

Chapitre 4

Amélioration du processus d'évolution et de maintenance logicielle

4.1. Introduction

Ce chapitre introduit les concepts fondamentaux de l'amélioration des processus d'évolution et de maintenance du logiciel. Le chapitre illustre comment les principes de l'amélioration des processus peuvent être mis en œuvre à l'aide d'un modèle d'amélioration des processus adapté à ce contexte spécifique. Ce chapitre s'inscrit dans le cadre d'une problématique globale portant sur l'amélioration de la qualité en technologies de l'information. Débutons en situant l'évolution et la maintenance du logiciel dans le contexte d'entreprise.

4.1.1. Les différences entre infrastructures/opération, développement et maintenance du logiciel

Pour bien situer les modèles d'amélioration des processus adaptés à la maintenance il est nécessaire de situer la maintenance telle qu'elle est vécue quotidiennement dans nos sociétés. Il est tout d'abord nécessaire d'établir une distinction claire entre l'infrastructure/l'opération d'un logiciel et sa maintenance. Ainsi, il est précisé dans la norme internationale de la maintenance du logiciel ISO/CEI 14764 [ISO 06] que les activités d'infrastructure/opération (copies de sécurité, recouvrement et administration des ordinateurs) sont effectuées par le personnel d'infrastructure/opération des systèmes informatiques et sont exclues de la portée de la norme ISO sur la maintenance des logiciels. Il y a donc une différence assez importante entre les infrastructures/opérations informatiques et la maintenance

2 Titre de l'ouvrage

pour que deux normes internationales soient créées pour chacune : la ISO/CEI 14764 pour la maintenance et la ISO/CEI 20000 [ISO 05] pour les infrastructures/opérations. Cette distinction de concepts est aussi exprimée dans la littérature qui décrit qu'il est peu commun qu'un gestionnaire confonde les activités d'infrastructures/opérations informatiques et celles de la maintenance des logiciels. Bien sur il existe une interface importante entre la maintenance et les infrastructures/opérations qui vise surtout à s'assurer que les infrastructures, qui supportent les logiciels applicatifs, soient opérationnelles et efficaces (i.e. processus de gestion des changements, discussions concernant une défaillance en production, processus de recouvrement du logiciel et des données, et reprise des travaux suite à une panne ou à un désastre, activités d'investigation des temps de traitements, copies de sécurité, gestion des systèmes d'horaires automatisées, gestion de l'espace disque et de bandothèques). La gestion de cette interface est un rôle principal de mainteneurs. Le référentiel de pratiques exemplaires le plus utilisé pour l'amélioration des processus d'infrastructures/opérations est celui de l'Information Technology Infrastructure Library (ITIL) qui a été synthétisé dans la norme ISO/CEI 2000.

Qu'en est-il des différences entre la maintenance et le développement du logiciel ? Le développement du logiciel possède, lui aussi, une interface avec la maintenance, mais elle peut être un peu plus difficile à différencier. Cette différenciation est encore plus difficile à observer si, dans une organisation, le développeur du logiciel effectue aussi la maintenance de celui-ci. En effet, un certain nombre des activités de la maintenance sont similaires à celles du développement de logiciels (analyse, conception, codage, gestion de la configuration, essais, revues et documentation technique). Ces activités doivent toutefois être adaptées au contexte spécifique de la maintenance, où le travail est souvent effectué par un ou deux programmeurs pour des travaux exécutés à très court terme, et non par une équipe de projet comme dans la majorité des projets de développement. Un employé de la maintenance du logiciel développera donc une partie de son expertise à partir des mêmes sources d'enseignement et de formation que ses collègues du développement. Khan [Kha 05] décrit que certains processus de maintenance sont uniques et que ces processus ne font pas partie de la fonction de développement de logiciels. Le CMMi [Bas 04], dont le centre d'intérêt porte sur la gestion de projet, ne traite pas des problèmes spécifiques de la maintenance quotidienne de logiciels telle que vécue dans les sociétés:

- les processus uniques à la maintenance ne sont pas traités adéquatement ;
- le concept de file d'attente et de requête de la maintenance n'est ni reconnu ni traité ;
- il n'y a pas suffisamment d'inclusion de pratiques spécifiques à la maintenance comme mécanismes d'amélioration de processus ;

- les plans reliés au rajeunissement du logiciel (c'est-à-dire redocumentation, réingénierie, ingénierie inverse, migration de logiciel et mise à la retraite) ne sont pas traités de façon satisfaisante.

Lorsque les processus uniques à la maintenance sont comparés aux domaines de processus du modèle CMMi, on peut observer que le modèle CMMi ne traite pas explicitement des particularités de la maintenance. Dans son livre de vulgarisation du CMMi, Basque décrit comment ce modèle ne couvre pas la petite maintenance [Bas04, p. 25], le modèle CMMi ayant ses limites d'interprétation dans un contexte d'activités continues qui n'ont pas de date de début ou de date de fin. Par contre, les projets de maintenance nécessitant une structure de gestion de projet devraient utiliser le CMMi comme référentiel pour l'amélioration. D'autre part, les évaluations et les améliorations de petites activités de maintenance qui n'utilisent pas de techniques de gestion de projet devraient utiliser des modèles de maturité de la maintenance mieux adaptés à ce domaine spécifique.

Quelques auteurs ont étudié les activités qui sont uniques à la maintenance et qui ne se retrouvent pas dans un cycle de vie de développement de logiciels. Certaines des caractéristiques qui sont propres au domaine de la maintenance du logiciel sont [Apr 06] :

- « les requêtes de modifications (RM) parviennent d'une manière plus ou moins aléatoire et ne peuvent pas être planifiées individuellement dans un processus annuel de budgétisation ;
- les RM sont évaluées et classées par ordre de priorité (par le programmeur ou son patron, ou parfois par un comité de priorités) ; aucune RM ne fait l'objet d'une autorisation spécifique par un gestionnaire en chef ;
- la charge de travail de la maintenance n'est pas gérée par des techniques de gestion de projet, mais plutôt par l'utilisation des techniques de gestion des files d'attente, souvent supportées par un logiciel de bureau d'aide help desk ;
- la taille et la complexité des requêtes de la maintenance font en sorte que le travail peut être, généralement, effectué par un ou deux programmeurs ;
- les travaux sont ordonnés de manière à satisfaire l'utilisateur, à court terme, et à s'assurer du bon fonctionnement quotidien des logiciels en opérations ;
- les priorités peuvent changer rapidement, à toute heure du jour, et les rapports de problèmes (RP) nécessitant des corrections de l'application de production auront la priorité sur n'importe quel autre travail en cours. »

L'équipe de maintenance doit faire face aux événements et aux requêtes journalières de sa clientèle tout en maintenant un service continu des applications opérationnelles sous sa responsabilité.

