



École de technologie supérieure
L'ÉTS est une constituante du réseau de l'Université du Québec

Projet individuel de session – MGL804

Travail de session n°18

Hiver 2011

Introduction

Ce rapport a pour but de répondre à la problématique « Faites une revue bibliographique concernant les modèles de maturité et mettez à jour l'inventaire des modèles. » dans le cadre du travail personnel de session pour le cours Maintenance de logiciels.

Depuis le milieu des années 80, plusieurs propositions de modèles de maturité ont été proposées, tant par l'industrie que par la recherche en génie logiciel. Les modèles de maturité ont connu un grand essor, surtout depuis la publication du CMM en 1993, et le concept est devenu de plus en plus populaire. Il s'est donc étendu à tous types d'industries, allant de la gestion financière à la gestion des Ressources Humaines.

Afin de répondre à la problématique, j'ai réalisé une recherche non-exhaustive des modèles de maturité liés à la maintenance de logiciels, parus depuis 2003, l'inventaire d'April-Abran s'arrêtant à cette date. Non-exhaustive, car il est facile de s'y noyer très rapidement.

Dans un premier temps, je ferai une remise en situation afin de situer la problématique dans son contexte. Puis je présenterai le nouvel inventaire réalisé, et enfin j'évaluerai les modèles selon des critères d'utilisabilité.

Table des matières

I. Remise en situation	4
I.1 Bref rappel de définitions	4
I.2 Les modèles de maturité : références ?	4
I.3 Remise en situation : Inventaire des modèles de maturité du génie logiciel (1991-2003)	5
II. Inventaire des modèles de maturité	5
III. Évaluation des modèles selon des critères d'utilisabilité	8
III.1 Présentation des critères	8
III.2 Évaluation des modèles	9
Conclusion.....	11
Bibliographie	12

I. Remise en situation

Cette partie a pour objectif de remettre en situation le lecteur afin qu'il puisse comprendre quel a été le point de départ de mes recherches pour la réalisation de ce nouvel inventaire de modèles de maturité en génie logiciel.

I.1 Bref rappel de définitions

Dans un premier temps, il est important de bien redéfinir ce qu'est la maturité, pour ensuite donner et comprendre la définition d'un modèle de maturité. La maturité est « l'étendue à laquelle un processus spécifique est défini explicitement, géré, mesuré, contrôlé, et efficace ». [Pau93] Au travers de cette définition nous apparaît un aspect important qui est la notion de planification, d'organisation d'un processus. Elle se retrouve pleinement dans la définition d'un modèle de référence pour l'évolution de la maturité des processus du logiciel, soit « une représentation des attributs clés, d'entités organisationnelles choisies, qui ont un effet sur la progression de ces unités organisationnelles vers la pleine réalisation de leur potentiel et de leur développement. » [Gar93] Plus explicitement, un modèle de maturité a pour objectif de fournir, dans une industrie spécifique, des niveaux de maturité de références ainsi que des pratiques pour développer de façon efficace des processus dans le but d'une réalisation maximale du potentiel de cette industrie. Dans notre situation, l'industrie concernée est la maintenance de logiciels.

I.2 Les modèles de maturité : références ?

Le concept de modèles de maturité est de plus en plus populaire, il y a donc un nombre croissant de nouveaux modèles et ce dans des industries de plus en plus diverses. Il est alors nécessaire pour la réalisation de l'inventaire de savoir lesquels peuvent ou non servir de références pour la maintenance de logiciels, afin de ne pas se noyer dans une multitude de modèles sans importance.

Les affirmations suivantes nous permettent d'identifier les modèles comme pouvant servir de référence :

- les pratiques détaillées de ce modèle sont documentées et accessibles ;
- le modèle possède des pratiques qui s'adressent indirectement ou directement à la maintenance ;
- le modèle et son utilisation sont récents ;
- le modèle est déployé et connu ;
- le modèle est spécifique d'une industrie.

