

Amélioration de processus : 10 bonnes raisons pour passer du CMM[®] au CMMI[®] ou un « iota » à forte valeur ajoutée

RICHARD BASQUE

Préambule: Cet article s'adresse à ceux qui ont déjà lu ou utilisé le modèle CMM et qui se demandent pourquoi il leur faudrait passer au CMMI. Les lecteurs qui ne croient pas aux bénéfices d'une approche centrée sur les processus devraient passer outre. Même chose pour ceux qui ne savent pas que CMM veut dire Capability Maturity Model[®] (Modèle d'évolution des capacités, tel que traduit en français dans le cadre d'un projet piloté par l'auteur au milieu des années 90). Ce n'est pas par élitisme que je donne ces précisions. Je sais, simplement, que votre temps est précieux. Lecteurs qui nous quittez, je vous salue et rappelez-vous : si un jour vous hésitez à passer du CMM au CMMI (vous saurez alors de quoi il en retourne), il y a au moins dix bonnes raisons de le faire. Vous reviendrez alors à cet article et le trouverez, cette fois, fort utile (du moins, je l'espère !). Et si vous êtes déjà passés aux CMMI, à la bonne heure ! Voici 10 bonnes raisons de vous convaincre que vous aviez bien fait !

Dans ce qui suit, le sigle CMM est utilisé pour désigner spécifiquement le (Software) Capability Maturity Model version 1.1 publiée en 1993 et le sigle CMMI pour désigner le Capability Maturity Model Integration version 1.1 publié en 2002. Le SEI^{ms} (Software Engineering Institute), qui publie ces modèles et veille à leur évolution, a annoncé officiellement qu'il abandonnerait le support du CMM au profit du CMMI à la fin de 2005 et invite les utilisateurs du CMM à passer au CMMI. Certains hésitent encore...

1. INTRODUCTION

Bon voilà : nous sommes entre personnes qui partagent les mêmes convictions. Vous savez que le CMM que l'on utilise aujourd'hui, la version officielle, est la version 1.1 éditée par le SEI^{ms} en 1993. Onze années d'utilisation intensive ont suscité de nombreuses propositions d'amélioration. Faut-il, dès lors, s'étonner qu'on en soit venu à imaginer une version 1.2 voire une version 2 ? C'est donc sans surprise et avec plaisir que la communauté utilisatrice du CMM a vu le SEI, vers 1995, annoncer qu'une version améliorée verrait le jour autour de 1997 - 1998. Pourtant, le principal bailleur de fonds du SEI

annonçait à l'auditoire de la conférence SEPG1998 à Chicago que le projet de CMM v2 était bel et bien terminé au profit d'un nouveau projet nommé CMMI !

Que voulait faire au juste le SEI en remplaçant le projet CMM version 2 par le projet CMMI ? Le SEI cherchait à faire émerger de la jungle des modèles de meilleures pratiques (SA-CMM, People CMM, SE-CMM, ...) un modèle qui soit intégrateur. Plusieurs ont vu le « *quagmire* » ou l'assiette de spaghettis des modèles, normes, guides, et j'en passe, sur le site du Software Productivity Consortium (<http://www.software.org/>)

quagmire). Il suffit d'y jeter un coup d'œil pour constater à quel point le monde des modèles sur les processus ressemble effectivement à une jungle et pour sentir le besoin d'épurer, d'élaguer, de synthétiser... Voilà la motivation principale de ce projet CMMI. En même temps, le SEI voulait proposer une approche holistique aux organisations qui, préoccupées par des performances décevantes en matière de développement logiciel, souhaitaient s'améliorer par le biais d'une approche centrée sur les processus : pourquoi s'arrêter à améliorer son processus de développement ou de maintenance logiciel quand la solution que l'on développe pour répondre aux besoins des clients comprend, certes, du logiciel mais ne repose pas *seulement* sur le logiciel ? Le SEI réalisait que le logiciel constituait un maillon dans une chaîne plus vaste, constituée de matériel, mécanique, électronique, documents, formation et... logiciel !