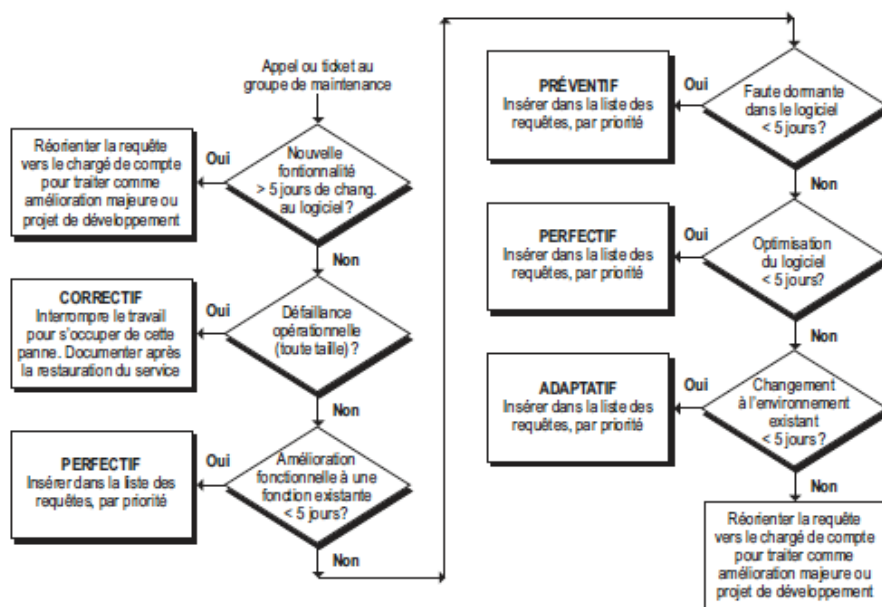


Figure 1.1. Exemple de processus d'acceptation et de refus d'une requête en maintenance

L'organisation des travaux de maintenance doit tenir compte de ces impératifs, et tant les travaux que l'équipe de maintenance doivent se structurer pour répondre aux exigences de ce type de travail dont les éléments individuels arrivent de façon aléatoire. Par contraste, un projet de développement est prévu à plus long terme, planifié, typiquement créé avec un échéancier déterminé, puis s'achève avec la livraison du logiciel. Pour bien mener à terme un projet, une structure d'équipe de projet est créée pour la durée du projet seulement, et c'est cette équipe de projet qui développe un plan de ressources qui vise à atteindre des objectifs fixes d'une livraison dans les limites du délai planifié de fin du projet.

Lorsque l'utilisateur fait une requête de modification, il est nécessaire d'estimer l'effort qui sera requis pour modifier le logiciel existant. L'étude de Dorfman et de Thayer [Dor 02] indique que les requêtes de modification (RM) passent par une étape d'investigation et d'impact qui est unique à la maintenance du logiciel. Si l'effort estimé est peu élevé et peut être satisfait à l'intérieur des ressources et des disponibilités de l'équipe de maintenance, la requête est alors placée en ordre de priorité et exécutée à partir du processus de la gestion de la file d'attente. Toutefois, si l'effort estimé est supérieur à une limite établie, propre à chaque organisation, et à sa marge de manœuvre budgétaire, la requête sera réacheminée à une équipe de

développement et traitée comme un petit projet qui devra être autorisé avec son budget spécifique et une allocation précise des ressources requises pour le mener à terme en fonction des nouveaux objectifs fixés.

Il y a donc, pour la maintenance du logiciel, un processus unique d'acceptation ou de rejet du travail, pour les requêtes de modifications (RM) des logiciels opérationnels. Ce processus tient compte de la taille estimée de la modification et de la capacité à la réaliser à l'intérieur des contraintes de coûts et de qualité de fonctionnalité. April [Apr 08] présente comme exemple le processus utilisé par une société membre de Cable & Wireless, où l'effort maximal d'une requête adaptative qu'un programmeur de la maintenance accepte est de cinq jours (figure 1.2). On y note que tout rapport de problème (RP), quel que soit l'effort estimé, sera pris en charge par l'équipe de maintenance en priorité avant les requêtes de modifications.

Cette limite de cinq jours d'effort est aussi reconnue par l'association de la mesure du Royaume-Uni (UKSMA) et le groupe international de normalisation de l'étalonnage du logiciel (ISBSG) : « On observe dans la pratique la distinction entre l'activité de maintenance des améliorations mineures et l'activité de développement du logiciel. Les auteurs ont observé que quelques organisations possèdent des activités de maintenance qui compte jusqu'à 80 jours d'effort, alors que dans d'autres le seuil est de cinq jours. L'ISBSG et l'UKSMA adoptent, actuellement, la convention qu'une amélioration de cinq jours ou moins sera considérée comme une activité de maintenance. » [ISB 04]. Ce seuil correspond à de la petite maintenance effectuée par des individus, et non pas par des équipes de projet qui peuvent entreprendre de grands projets. Cette notion de seuil est très importante, car elle dicte le seuil des travaux entre les développeurs et les mainteneurs. Cette limite de cinq jours ne doit pas être utilisée avec dogmatisme. Ce qui est essentiel, c'est qu'une limite du travail de maintenance soit identifiée clairement dans l'organisation, quelle que soit la valeur d'effort choisie, et reflète bien le travail d'individus, et non pas d'équipes de projet.

Bennett [Ben 00] identifie d'autres activités uniques à la maintenance : « l'étude des différents types de requêtes de changements supportées par un centre d'appel help desk et son logiciel de support, les activités d'évaluation d'impact d'un changement, et la spécialisation en essais et vérification de régression ». Il identifie aussi sept processus clés et certains rôles qui sont spécifiques de la maintenance du logiciel :

- Gestion des requêtes de changement et de demandes de modifications. Un processus de gestion des problèmes utilisé par les mainteneurs pour établir la priorité, documenter et acheminer les demandes qu'ils reçoivent ;

6 Titre de l'ouvrage

- gestion de la transition du développement à la maintenance, qui est une séquence contrôlée et coordonnée d'activités durant laquelle le système est progressivement transféré du développeur vers le mainteneur ;
- rôle des utilisateurs, des infrastructures/opérations et des employés de la maintenance ;
- planification de la maintenance ;
- gestion des employés de la maintenance ;
- gestion du logiciel (améliorations et performances).

La dernière version du chapitre de la maintenance du Guide SWEBOK [SWE 05] identifie aussi un bon nombre de processus spécifiques à la maintenance, soit :

- gestion et planification annuelle de la maintenance ;
- ententes de services et contrats spécifiques de la maintenance ;
- interception et surveillance des applications en production (prévention de problèmes) ;
- mesure d'indicateurs de services spécifiques aux activités du support et de la maintenance ;
- soutien à la clientèle (système de réponses aux problèmes) concernant une panne, une maintenance préventive et un retour en service après panne ;
- analyses d'impact de différents types de requêtes de changements supportées par un centre d'appel help desk et son logiciel de support ;
- spécialisation en essais de régression suite à une modification ;
- investigations et réponses aux questions concernant les règles d'affaires des systèmes opérationnels ;
- processus d'acceptation et de rejet de requêtes de modifications (RM) des logiciels opérationnels selon leur taille ;
- gestion de l'horaire de support aux opérations 24 heures sur 24 et du processus d'escalade en cas de problèmes ;
- gestion de l'interface avec les infrastructures/opérations portant sur : la gestion du changement, les appels concernant une faute en production, le recouvrement de l'environnement et des données suite à une défaillance majeure ou un désastre, le recouvrement des données et reprise des travaux, l'investigation des temps de traitements, les cédules automatisées, les copies de sécurité, la gestion des systèmes, de l'espace disque et de la bandothèque ;
- gestion de la sous-traitance, des contrats de services de maintenance, de licences, d'entièrement et d'impartition ;
- optimisation du portefeuille d'application plus performant (gestion du logiciel).