Tous les items de cette énumération ont à chacun un réel impact dans le choix des modèles. Il semble tous de grande évidence, mais nous verrons par la suite qu'il est finalement difficile de trouver un modèle de maturité validant toutes ces affirmations.

I.3 Remise en situation : Inventaire des modèles de maturité du génie logiciel (1991-2003)

Le point de départ de ma recherche est l'inventaire des modèles de maturité du génie logiciel publié dans « Améliorer la maintenance logiciel » d'April-Abran. Cet inventaire liste les modèles parus entre 1991 et 2003. C'est donc à cette date que débutera le nouvel inventaire. Il est à noter que ces modèles ont autant été développés par l'industrie que par des chercheurs.

Il est important de souligner qu'une trentaine de propositions y concernent le génie logiciel ; seulement douze touchent une perspective de la maintenance ; seulement trois valident les critères de sélection pour être utilisable pour la conception d'un modèle de maturité spécifique de la maintenance de logiciel [Cam94-Tri96 ; Nie02 ; COB00] ; et seulement deux visent directement la maintenance de logiciel. [Zit96 ; Kaj01d]

Précédemment, nous avons indiqué qu'il semblait difficile pour un modèle de valider les diverses affirmations pour pouvoir servir de références, le faible nombre de modèles visant directement la maintenance en est une confirmation directe.

II. Inventaire des modèles de maturité

Mon travail de session a donc pour objectif de faire un inventaire des modèles de maturité depuis 2003. Les nouveaux modèles sont nombreux, certains sont même déjà tombés dans la désuétude. J'ai donc décidé de n'en sélectionner que quelques uns qui me paraissaient encore valables et surtout pouvant être reliés au monde de la maintenance de logiciel.

Le tableau suivant présente l'inventaire non-exhaustif des modèles de maturité depuis 2003 :

Dates	Modèles de Maturité	Description sommaire
2004	Information Process Maturity Model (IPMM)	Décrit les pratiques pour une organisation de développement d'information au niveau software notamment.
2004 (2006)	PRTM Customer Requirement Maturity Model	Mettre en place le traitement avec le client: retours, demandes...
2005	Configuration Management Maturity Model	Mappage entre les différents composants du software.
2007	Requirement Maturity Model (RMM)	Définir les exigences.

2008	Change proficiency Maturity Model	Mettre en place, guider une stratégie Agile.
2009	Continuous integration Maturity Model (CIMM)	Intégration en continue du logiciel.
2010	Inspection Capability Maturity Model (ICMM)	Le but de l'inspection est d'identifier les défaillances.
-	Documentation Maturity Model (DMM)	Archivage, documentation sur logiciel.
-	Agile Maturity Model (AMM)	Organisation à mettre en place pour être en progression continue.

Nous allons décrire plus en profondeur les modèles énumérés ci-dessus.

JoAnn Ackos [1] présente dans sa publication, en 2004, une update de l'IPMM : modèle qui décrit les pratiques qui rendent une organisation de développement d'information effective. En réalité ce modèle existe depuis une douzaine d'années. Les niveaux, ainsi que les pratiques y sont clairement définis. Les cinq niveaux proposent à la fois d'assister l'organisation courante, mais aussi de voir plus loin notamment au niveau de l'amélioration de processus. Huit caractéristiques y sont aussi décrites, il est possible de les apparenter à des pratiques ; leurs rôles étant d'évaluer le degré d'organisation d'une industrie. Au final cela revient à dire si tel ou tel critère a été établi, et si non, qu'il serait bien de le faire, d'où l'affiliation à des pratiques. De plus, il fait une approche intéressante de ce modèle en le présentant comme un besoin pour le monde du développement software. En effet le SEI s'étant développé en même temps, mais ne présentant aucun signe de pratiques quant à l'organisation de l'information, il voit dans ce modèle une aide pour ce domaine. Nous pouvons directement en faire le lien avec la maintenance, qui a vraiment besoin de toutes documentations possibles sur un logiciel. Il est apparent que ce modèle est encore utilisé et ne tend qu'à se développer.

