-01	2001-01-31	27
5	Révision de la grille par les experts	2001-02-01	2001-02-15	18
6	Rédaction du rapport final	2001-02-16	2001-03-15	27
7	Présentation du rapport final	2001-04-01	2001-04-01	1
8	Correction du rapport final	2001-04-02	2001-04-07	8
9	Dépôt du rapport final	2001-04-08	2001-04-08	

RÉFÉRENCES

- Abran, Alain, Lucie Laframboise et Pierre Bourque. 1999. «A Risk Assessment Method and Grid for Software Measurement Programs ». Rapport technique no 99-03, Software Engineering Management Research Laboratory, Montréal, Université du Québec à Montréal
- Abran, Alain, Jean-Marc Desharnais, Marcela Maya, Denis St-Pierre et Pierre Bourque. 1998. «Design of a Functional Size Measurement for Real-Time Software». Rapport de recherche, no 13, 23 novembre, Software Engineering Management Research Laboratory, Université du Québec à Montréal, SELAM, Software Engineering Laboratory in Applied Metrics.
- Basili, Victor R. et Richard W. Selby, David H. Hutchens. 1986. «Experimentation in Software Engineering». IEEE Transactions on Software Engineering, vol. SE.12, no 7.
- Basque, Richard. 1998. «ARTICLE NO. 1, le CGLA: un centre de référence qui collabore avec le SEI pour un transfert technologique plus efficace en matière d'amélioration de processus logiciel auprès de notre communauté informatique». *L'INFO-QUÉBEC*, revue de la FIQ (Fédération de l'Informatique du Québec).
- Basque, Richard. 1999. Commentaires provenant de l'instructeur Richard Basque pendant le cours *Introduction au CMM* donné au CGLA, à Montréal, du 20 au 22 juillet.
- Bombardier. 1999. Communiqué de presse intitulé «Bombardier obtient le contrat de fourniture des trains de banlieue de la régie de transport MTA/Long Island Rail Road>>. www.bombardier.com, St-Bruno.
- Caputo, Kim. 1998. *CMM Implementation Guide*. Reading, Mass :Addison Wesley Longman, Inc.
- CMM. 1999. «Process Maturity Profile of the Software Community : 1999 Mid-Year Update». Software Engineering Institute. Pittsburg, Carnegie Mellon University.
- Fowler, Priscilla et Stan Rifkin. 1990. «Software Engineering Process Group Guide». Technical Report. Software Engineering Institute. Pittsburg, Carnegie Mellon University.
- Girard, Daniel. 1998. «Une Méthode d'Analyse du Risque de Projets de Nouvelle Technologie Informatique». Rapport final d'activité de synthèse de la Maîtrise en informatique de gestion, Montréal, Université du Québec à Montréal.
- Goldenson, Dennis R. et James D. Herbsleb. 1995. «After the Appraisal: A Systematic Survey of Process Improvement, its Benefits, and Factors that Influence Success». Technical Report SEI-95-TR-009. Software Engineering Institute, Pittsburg, Carnegie Mellon University.