La définition du service rendu par la maintenance peut aussi aider à identifier des différences avec les autres activités informatiques. Ainsi, Bouman [Bou 99] décrit la maintenance du logiciel comme un service. Les caractéristiques d'un service peuvent généralement être reconnues ainsi :

- « l'insistance de la vente directe avec le client ;
- un contact fréquent et direct avec le client ;
- un service rendu immédiatement plutôt que plusieurs mois après ;
- un temps de service court ;
- le produit n'est pas toujours un bien physique ;
- le produit ne peut pas toujours être remisé ou transporté ;
- les services sont plus spécialisés et façonnés que les biens physiques. »

En résumé, la maintenance du logiciel possède un certain nombre de processus et d'activités qui ne sont pas effectués par les groupes de développement et d'infrastructures/opérations. Bien que la maintenance du logiciel fasse appel à des processus et à des activités similaires au développement du logiciel elle est profondément différente. Des référentiels de pratiques exemplaires ont donc été développés spécifiquement pour ce domaine. La prochaine section fait la synthèse des modèles d'amélioration des processus qui ont été proposés en génie logiciel et identifie ceux qui peuvent contribuer à l'amélioration des processus d'évolution et de la maintenance du logiciel.

4.1.2. Un inventaire des modèles d'amélioration des processus du logiciel

Depuis le milieu des années 80, plusieurs modèles de maturité ont été proposés autant par l'industrie que par le milieu de la recherche en génie logiciel. Le concept devient tellement populaire que des propositions de modèles de maturité vont être développées pour des secteurs autres que celui du logiciel, par exemple la gestion des ressources humaines, la gestion financière et les soins primaires.

Sarah Sheard [She 97] présente, à la figure 1.2, les normes, modèles et méthodes d'évaluation les plus connus aux États-Unis, ainsi que les principales normes qui leur sont pertinentes. Son schéma représente le point de vue historique des influences qui ont mené à l'évolution de tous les modèles (d'un point de vue Américain) d'amélioration des processus de logiciels.

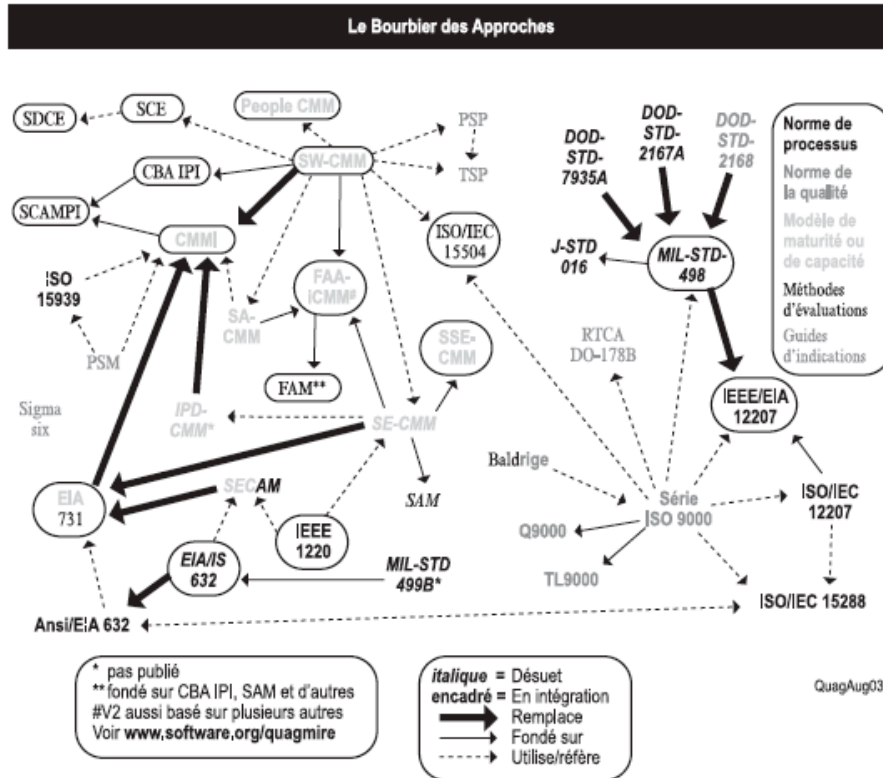


Figure 1.2. Le bourbier des modèles et normes du processus logiciel

Des modèles sont continuellement supplantés par de nouvelles propositions, et ceux non tenus à jour, tombent rapidement dans la désuétude. Il en ressort que seuls quelques modèles deviennent plus populaires et survivent grâce à des investissements plus importants pour les tenir à jour, en faire la promotion et les améliorer. Deux seuls modèles sont centrés sur la maintenance du logiciel. Ces deux modèles, le CM3 [Kaj 01] et le S3m [Apr 06], peuvent contribuer à améliorer la maintenance du logiciel et leurs pratiques détaillées et sont accessibles publiquement. Depuis quelques années le modèle S3m (disponible au www.s3m.ca), inclus les pratiques de plusieurs autres modèles, dont le CM3, ce qui le rend plus facile à utiliser et populaire auprès de centaines de sociétés en Amérique et en Europe .

4.2. Aperçu du modèle S3m

L'objectif du modèle S3m est d'offrir aux organisations de maintenance de logiciels un moyen permettant de démarrer et de guider un programme d'amélioration continue spécialisé pour la maintenance du logiciel.

L'architecture du modèle S3m est fondée sur celle du modèle CMMi pour le logiciel. Cet arrimage au modèle CMMi permet ainsi de s'y référer plus facilement et d'utiliser certains documents existants du SEI, lorsqu'ils sont pertinents, pour mieux comprendre certaines activités de la maintenance du logiciel. Ce modèle a aussi adopté la représentation continue qui prévaut dans l'ISO/CEI 15504 [ISO 08] et le CMMi. Cela permet plus facilement : a) de se conformer à la norme ISO/CEI 15504 ; b) d'obtenir une approche et une cote plus détaillée pour chaque domaine, itinéraire et facette du modèle; c) de facilement situer une pratique à travers les niveaux de maturité pour en démontrer la progression du niveau 0 (incomplet) aux niveaux supérieurs visés.