Quelques mois ont suffi pour que l'équipe CMMI réalise avoir besoin d'un chef de projet à l'écoute de la communauté d'affaire et de la communauté internationale. On a désigné un tel chef de projet CMMI ; il s'est installé à la barre et a fait prendre au navire CMMI le cap du succès. Il s'en est fallu de peu qu'il échoue sur les récifs de l'inquiétude des « CMMistes » de première cuvée (celle de 1993). Chers lecteurs : si vous croisez sur votre chemin un pauvre hère, CMM version 1.1 à la main et larmes aux yeux, compatissez à sa douleur et bercez-le de l'hymne si doux du regretté Ferré (« *Tout s'en va...* ») qu'a rejoint ce cher Harrison (« *All things must pass...* ») ! J'ai failli être du nombre ! Il m'a fallu plus d'une année, moi, tombé dans la marmite du CMM en 1990 comme un Obélix moderne, pour voir enfin les avantages très clairs que présentaient le CMMI sur mon gros bouquin CMM qui était devenu confortable comme de vieilles pantoufles.

Si donc vous êtes prêts à renoncer au confort de vos pantoufles CMM bien chaudes pour découvrir le bienfait que procure un CMMI bien amené, lisez ce qui suit. Je vous livre mes dix bonnes raisons pour franchir le Rubicon.

RAISON 1 : CMMI COUVRE UN DOMAINE CIBLE QUI COLLE MIEUX À LA SOLUTION GLOBALE

Rares sont les organisations qui ne produisent que du logiciel. En existe-t-il d'ailleurs vraiment ? Même les éditeurs de logiciel livrent leur logiciel enrobé non pas de chocolat (hélas, parfois, car la pilule est amère) mais d'un CD dans un joli coffret avec un manuel d'installation et d'utilisation. Il y a donc au moins ces produits annexes (documentation et CD par exemple) qui doivent être *intégrés* avec le code lui-même. Donc, même dans ce cas extrême, la solution comprend un ensemble plus vaste que le code.

Par ailleurs, la plupart des organisations qui développent des produits fonctionnant avec du logiciel cachent le logiciel sous des couches de mécanique, d'électronique, de papier-documentation, de matériaux de toutes sortes. Pensons aux téléviseurs, aux jeux vidéos, aux voitures, aux trains, aux avions, aux guichets bancaires, bref à tout ce qui nous entoure. Le logiciel est là mais on ne le voit pas. Il est intégré au sein d'une solution qui a impliqué, pour son développement, bien d'autres disciplines que celle de l'informatique. Le CMM ne couvrait que la discipline du logiciel. Il débutait lorsque le groupe de développement logiciel recevait le sous-ensemble des spécifications de solution qui devaient être implémentées par du logiciel. Le CMMI est holistique : il couvre toutes les disciplines impliquées dans la solution. Il débute lorsqu'un client (ou un public cible pour les produits de consommation de masse) exprime un besoin qui doit être élucidé pour devenir une solution opérationnelle. Le champ opératoire du CMMI est donc beaucoup plus vaste que le CMM. Ce n'est pas une qualité en soi. Sa qualité est plutôt de couvrir bien, complètement et de façon cohérente ce qui est utile au développement de tout l'ensemble - solution (logiciel, matériel, électronique, etc.) plutôt que d'une seule partie (logiciel).

RAISON 2 : CMMI RASSEMBLE LES TROUPES POUR MIEUX COMBATTRE LES PÉRILS DES PROJETS

Je ne sais si la guerre froide a fait ses ravages sur toute la planète. Je parle de la guerre des informaticiens et des ingénieurs. Je sais qu'en France, on parle d'ingénieurs logiciel. Mais au Québec, le mot « ingénieur » est réservé aux membres d'un ordre professionnel qui a vu le jour avant la naissance de l'informatique. Son périmètre n'incluait pas explicitement la discipline informatique. Dans les écoles de génie, on formait des ingénieurs (sans relation à l'informatique) et dans les universités et autres écoles spécialisées, on formait des informaticiens. Les choses ont évolué et les distinctions entre disciplines deviennent de plus en plus floues. Mais sur le terrain, la méfiance souvent inexplicable des uns envers les autres a créé un univers plutôt fragmenté : d'un côté les ingénieurs (non logiciel) et de l'autre les informaticiens. Ces derniers ont surgi à l'époque du Flower Power et ont adopté le mouvement quasi anarchique qui a culminé en mai 68. Les informaticiens étaient, à leur façon, des soixante-huitards. Ce qui a eu l'heur d'horripiler les diplômés des grandes écoles de génie, plutôt du genre sérieux. En entreprise, ceux-ci se consacraient à élaborer les solutions dans ce qu'elles avaient de plus visible : mécanique, électronique, matériau, etc. Pendant ce temps, de plus en plus de barbus ou filles aux jupes longues codaient parfois bercés par les effluves