Christina Hepner de chez PRTM décrit le *Customer Requirement Maturity Model* dans sa publication [2], plus exactement quelles sont leurs actions dans la réalisation de ce modèle. L'objectif du CRMM est de développer une bonne stratégie quant au traitement du client et d'identifier à quel niveau de maturité se trouve une entreprise en particulier. Pour cela deux types d'exigences sont mis en avant : les exigences d'identification (collecter et interpréter les besoins du client) et celles de management (comment les besoins sont gérés pendant le processus de développement du produit). Les pratiques sont énoncées de façons plus ou moins claires, mais sont présentes. Je doute tout de même que ce modèle soit incontournable, mais il est certain qu'il se lie à la maintenance du fait de la relation mainteneur-client qui est indispensable. Notons tout de même une publication plus récente en 2006.

Le *Configuration Management Maturity Model* est décrit au travers du site ERP [3], dédié pour les entreprises IT Automotion afin de promouvoir leurs sujets émergents. La description du CMMM est sous forme d'un post et l'auteur spécifie que le modèle n'est pas strict, mais que ce sont quelques pensées qui y sont décrites. Cependant les cinq niveaux y sont décrits, notamment pour le logiciel. C'est pour cela qu'il m'a semblé pertinent de l'intégrer à l'inventaire. Il présente certaines relations ayant lieu entre différents composants d'un logiciel, ce qui est un atout pour la maintenance afin de pouvoir se retrouver plus facilement dans l'architecture d'un logiciel. Les pratiques ne sont pas apparentes cependant.

Le RMM a pour objectif de décrire les étapes et les éléments pour le management des exigences de processus. La firme Tyner blain décrit dans son site [4] les cinq différents niveaux du RMM, notons que la description y est précise et pourrait donc faire office d'énumération des pratiques pour ce modèle. À dire vrai, le RMM a déjà été développé par IBM en 2003, mais le rôle de Tyner Blain est de faire un mappage entre le CMMi et le RMM. Leur conclusion est que la validation du niveau 5 du RMM reviendrait à valider le niveau 3 du CMMi. Le lien avec le monde du logiciel est ici assez évident. Pour celui avec la maintenance, nous pouvons dire qu'il est important de respecter des exigences lors d'un processus, et avoir un modèle permettant d'en organiser le respect est intéressant.

Paradigm Shift International présente le *Change Proficiency Maturity Model* [5] comme un outil pour prioriser et guider une transformation de type Agile ou une stratégie d'amélioration. La structure est basée sur la progression au travers de cinq niveaux de connaissances et de stratégies appuyés par des pratiques et des procédures. Les entreprises ou les départements au sein des entreprises tendent de plus en plus vers des modèles Agile. J'ai trouvé ce modèle pertinent du fait qu'il propose en quelque sorte une méthode d'implantation du modèle Agile, ce qui me semble un plus pour le domaine de la maintenance.

Le CIMM [6] est une création suite à une session du CITCON (Continuous Integration and Testing Conference) en 2009. La notion d'intégration continue se développe de plus en plus. Pour le monde du logiciel, cela permet d'intégrer et tester le code au fur et à mesure et non de réaliser un travail de deux mois et se rendre compte des erreurs au moment de l'implantation. L'intérêt pour la maintenance y est évident : la répartition du travail et ainsi de l'efficacité et donc de la satisfaction pour le client. Les niveaux et les pratiques de ce modèle ne sont pas accessibles à tous publics, cependant le modèle semble être récent et à l'origine d'intérêt.

L'objectif de l'inspection est d'identifier les défaillances, cette méthode est largement reconnue dans le monde de l'ingénierie logicielle. Le rapport de l'ICMM avec la maintenance est immédiatement reconnaissable. Plus précisément, il produit un support pour l'atteinte et l'amélioration de pratiques d'inspection logicielle dans une organisation logicielle. Huit cas d'organisations y sont présentés, ainsi que sa structure qui est explicitement décrites. Malheureusement ce modèle n'est pas non plus accessible à tous publics.