Pour la cueillette d'informations sur les processus de la maintenance du logiciel, l'outil sélectionné a été un questionnaire. Ce questionnaire utilise directement les pratiques décrites dans le modèle S3m. Ce questionnaire est aussi utilisé pour évaluer la capacité d'un fournisseur d'impartition ou d'un sous-traitant de la maintenance du logiciel. On peut aussi utiliser le questionnaire pour faire une autoévaluation de l'état de la maintenance du logiciel dans sa propre société.

Le modèle S3m, la méthode d'évaluation et les différents instruments de cueillette, dont le questionnaire, sont utilisés ensemble pour toutes sortes d'objectifs, par exemple :

- un guide de bonnes pratiques de la maintenance ;
- une source d'idées et d'inventaire des processus à définir ou à améliorer ;
- un référentiel pour l'étalonnage des performances du processus de maintenance de logiciels d'une organisation par rapport aux pratiques exemplaires en industrie ;
- un outil d'aide dans le choix d'un fournisseur dans le cadre des négociations précontractuelles pour de la sous-traitance ou de l'impartition de la maintenance;
- un guide pour déterminer des occasions d'amélioration au sein d'une organisation de maintenance du logiciel selon un mode d'autoévaluation.

Le modèle S3m et les outils qui l'accompagnent ne constituent eux-mêmes ni un processus ni un modèle de cycle de vie de la maintenance du logiciel. Le modèle S3m inventorie et réfère aux pratiques exemplaires en maintenance du logiciel. Ces pratiques exemplaires peuvent être utilisées pour améliorer le processus ou le cycle

de vie et les services de la maintenance existant. On se doit d'utiliser le jugement professionnel pour les interprétations reliées aux circonstances spécifiques de son environnement. Les pratiques peuvent être adaptées mais il est important de conserver leur intention et l'objectif initial.

4.2.1. Portée organisationnelle

Le modèle S3m est basé sur l'hypothèse que toute organisation vise avant tout à satisfaire ses clients à atteindre ses objectifs stratégiques. Les clients perçoivent souvent un fournisseur comme une boîte noire (par exemple, un département : a) de commercialisation ; b) des ventes ; c) de soutien à la clientèle, etc.). Dans le cadre d'une évaluation basée sur ce modèle, il faut faire abstraction des frontières organisationnelles internes et il faudra évaluer l'ensemble des services de la maintenance du logiciel, en considérant cet ensemble comme une unité autonome qui agit comme un centre de profits afin de démontrer sa valeur stratégique et commerciale.

4.2.2. Perspective et objectifs du modèle

Le modèle S3m a été développé selon le point de vue du client, tel que perçu dans un environnement commercial concurrentiel. Dans ce contexte, la capacité se définit comme suit :

« L'aptitude de l'organisation à fournir de façon constante un service de la maintenance du logiciel qui répond aux critères suivants :

- le respect des attentes et des priorités du client ;
- l'utilisation des pratiques exemplaires de l'industrie de la maintenance du logiciel ;
- des coûts de services transparents et compétitifs ;
- des délais de services les plus courts possible. »

L'évaluation selon le modèle S3m permet d'établir un programme d'amélioration dont l'objectif ultime est la satisfaction du client et l'atteinte d'objectifs stratégiques plutôt qu'une conformité rigide aux critères documentés dans le présent document et ses documents de référence. Afin d'atteindre le plus haut niveau de capacité possible, l'organisation de la maintenance devrait chercher à :

- s'assurer que le principe d'amélioration continue fasse partie de sa culture et que les mécanismes et les processus associés à cette culture soient mis en place ;

- concevoir et à maximiser le processus de la maintenance du logiciel pour que ce dernier permette de respecter les exigences et objectifs définis pour le produit ou le service de maintenance visé ;
- maximiser de façon continue l'utilisation des ressources engagées dans l'exécution des services de la maintenance du logiciel.

4.2.3. Avantages de l'utilisation du modèle pour les mainteneurs

Du point de vue du client de la maintenance, un niveau de capacité plus élevé signifie que l'organisation offrant les services de maintenance du logiciel répond mieux à ses exigences et à celles du marché, par exemple :

- les services et prestations sont clairs et font l'objet d'une entente de services ;
- les coûts et les temps d'attente sont rapportés et minimisés ;
- l'organisation procure un taux maximal de satisfaction.

Pour l'organisation de maintenance, un niveau de capacité plus élevé se traduira par les résultats suivants :

- des coûts de maintenance expliqués à la clientèle et alloués clairement à chaque service de la maintenance selon les termes de l'entente de services ;
- des requêtes et des activités placées en ordre de priorité, suivies et supervisées, ce qui permet de bien répondre aux niveaux de services convenus par l'entente de services ;
- les meilleures pratiques de l'industrie sont utilisées, vérifiables et promues résultant en une approche systématique de travail dont les employés et le ministère, l'organisation, la société ou l'entreprise peuvent être fiers de démontrer les avantages à leur clientèle ;
- une capacité croissante d'atteindre, pour ce qui est de la qualité, des objectifs quantifiables à toutes les étapes du processus de maintenance du logiciel.

4.2.4. La mise en œuvre de l'amélioration avec le S3m

Comme nous l'avons déjà présenté, le modèle S3m doit faire partie du programme interne d'amélioration continue du processus de la société et s'arrimer avec les autres initiatives d'amélioration.

12 Titre de l'ouvrage

Un programme d'amélioration continue devrait normalement comprendre des évaluations périodiques (tous les 12 à 24 mois) menées à l'échelle de l'organisation. Les résultats des évaluations périodiques servent à alimenter un programme d'amélioration continue de la qualité, ainsi que des autres activités permanentes ou stratégiques de l'organisation. Les pratiques du S3m sont génériques et ne se limitent pas à des technologies, à des structures organisationnelles spécifiques ou d'implantations, ou à des cultures de sociétés spécifiques. Cette approche est intentionnelle, car chaque organisation doit décider comment elle doit réaliser les buts visés par les objectifs génériques et spécifiques du domaine des processus de cette pratique.

Le S3m classe les différentes suggestions du modèle selon trois catégories : a) exigée ; b) attendue ; c) informative. Ce classement permet de conserver la cohérence avec le modèle CMMi :

Exigée : Les buts de chaque domaine font partie de la catégorie exigée et doivent être présents, planifiés et exécutés par les processus de la maintenance du logiciel.

Attendue : Les pratiques détaillées font partie de la catégorie attendue. C'est-à-dire qu'elles représentent les implantations de processus qu'on s'attend à trouver en place dans une organisation de la maintenance. Ces pratiques telles que décrites dans le modèle S3m, ou des alternatives acceptables de ces pratiques, doivent être présentes, planifiées et exécutées par les processus de l'organisation pour considérer que les buts sont atteints.

Informative : D'autres informations sont présentées, au niveau des pratiques spécifiques, pour aider le lecteur à mieux les comprendre. Ces informations décrites sous forme d'instructions, de gabarits de produits de travail, d'exemples et de références permettent de mieux interpréter les pratiques du modèle.