piquants des herbes plus ou moins licites. J'exagère certes mais il y avait bien deux cultures, deux mondes. Entre les deux, une sorte de mur de Berlin au-dessus duquel on voyait parfois un manuel de spécifications lancé par les sérieux ingénieurs à l'intention des informaticiens cachés de l'autre côté du mur. Les premiers l'avaient élaboré en dialoguant avec le client, en amont, et le faisaient retomber au milieu du désordre des *code people* en aval. Était-ce encore les effluves étranges ou autre chose ? Ces derniers n'arrivaient pas toujours à saisir de quoi on leur causait. Mais beaux joueurs (et puis cool, quoi, la musique qui envahissait les locaux des codeurs adoucissant les mœurs), les informaticiens se mettaient à la tâche relançaient bientôt au-dessus du mur leur grand œuvre qu'ils appelaient code et qui était rédigé dans une langue incompréhensible, sauf par ceux de la secte. Saisi au bond par les sérieux messieurs dames ingénieurs, il devenait bientôt le maillon le plus faible d'une chaîne qu'on n'arrivait plus à fermer pour livrer le paquet cadeau au client. À qui la faute ? Que de têtes ont roulé ! Dans les deux camps. Et chacun de brandir *son* modèle de meilleures pratiques, écrit dans un langage rébarbatif (à dessein ?) pour l'autre camp. Jusqu'au jour où on réalisa que le mur qui les séparait pouvait tomber. Que les projets naviguaient sur un seul navire et qu'il est inutile qu'un camp tire à bâbord et que l'autre tire à tribord : ça ne peut que ralentir le navire ! Le CMMI s'infiltra alors dans cette grande brèche et proposa un monde de paix entre les mécaniciens, les électroniciens, les « documenticiens », les « matériiciens » et les informaticiens. Soyons tous des « soluticiens » ! Si on craint encore cette mutation, on peut toujours faire des projets en les gardant dans des compartiments plus ou moins étanches. Mais au moins tous auront accès à l'espéranto des meilleures pratiques, le CMMI, et abandonneront leurs pratiques et comportements sectaires.

RAISON 3 : CMMI EST MULTIDISCIPLINAIRE ET RESTE OUVERT AUX ÉVOLUTIONS

On l'a vu ci-dessus : le CMMI est ouvert à plusieurs disciplines. Mais il y a encore mieux. Dans son énoncé même, le CMMI se présente comme un *work-in-progress*. Il imploiera les corpus de pratiques que les responsables des évolutions du modèle CMMI voudront laisser entrer dans le giron du CMMI. Ainsi, dans la cuvée existante, le CMMI couvre les disciplines suivantes : l'ingénierie de logiciel, l'ingénierie de système, le développement en équipe multidisciplinaire et le développement en équipe intégrant des fournisseurs. La communauté du CMMI s'est structurée et s'est dotée d'un bureau des modifications qui veille à son évolution. Si des gens souhaitent ajouter des pratiques, disons de sécurité, de gestion de

personnel ou de quoi que ce soit qui peut s'exprimer à travers un corpus de connaissances spécifiques, alors ils peuvent soumettre un projet de changement au bureau des modifications pour ajouter une discipline au CMMI. En ce sens, le CMMI est une structure d'accueil qui intégrera au fil du temps de plus en plus de pratiques dans des disciplines sans cesse en évolution.

RAISON 4 : CMMI INTÈGRE TOUTES LES DEMANDES D'ÉVOLUTION ACCEPTÉES PAR LE COMITÉ CMMI VERSION 2

Dans les années qui ont précédé l'avènement du CMMI, les adeptes du CMM ont patiemment proposé des milliers d'améliorations. Les retours d'expérience ont été soigneusement analysés et soumis à la sagacité de comités de sages puis de la grande communauté. Des versions de travail ont circulé et furent abondamment commentées. On a patiemment collecté tout ce savoir pour le réintégrer dans la version 2 du CMM. Mais cette version 2 est restée sous les voûtes du palais CMM. Pendant que se mettait en marche la grande machine du projet CMMI, quelqu'un a eu l'idée brillante de monter une grande procession vers la voûte du CMM version 2, de prendre le trésor et de l'amener dans le projet CMMI. Ainsi, la version 2 du CMM s'est-elle trouvée absorbée dans le CMMI à la satisfaction de tous !