Les deux derniers modèles que nous allons présenter sont postérieurs à 2003, mais ne sont pas datés car ils ne font pas l'objet d'une date de publication précise.

Le DMM est spécifiquement axé sur comment atteindre la qualité de documentation utilisée dans la compréhension de programmes. Les ingénieurs logiciels, ainsi que les techniciens produisent de la documentation durant tous types de cycle de vie de développement. L'article « Towards a documentation Maturity Model » [8] présente le travail réalisé dans cet optique de modèle de maturité. Pour la maintenance, il est indispensable d'avoir de la documentation. Nous l'avons mentionné plus tôt, mais il me semble important d'insister sur ce point. Une bonne documentation, bien organisée, est indispensable. Une fois de plus les pratiques et niveaux de ce modèle ne sont pas accessibles.

L'AMM [9] se concentre principalement sur l'application d'activités de développement. Les dimensions sont : tester, le management de code source, la propriété de code collectif, la collaboration, l'assurance et la gouvernance, la simplicité du design et les processus de construction. Nous remarquons un rapport direct avec le monde logiciel, il est même évident d'en faire le lien avec la maintenance. Les pratiques et les niveaux de l'AMM sont disponibles. Une des grands avantages de ce modèle est la rapidité qui en découle du fait d'une meilleure gestion de tous types de points. Cela peut, pour la maintenance, être à l'origine d'un temps de traitement plus court, ce qui est un plus. Notons aussi que depuis 2008, il existe un lien officiel entre Agile et le CMMi.

III. Évaluation des modèles selon des critères d'utilisabilité

L'objectif de cette partie est d'évaluer les modèles de maturité, dont nous avons réalisé l'inventaire précédemment, selon des critères d'utilisabilité que nous allons énoncer ci-dessous.

III.1 Présentation des critères

Il existe quatre critères qui nous permettent d'évaluer les modèles de maturité. Ils ont été définis lors de la réalisation de l'inventaire par April-Abran. Le tableau suivant les présente :

Critères	OUI	NON
1.	Pratiques détaillées disponibles.	Seulement un cadre général est proposé.

2.	Le modèle est établi dans l'industrie et est incontournable.	Le modèle est expérimental, marginal, ou désuet.
3.	Plusieurs publications récentes démontrent l'intérêt du modèle.	Il y a peu de publications importantes, récentes ou provenant d'autres personnes que l'auteur.
4.	Contribution générique au domaine du génie logiciel.	Spécialisé dans une industrie spécifique.

Tout d'abord notons qu'ils reprennent les conditions nécessaires pour qu'un modèle puisse servir de référence.

De plus, il est indispensable pour l'installation d'un modèle de maturité dans une entreprise de disposer de pratiques détaillées afin de réaliser une « parfaite » implantation. Le critère n°1 est ainsi justifié. Nous considérons par pratiques disponibles, des pratiques accessibles à tous publics. Il est aussi intéressant de noter que dans le cas de non validation du critère n°1, il sera difficile de réaliser un exercice de mapping avec des pratiques détaillées de la maintenance.

Le critère 2 permet de réaliser une différenciation rapide entre des modèles de maturité de quelques pages proposées par des consultants et d'autres qui sont « incontournables ». Par là, je veux dire des modèles qui ont été repris, appliqués et validés par d'autres personnes/industries. C'est donc vraiment sur le sens du mot « incontournable » que je baserai la validation ou non pour ce critère.

Le critère n°3 assure que le modèle ait été développé, implanté ou validé par d'autres entreprises ou chercheurs. En effet, de telles actions sont le plus souvent à l'origine de nouvelles publications. Il est important que celles-ci soient récentes, afin d'éviter que le modèle soit déjà tombé en désuétude.