4.3. Aperçu du modèle S3m

Tout modèle est une représentation simplifiée de la réalité. Le modèle S3m a l'avantage de référer à plusieurs autres référentiels (ITIL, CMMi, CM3, Cobit [ISA 07], ISO9000 et bien d'autres) et ainsi s'assure de la cohérence de ses recommandations.

Ce modèle propose des pratiques exemplaires ainsi que des itinéraires d'amélioration des processus de la maintenance. Comme nous l'avons dit précédemment, il n'offre pas de techniques, de cycles de vie ou de technologies spécifiques. Les décisions de sélection et d'implantation technologique sont spécifiques de chaque organisation. Il faut donc implanter le modèle générique

d'évaluation dans le contexte d'utilisation de technologies sélectionnées spécifiquement dans le contexte propre à l'organisation. Pour ce faire, on doit utiliser son jugement professionnel pour évaluer comment le contexte spécifique se compare au modèle générique d'évaluation, c'est-à-dire que les domaines, les itinéraires, les facettes et les bonnes pratiques du modèle devraient être couvertes par les processus de maintenance du logiciel de votre entreprise. Ces pratiques sont attendues. Elles ne doivent pas être omises, mais doivent être interprétées pour la société l'utilise selon son environnement spécifique, ainsi qu'en fonction des contraintes et des objectifs d'affaires. La figure 1.3 décrit le diagramme de contexte d'une unité organisationnelle typique de la maintenance du logiciel.

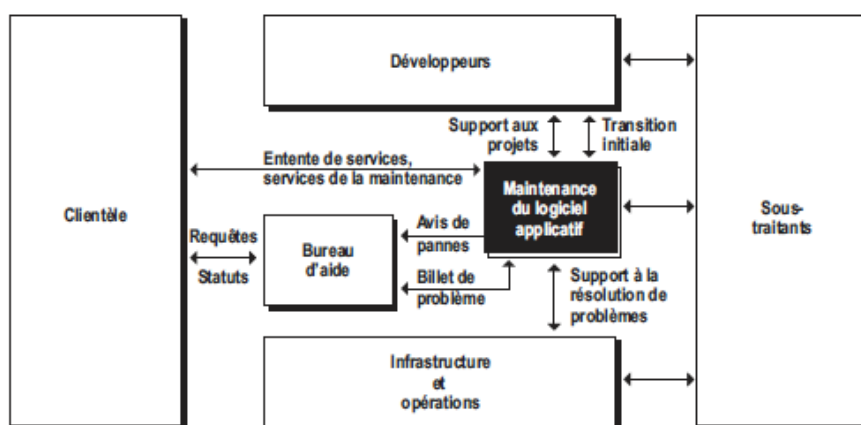


Figure 1.3. Diagramme de contexte du S3m

4.3.1. Les interfaces organisationnelles avec la maintenance

L'unité organisationnelle de la maintenance doit interagir avec plusieurs intervenants. « Le gestionnaire de la maintenance doit s'assurer que les applications opérationnelles opèrent en douceur. Il doit réagir rapidement aux pannes et s'assurer que le service est restauré le plus rapidement possible. Il doit aussi observer un niveau de service établi et conserver la confiance de la clientèle quant à la compétence de son équipe à fournir un service de qualité dans les limites du budget attribué [Apr 08]. »

Dans la figure 1.3, la première et la plus importante interface est celle avec la clientèle. Cette interface se fait bien souvent au quotidien par plusieurs communications et services rapides concernant la résolution de problèmes opérationnels (RP), le support opérationnel et la gestion de requêtes de

modifications (RM) aux logiciels applicatifs en production. C'est donc directement, ou par l'intermédiaire d'un bureau d'aide, que les requêtes (billets) de problèmes parviennent au groupe de la maintenance. Des activités, moins fréquentes, consistent aussi en des négociations et des ententes concernant les priorités des requêtes de modifications qui influencent l'entente de services, la planification, les budgets et la satisfaction de la clientèle.

Une autre interface quotidienne de la maintenance est celle avec le groupe des infrastructures/opérations [ITIL]. Dans cette interface, on retrouve les échanges quotidiens dans le cadre du processus plus global de la résolution de problèmes des logiciels. Le billet qui identifie une requête de la clientèle circulera du bureau d'aide au groupe de la maintenance du logiciel, puis au groupe d'infrastructures/opérations, si nécessaire, afin d'isoler un problème et de le diriger à l'unité organisationnelle qui peut le solutionner [Apr 06]. Une autre activité, moins fréquente, consiste à coordonner les reprises de services, par exemple à la suite d'une panne importante ou d'un désastre où l'on doit aider à restaurer l'accès aux logiciels et aux données de l'organisation, selon les termes de l'entente de services.

Toujours dans la figure 1.3, une autre interface est celle entre l'unité organisationnelle de développement du logiciel (les développeurs) et la maintenance du logiciel (les mainteneurs). C'est à cette étape que le logiciel devient opérationnel et qu'il doit satisfaire les besoins quotidiens de la clientèle. Plusieurs problèmes ont été identifiés à cette interface, et on se doit de bien contrôler la transition des logiciels afin de diminuer le nombre de problèmes, sinon tenter d'en réduire l'impact. Il ne faut pas oublier, non plus, les activités de support aux développeurs lors de ces projets qui sont fournies par des mainteneurs expérimentés. Ces activités nécessitent l'expertise acquise sur les logiciels existants, par exemple : a) développer des stratégies de transitions pour remplacer des logiciels existants ; b) concevoir des interfaces temporaires ou nouvelles pour les développements en cours ; c) vérifier des règles d'affaires ou comprendre les données des logiciels existants ; d) aider dans la migration de données au nouveau logiciel.

Enfin, l'utilisation de sous-traitants est une autre pratique de l'industrie et se manifeste selon toutes sortes d'interfaces avec : a) des sous-traitants qui développent ou configurent des logiciels ou des progiciels pour l'organisation et qui effectueront la transition vers l'organisation de la maintenance ; b) des sous-traitants faisant partie de l'équipe de maintenance au quotidien ; c) des sous-traitants qui effectuent des activités de maintenance d'un progiciel existant selon un contrat de services spécifique ; d) des ententes d'impartition qui remplacent complètement ou partiellement les unités organisationnelles de la maintenance. Le groupe de la maintenance devra, ici, en arriver à négocier et à bien comprendre les ententes contractuelles, et gérer les contrats, les relations et les services rendus.

4.3.2. Les classes de processus de la maintenance

La figure 1.4 regroupe les processus de la maintenance en trois classes : a) les processus opérationnels de la maintenance ; b) les processus de support aux processus opérationnels et c) les processus organisationnels qui sont offerts à toute la société par d'autres unités organisationnelles de l'organisation (par exemple : finances, ressources humaines, achats, etc.).