- Hall, Elaine M. 1998. *Managing Risk: Methods for Software Systems Development*. Reading, Mass. : Addison-Wesley.
- Hayes, Will et Dave Zubrow. 1995. «Moving On Up: Data and Experience Doing CMM-Based Process Improvement». Technical Report SEI-95-TR-008. Software Engineering Institute Pittsburg, Carnegie Mellon University.
- Hetzel, Bill. 1993. *Making Software Measurement Work : Building an Effective Measurement Program*. QED Publishing Group.
- Humphrey, Watts. 1990. *Managing the software process*. SEI series in software engineering. Reading, Mass. ; Don Mills, Ont. : Addison-Wesley.
- Krasner, Herb. 1994. «The Payoff for Software Process Improvement (SPI): What it is and How to get it». Software Process Newsletter, IEEE Computer Society, no 1, September, p. 3-8
- Laframboise, Lucie. 1996. «Grille d'Évaluation des Facteurs de Risque d'un Programme de Mesures en Génie Logiciel». Rapport final d'activité de synthèse de la Maîtrise en informatique de gestion, Montréal, Université du Québec à Montréal.
- McFeeley, Bob. 1996. «IDEAL: A User's Guide for Software Process Improvement». Technical Report SEI-96-HB-001. Software Engineering Institute. Pittsburg, Carnegie Mellon University.
- McGarry, Frank E. 1995. «Product-Driven Process Improvement». Computer Sciences Corporation, Software Process Newsletter, IEEE Computer Society, No. 3, Spring.
- Paulk, Marc C., Bill Curtis, Mary Beth Chrissis et Charles V. Weber. 1993a. «Modèle d'Évolution des Capacités Logiciel». Version 1.1. Richard Basque, coordonnateur pour le projet de traduction. Traduction de l'anglais réalisée par le Centre de génie logiciel appliqué, une division du Centre de recherche informatique de Montréal avec le parrainage du gouvernement du Québec (Ministère de l'Industrie, du Commerce, de la Science et de la Technologie) et du gouvernement du Canada (Industrie Canada). Rapport technique CMU-SEI-93-TR-24. Software Engineering Institute. Pittsburg, Carnegie Mellon University.
- Paulk , Marc C., Charles V. Weber, Bill Curtis et Mary Beth Chrissis. 1993b. «Pratiques du Modèle d'Évolution des Capacités Logiciel». Version 1.1. Richard Basque, coordonnateur pour le projet de traduction. Traduction de l'anglais réalisée par le Centre de génie logiciel appliqué, une division du Centre de recherche informatique de Montréal avec le parrainage du gouvernement du Québec (Ministère de l'Industrie, du Commerce, de la Science et de la Technologie) et du gouvernement du Canada (Industrie Canada). Rapport technique CMU-SEI-93-TR-25. Software Engineering Institute. Pittsburg, Carnegie Mellon University.

- Paulk, Marc C., Charles V. Weber, Bill Curtis et Mary Beth Chrissis. 1995. *The Capability Maturity Model : Guidelines for Improving the Software Process*. SEI series in software engineering. Reading, Mass. ; Don Mills, Ont. : Addison-Wesley, 461 p.
- Pressman, Roger. 1988. *Making Software Engineering Happen : A Guide for Instituting the Technology*. Englewood Cliffs, N.J. : Prentice-Hall, 258 p.
- Sekaran, Uma. 1992. *Research Methods for Business : A Skill-Building Approach*. - 2nd ed. New York ; Toronto : J. Wiley, 428 p.
- Sisti, Frank et Joseph Sujoe. 1994. «Software Risk Evaluation Method». Version 1.0. Technical Report SEI-94-TR-19. Software Engineering Institute. Pittsburg, Carnegie Mellon University.
- SPICE, 1998. «Phase 2 Trials Interim Report». Version 1.00. Initié par International Standards group for Software Engineering, ISO/IEC JTC1/SC7.
- Standish Group. 1995. «The CHAOS report». Sample Research Paper. The Standish Group.
- IEEE Computer Society et ACM Committee. 2000. «Guide to the Software Engineering Body of Knowledge : A Stone Man Version (Version 0.7) ». A project of the Software Engineering Coordinating Committee (Joint IEEE Computer Society – ACM committee). Executive Editors : Alain Abran, Université du Québec à Montréal, James W. Moore, The MITRE Corp.; Editors : Pierre Bourque, Université du Québec à Montréal, Robert Dupuis, Université du Québec à Montréal; Chair of the Software Engineering Coordinating Committee : Leonard L. Tripp, IEEE Computer Society.
- Thomson, H.E. 1998. «Software Process Assessment and Improvement». Chap. 9 In *The SwIM - a practical approach for software process improvement*. Ed. by T. Rout. University of East Anglia. Norwich (England) : Computational Mechanics Publications.
- Wieggers, Karl Eugene. 1996. *Creating a Software Engineering Culture*. New York : Dorset House, 358 p.
- Wolff, Sibylle. 1999. «La Place de la Mesure au Sein des Principes Fondamentaux du Génie Logiciel» Rapport final d'activité de synthèse de la Maîtrise en informatique de gestion, Montréal, Université du Québec à Montréal.
- Zahran, Sami. 1998. *Software Process Improvement : Practical Guidelines for Business Success*. New York : Dorset House, 358 p.