Enfin, il est évident que le modèle de maturité en question ait apporté une contribution générique au domaine du génie logiciel. Le fait qu'il soit spécialisé dans une industrie spécifique remettra en question sa validité pour la maintenance de logiciel au général.

III.2 Évaluation des modèles

Le tableau suivant présente l'évaluation des modèles de maturité selon les critères d'utilisabilité définis précédemment.

Modèles de Maturité	Critère 1	Critère 2	Critère 3	Critère 4
IPMM	Oui	Oui	Oui	Oui
PRTM Customer Requirement Maturity Model	Oui	Non	Oui	Non
Configuration Management Maturity Model	Non	Non	Non	Oui
RMM	Oui	Non	Oui	Oui
Change proficiency Maturity Model	Oui	Non	Oui	Oui
Continuous integration Maturity Model (CIMM)	Non	Non	Oui	Oui
ICMM	Non	Non	Oui	Oui
DMM	Non	Non	Oui	Oui
AMM	Oui	Non	Oui	Oui

Notons que nous retrouvons la justification de tous les critères dans la description des modèles de l'inventaire de la partie II. Il ne me semble donc pas nécessaire de re-justifier la validation ou non des critères.

Conclusion

Tout au long de ce rapport nous avons réalisé un nouvel inventaire des modèles de maturité en lien avec la maintenance de logiciel et ce depuis 2003. Neufs modèles ont été présentés et évalués, mais il est certain qu'un plus grand nombre de modèles a été publié depuis 2003. Cependant, la recherche fut axée sur des modèles pertinents et surtout toujours d'actualité. En effet, de nombreux modèles sont supplantés par de nouvelles propositions, et ceux qui ne sont pas tenus à jour tombent dans la désuétude. Au final, ceux qui survivent sont les modèles qui deviennent populaires et attirent diverses industries qui réalisent d'importants investissements pour leur mise-à-jour et leur développement.

Cet inventaire répond à sa façon à la problématique énoncée. La réalisation d'un inventaire exhaustif demanderait un travail de plus grande ampleur et pourrait s'inscrire dans la démarche d'une nouvelle publication sur la maintenance en génie logiciel. Ceci n'est malheureusement pas mon domaine de prédilection, bien que j'aie pu agrandir mes connaissances dans ce domaine tout au long de ce projet.

Bibliographie

- [1] http://www.comtech-serv.com/pdfs/IPMM_update_04.pdf
- [2] http://www.prtm.com/uploadedFiles/Strategic_Viewpoint/Articles/Article_Content/PRTM_Are_You_Hearing_Customers.pdf
- [3] http://erp4it.typepad.com/erp4it/2005/10/a_configuration.html
- [4] <http://tynerblain.com/blog/2007/01/25/cmml-and-rmm-intro/>
- [5] <http://www.parshift.com/docs/aermodA2.htm>
- [6] <http://www.anthillpro.com/blogs/anthillpro-blog/2009/05/05/1241542860000.html>
- [7] <http://onlinelibrary.wiley.com/doi/10.1002/smr.456/abstract>
- [8] <http://www.mendeley.com/research/towards-a-documentation-maturity-model/>
- [9] <http://whattodowearelikethatonly.blogspot.com/2008/08/agile-maturity-model.html>
- [Pau93] Paulk, M., B. Curtis, *et al.* *CMM for Software v1.1*, CSMU/SEI-93-TR24, ADA263403, Software Engineering Institute, 1993
- [Gar93] Garcia, S. «Principles of Maturity Modeling », Proceedings of SEI Symposium, Pittsburgh, August 23-26, 1993.
- [Cam94 -Tri96] Tritium/Camelia
- [Nie02] Modèle de maturité des services des Technologies de l'information
- [COB00] Modèle de maturité de gestion, contrôle, et audit des Technologies de l'information
- [Zit96] Zitouni
- [Kaj01d] CM³