Ce modèle générique de processus présenté à la figure 1.4 a pour objectif d'aider le lecteur, pendant l'utilisation du modèle S3m, à avoir un point de référence pour interpréter et mieux comprendre le point de vue des différentes bonnes pratiques du modèle. Le modèle générique de la figure 1.4 établit un cadre conceptuel et une terminologie qui servira, lors de la mise en situation, d'explications dans les exemples de pratiques spécifiques. Trois points de contact externe très opérationnels (transition, gestion des requêtes et résolution de problèmes), relativement plus importants, y sont représentés. Voici une description plus détaillée de la figure 1.4.

4.3.2.1. Les processus opérationnels de la maintenance

Une unité organisationnelle générique de maintenance du logiciel utilise des processus opérationnels (aussi appelés processus primaires) mis en œuvre au moment où un logiciel, qui lui est destiné, débute son cycle d'acquisition ou de démarrage de développement.

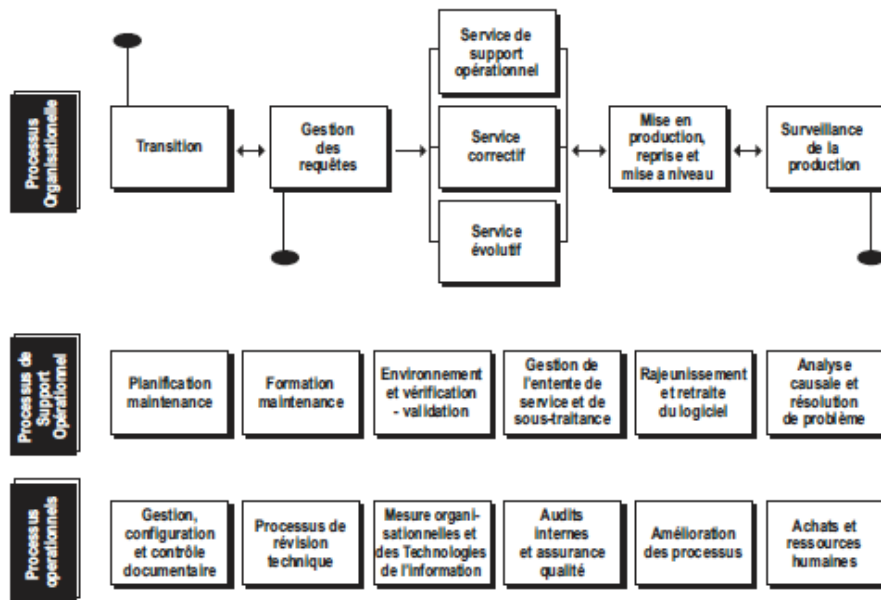


Figure 1.4. Les trois classes de processus de maintenance du logiciel

Le processus de « transition » vise à s'assurer du transfert, d'une manière structurée et contrôlée, du logiciel développé/acheté vers le mainteneur. Ce processus fait appel au processus de support de la gestion des ententes de services qui permet de préciser la nature et le volume des services de la maintenance qui seront offerts sur un logiciel.

Une fois le logiciel sous la responsabilité de la maintenance et encadré par une entente de services, le processus 'gestion des requêtes' gère les requêtes provenant de toutes sources (par exemple : a) de la clientèle ; b) du bureau d'aide ; c) des développeurs ; d) du groupe infrastructures/opérations). Ce sont ces requêtes de services quotidiennes qui doivent être gérées efficacement. La première étape est de décider si on les traite ou si on les achemine vers d'autres unités organisationnelles (en fonction de leur nature et de leur taille voir figure 1.1). Les requêtes qui sont acceptées sont identifiées, placées en ordre de priorité, assignées et traitées selon les catégories de services opérationnels :

- support opérationnel ne nécessitant pas de modification du logiciel ;
- correction du logiciel ;

- évolution du logiciel.

Il est à noter que certaines requêtes ne nécessitent pas de modification du logiciel. Dans le modèle S3m, elle sont identifiées sous la catégorie 'support opérationnel', car ces requêtes consistent à : a) répondre à des questions ; b) fournir des informations et des conseils ; c) permettre à la clientèle de mieux comprendre le logiciel, une transaction ou la documentation.

Les requêtes nécessitant des modifications seront analysées, autorisées, effectuées, vérifiées et « mises en production » (c'est-à-dire mises en service). Finalement, elles seront surveillées pour s'assurer que l'environnement opérationnel préserve ses qualités opérationnelles à la suite d'un changement. La surveillance consiste principalement à s'assurer du niveau de la performance et de la capacité opérationnelle des logiciels et à s'assurer d'un service continu.

Enfin, les processus opérationnels comprennent toujours une « surveillance de la production » pour observer et analyser le comportement des logiciels et avertir les autres intervenants (opérateurs, programmeurs de systèmes, réseaux, etc.) de la nécessité d'enquêter sur une dégradation du service.

4.3.2.2. *Les processus de support opérationnels de la maintenance*

Un processus qui est utilisé, lorsque qu'il est requis par un processus opérationnel de la maintenance du logiciel, est classé 'processus de support opérationnel'. Dans cette classe, on inclut les processus suivants :

- la planification de la maintenance, à trois niveaux : 1. stratégique et de transition ; 2. versions et de mises à niveau ; et 3. des requêtes de modifications ;
- de formation des ressources de la maintenance et des intervenants qui utilisent ou doivent comprendre les processus et les services de la maintenance ;
- des environnements pour les modifications et les essais, ainsi que pour la vérification et la validation du travail et des logiciels ;
- de gestion de l'entente de services (*de l'anglais Service Level Agreement*) et d'autres contrats ;
- de rajeunissement ou de retraite d'un logiciel ;
- d'analyse causale et de résolution de problèmes.

Ce sont tous des processus qui ne sont pas gérés par des billets et qui viendront appuyer certaines activités des processus opérationnels à un moment choisi.

4.3.2.3. *Les processus organisationnels de la maintenance*

Les processus organisationnels sont souvent offerts par la division des Technologies de l'information et par d'autres divisions de l'organisation (par exemple : ressources humaines, finances, assurance qualité et ISO 9001). La maintenance doit aussi s'intégrer dans les activités d'amélioration de l'organisation et des Technologies de l'information. La maintenance doit suivre, par exemple, les processus :

- de gestion de la configuration et de gestion documentaire des logiciels ;
- de révisions techniques ;
- de mesure organisationnelle ;
- d'audits internes et d'assurance qualité indépendante ;
- d'amélioration des processus ;
- d'achats et de ressources humaines.

Ce modèle générique devrait aider à faire la différence entre les trois classes de processus impliqués dans la maintenance du logiciel. L'important, au tout début d'un processus d'amélioration, c'est l'identification des processus plus opérationnels et ceux qui vont supporter l'unité organisationnelle. La liste présentée ici ne représente qu'un point de vue (celui d'organisations de maintenance spécifique qui ont participé avec nous aux essais du modèle), et il faut créer son propre diagramme en élaborant une classification qui utilise la terminologie et les processus de votre organisation. Ce diagramme de haut niveau est important pour présenter un sommaire des grands processus de la maintenance du logiciel de l'unité organisationnelle. En plus, il pourrait être utile pour une certification ISO 9001.

Le S3m a été conçu pour inclure :

- les processus de la figure 1.4 ainsi que les interfaces organisationnels (figure 1.3) ;
- des logiciels développés et maintenus à l'interne du ministère, de l'organisation, de la société ou de l'entreprise (on possède le code source) ;
- les logiciels maintenus à l'interne et qui ont été acquis d'un fournisseur (on ne possède pas tout le code source, et il faut travailler quotidiennement avec le fournisseur) ;
- l'évaluation des processus utilisés par le fournisseur d'impartition de la maintenance.

4.3. Étude de cas d'amélioration des processus avec le S3m

Cette étude de cas présente un exemple pratique de l'application du modèle S3m dans une multinationale de fabrication de moteurs d'avions située à Montréal au Canada. Comme bien d'autres sociétés l'utilisation de progiciels achetés est en croissance et le développement interne diminue constamment. L'entreprise effectue donc de plus en plus de maintenance de logiciel. Ce groupe logiciel est certifié ISO 9001 depuis plusieurs années. Un des logiciels de cette société sert à la facturation. Suite à des plaintes de clients le groupe d'audit interne a effectué une vérification des factures de ce système et a identifié un certain nombre de problèmes. Certains problèmes, non rapportés, causaient une sous facturation et certains autres problèmes causaient une sur facturation des clients. Un certain nombre de recommandations ont mené à la décision d'effectuer une évaluation de la maturité des processus de la maintenance de cette équipe de maintenance.

L'équipe de maintenance a donc procédé à une évaluation S3m en utilisant le questionnaire Excel des pratiques exemplaires pour les niveaux de maturité zéro, un et deux. Les pratiques exemplaires de niveau zéro sont évaluées d'une manière binaire (i.e. Oui/Non). Une réponse 'Oui' obtient la note de zéro. Une réponse 'Non' obtiens 100 points. Par la suite, les pratiques des niveaux supérieurs (un et deux) sont évaluées avec plus de nuance. L'évaluateur interne utilise une grille d'évaluation qui assigne les cotes suivantes (selon son observation et un consensus obtenu avec l'équipe de maintenance évaluée):

A: Processus absent, note de 0%: Il n'y a aucune évidence que la pratique recommandée existe ou soit mise en œuvre.

P: Processus partiellement exercé, note de $(50\% - 16\%) / 2 + 16\% = 33\%$: Quelques objectifs et buts de la pratique recommandée sont atteints.

G: Processus grandement déployé et fonctionnel: note de $(85\% - 51\%) / 2 + 51\% = 68\%$: Une large portion des objectifs et des buts de la pratique recommandée sont atteints.

C: Processus complètement déployé et fonctionnel: note de $(100\% - 86\%) / 2 + 86\% = 93\%$: Les objectifs et les buts de la pratique recommandée sont entièrement atteints.

4.3.2. Les résultats de l'évaluation S3m

Cette section décrit les résultats obtenus par l'équipe de maintenance du logiciel de facturation. Les résultats de l'évaluation du niveau zéro n'est seulement que de

37%. En observant les réponses on peut voir (cases en vert dans le tableau 1.1) se dessiner graduellement la situation actuelle de la maintenance pour le logiciel de facturation. Les mainteneurs n'ont pas de processus corporatif, ni de formation. Ils tentent de faire de l'innovation, localement, sans aide externe et effectuent le support, l'évolution et les tests du logiciel au mieux de leurs connaissances et aptitudes personnelles. Le logiciel est sous gestion de configuration.

Domaine de Processus	Secteurs clés de la maintenance	Pratiques S3m	Cote assignée	Note (%)
Management des processus	Focalisation sur les processus	1.0.1	Oui	0%
	Définition des processus/services	2.0.1	Oui	0%
	Formation	3.0.1	Oui	0%
	Performance des processus	4.0.1	Oui	0%
	Innovation et Déploiement	5.0.1	Oui	0%
		5.0.2	Non	100%
		5.0.3	Non	100%
Total				29%
Management des requêtes	Management des requêtes de services et événements	1.0.1	Oui	0%
	Planification de la maintenance	2.0.1	Oui	0%
	Suivi et supervision des requêtes	3.0.1	Oui	0%
	Management des ententes de services et de la sous-traitance	4.0.1	Oui	0%
Total				0%
Évolution du logiciel	Coordination avant livraison et transition	1.0.1	Non	100%
	Services de support opérationnel	2.0.1	Non	100%
	Services d'évolution et de correction	3.0.1	Non	100%
	Vérification et validation	4.0.1	Non	100%
Total				100%
Support à l'évolution du logiciel	Management de versions et de configuration	1.0.1	Non	100%
	Assurance Qualité (services, processus et produits)	2.0.1	Oui	0%
	Analyse et mesure des services/logiciels	3.0.1	Oui	0%
	Analyse causale et résolution de problème	4.0.1	Oui	0%
	Rajeunissement, retraite et migration	5.0.1	Oui	0%
Total				20%

Note Finale:	37%
---------------------	-----

Tableau 1.1. *Sommaire des résultats de l'évaluation du niveau de maturité 0*

L'évaluateur a voulu en savoir plus et il a évalué les pratiques recommandées par le S3m pour le niveau de maturité 1 (voir les résultats au tableau 1.2). L'approche d'évaluation est la suivante. Si une pratique de niveau zéro est rencontrée alors on devrait pousser plus loin l'évaluation de cette pratique afin de voir si elle va rencontrer les exigences du niveau supérieur. Au niveau de maturité un, le modèle S3m décrit que ce sont typiquement les individus qui décident des processus de maintenance utilisés. Il y a donc autant de processus que de mainteneurs. Cette situation décrit bien à la société évaluée. L'évaluation de la pratique Pro1.1.1 tente de déterminer à quel point l'amélioration des processus maintenance est informelle. L'évaluateur a assigné la cote 'G' car il y a une initiative d'évaluation en ce moment. La pratique Pro1.1.2 tente d'identifier les améliorations concernant les aspects techniques du travail de maintenance. Dans cette société, c'est un aspect majeur d'investissement et la cote 'C' a été assignée.

Les services de maintenance et de support sont aussi fondé seulement sur l'expertise des individus. Quelques documents et dessins personnels existent d'une manière ad hoc et sont utilisés par chaque mainteneur. Bien que ces initiatives soient personnelles l'évaluateur a évalué les pratiques suivantes avec une cote 'G' car elle reflètent assez bien la situation vécue : Pro2.1.1 : Est-ce que les processus et services, dans l'unité de la maintenance, sont informels et seulement basés sur l'expérience des individus ?, Pro2.1.2 : Retrouvez vous des initiatives individuelles de documentation des processus et des services de la maintenance visent principalement des aspects techniques particuliers à certains logiciels ou décrivent, dans un format local, les activités d'une unité organisationnelle spécifique de la maintenance du logiciel ?, Pro5.1.2 Est-ce que les initiatives individuelles d'amélioration et d'innovations, visant principalement les aspects techniques internes delà maintenance du logiciel, sont accomplis ? et finalement Pro5.1.3 Est-ce que les évaluations de nouvelles technologies, méthodologies et nouveaux outils de la maintenance sont réalisées informellement ?

Domaine de processus	Secteurs clé de la maintenance	Pratique S3m	Cote assignée	Note (%)
Management des processus	Focalisation sur les processus	1.1.1	G	68%
		1.1.2	C	93%
	Définition des processus/services	2.1.1	G	68%
		2.1.2	G	68%
	Innovation et déploiement	5.1.2	G	68%
		5.1.3	G	68%
Total				36%
Évolution du logiciel	Coordination avant livraison et transition	1.1.1	C	93%
	Services de support opérationnel	2.1.1	C	93%
	Services d'évolution et de correction	3.1.1	C	93%
	Vérification et validation	4.1.1	C	93%
Total				93%
Support à l'évolution du logiciel	Management des versions et de la configuration	1.1.1	C	93%
Total				15,5 %
Note Finale:				36%

Tableau 1.2. Sommaire des résultats de l'évaluation du niveau de maturité 1

Le domaine de processus suivant, l'évolution du logiciel, est de nouveau jugé assez bien. Tous les secteurs clés ont été cotés 'complètement atteint' ce qui donne une note de 93%. Dans cette société, le développeur est aussi le responsable de la maintenance: la transition n'est donc pas un problème. L'activité de support est bien structurée, et est effectuée par la personne qui a développé le logiciel: en cas de panne, les utilisateurs peuvent le contacter facilement (il est situé près des utilisateurs du logiciel). Tous les services de maintenance sont exécutés d'une manière ad hoc. La vérification et validation de modifications sont faite de manière informelle et les mises en production sont faites selon la disponibilité du mainteneur.

Une seule pratique du domaine de support à l'évaluation du logiciel avait passé le niveau zéro alors elle est évaluée au niveau de maturité supérieur. Encore une fois, le management des versions et de la configuration est réalisé par une seule ressource. Cette personne s'assure de la validation et du contrôle des versions et de la configuration. Toutes les autres pratiques du niveau un sont cotés 'A' «absentes».

La note globale de ce niveau de maturité est de 36%, c'est-à-dire 'partiellement atteint'. Nous pouvons voir ici que le modèle S3m suggère que cette organisation n'a pas encore complètement atteint le niveau un de la maturité des processus de la maintenance. Elle est donc encore immature étant donné que la plupart des sociétés visent un niveau minimum de trois.

4.4. Conclusion

Ce chapitre a présenté un référentiel de pratiques exemplaires développé spécifiquement pour l'amélioration des processus d'évolution et de maintenance du logiciel. Le S3m propose des pratiques exemplaires qui permettent aux sociétés d'évaluer la maturité de leur processus d'évolution et de maintenance du logiciel. L'exemple de son utilisation, dans une grande entreprise de fabrication de moteurs d'avion Montréalaise, a permis l'identification du niveau de maturité des processus de l'équipe de maintenance d'un logiciel de facturation. Cette évaluation a permis de soulever des problématiques et des solutions possibles pour l'équipe de mainteneurs.

L'équipe de management est désormais au courant des résultats de l'évaluation avec le modèle S3m, qui a fourni la preuve que les processus de maintenance se situent au entre le niveau zéro et un sur l'échelle de maturité. Ce constat permettra d'initier des actions d'améliorations pour le futur. L'objectif du démarrage de l'amélioration des processus est atteint et pourra se poursuivre en utilisant les recommandations du modèle. Des évaluations subséquentes, à chaque année, permettront de s'assurer que la maturité progresse et que des objectifs spécifiques soient établis et rencontrés.

Bibliographie

24 Titre de l'ouvrage

- [Apr 01] APRIL A., BOUMAN J., ABRAN A., Al-Shurougi., *Software maintenance in a service level agreement : controlling the Customer expectations*, Fourth European Software Conference FESMA, Heidelberg, Germany, May 2001.
- [Apr 06] APRIL A., ABRAN A., *Améliorer la maintenance du logiciel*, Loze-Dion éditeur, Longueuil, 2006, www.s3m.ca
- [Apr 08] APRIL A., ABRAN, A., *Software maintenance management : évaluation and continuous improvement*, IEEE Wiley interscience, 2008, www.s3m.ca
- [Bas 04] BASQUE R., *Un itinéraire fléché vers la Capability Maturity Model Integration*, Dunod, 2004.
- [Ben 00] BENNETT K. H., *Software maintenance : a tutorial*, Software Engineering, Dorfman and Thayer editors, IEEE Computer Society Press, 2000.
- [Bou 99] BOUMAN J., TRIENEKENS J., VAN DER ZWAN M., *Spécifications of service level agreement : clarifying concepts on the basis of practical research*, Proceedings of Software Technology and Engineering Parctice, 1999, (2^e édition).
- [Dor 02] DORFMAN M., THAYER R. Y., *Software engineering : the supporting processes*, Dorfman and Thayer editors, IEEE Computer Society Press, 2002.
- [ISA 07] IT Governance Institute (ISACA), *Control and Audit for Information and Related Technology (CobIT)*, version 4.1, 2007, www.isaca.org
- [ISB 04] International Software Benchmarkin Standars Group, *Data collection questionnaire : application software maintenance and support*, version 2.3, 2004, www.isbsg.org
- [ITI 09] ITIL LAGRANGE X., GODLEWSKI P., TABBANE S., *Réseaux GSM-DCS*, Hermès, 1999 (4^e édition). [Style : Références]
- [ISO 05] International Organization for Standardization, *ISO 20000-1 :2005, Information technology -- Service management -- Part 1: Specification*, 2005.
- [ISO 06] International Organization for Standardization, *ISO/IEC 14764 :2006, Software Engineering -- Software Life Cycle Processes -- Maintenance*, 2006.
- [ISO 08] International Organization for Standardization, *ISO/IEC TR 15504-6 :2008, Information technology -- Process assessment -- Part 6: An exemplar system life cycle process assessment model*, 2008.
- [Kaj 01] KAJKO-MATTSSON M., *Corrective maintenance maturity model, PhD thesis report 01-015*, Stockholm University, 2001. (4^e édition). [Style : Références]
- [Kha 05] KHAN K., ZHANG Y., *Managing Corporate Information System Evolution and Maintenance*, Idea Group, 2005.
- [She 97] SHEARD S., *The frameworks Quagmire Software Productivity Consortium*, 1997.
- [SWE 06] ABRAN A., MOORE J. W., BOURQUE P. Dupuis R., *Guide to the software engineering body of knowledge*, IEEE Computer Society Press, 2005, www.computer.org/portal/web/swebok