RAISON 5 : CMMI PARLE MIEUX ET EST PLUS CARTÉSIEEN

Le CMM parle un langage parfois étrange. Ayant été au cœur de sa francisation, j'en sais quelque chose. Quand on traduit, on doit comprendre très intimement. Or, il nous est arrivé assez souvent, la joyeuse bande des CMMistes francophones, de buter sur du jargon anglo-américain même difficile à décoder par des purs Américains. Sans compter les discussions parfois âpres entre les cousins de France, du Québec, de Belgique, de Suisse et de tous les autres pays francophones autour de certaines expressions. À part ce phénomène culturel inévitable quand on rassemble des peuples partageant la même langue, force fut de constater que la version anglaise était parfois rédigée assez rudement, voire confusément. Par exemple, les énoncés de pratiques d'institutionnalisation, qui devaient en principe se ressembler un peu, beaucoup, passionnément d'un secteur clé à l'autre, se payaient le luxe de variations qui suscitaient des débats d'étymologistes. Le groupe de travail CMMI a vu ces écueils et les a évités. Il a appliqué une logique froide mais efficace, toute cartésienne, dans les énoncés de pratiques communes d'institutionnalisation qui, au contraire du CMM, se répètent de façon identique d'un secteur à l'autre.

RAISON 6 : CMMI EST ALIGNÉ SUR LA NORME ISO/IEC 15504

Vers le milieu des années 90, des britanniques, férus de normes ISO, lançaient un projet qui allait en rallier plusieurs. Surtout quand on annonce que le projet serait aux couleurs d'un groupe de Girls bien connues : SPICE (Software Process Improvement and Capability dEtermination). Le navire du SPICE project est parti en grande pompe, battant pavillon international ! Les grands prêtres du monde ISO ont donné ses chiffres de noblesse à la norme qui allait sortir des cartons de SPICE ; tu t'appelleras 15504 ont-ils décidé au berceau du nouveau poupon. Le poupon devait rassembler le monde autour d'un énoncé universel de pratiques cibles et de moyens pour évaluer des pratiques réelles de développement. En fait, il s'est avéré finalement que la norme ne serait pas le modèle de pratiques universelles mais plutôt des exigences de modèles et qu'il faudrait les instancier dans des modèles. Dès lors, à l'annonce de l'arrivée prochaine du nouveau bébé 15504, le SEI a envoyé ses fées pour faire des vœux. L'un des vœux des fées SEI fut que le CMMI ressemblât en tout point au nouveau-né 15504. Puis, le SEI s'est mis à l'ouvrage. Aujourd'hui, les gens du SEI maintiennent le cap. Ils affirment que le CMMI respecte les énoncés de modèle de 15504. La démonstration officielle et patente reste à faire mais ceux qui connaissent les deux reconnaissent énormément les traits de 15504 dans la figure de CMMI. L'avantage ? Quand un client demandera la conformité à la norme 15504, les adeptes du CMMI pourront dire : « Présents ! ».

RAISON 7 : CMMI VA PLUS À FOND DANS LES PRATIQUES D'INGÉNIERIE

Pauvre codeurs, pauvres testeurs, pauvres analystes, pauvres « ingénieurs » du logiciel. Le CMM ne leur donnait qu'une ou deux pratiques. Ils se sentaient délaissés au regard des 300 autres pratiques qui touchaient les uns et les autres, mais pas eux. Un peu plus et on les entendait répéter la célèbre réplique : « Si j'aurais (sic) su, j'aurais (resic) pas venu ! ». Le projet CMMI les a secoués. « Non Jeff, t'es pas tout seul... ». Voici des pratiques, des buts et des secteurs et voici mon modèle qui ne bat que pour vous. Pas à ce point là. Mais quand même... Ils en voulaient des secteurs : on leur a donné du « Requirements Development », du « Technical Solution », du « Integration », du « Validation », du « Verification ». Ils ont lu et se sont dit : enfin on parle de nous ! Le regretteront-ils quand on utilisera ces fruits, ces feuilles et ces branches pour examiner leurs pratiques et les coter ?

RAISON 8 : CMMI INTRODUIT DES ASPECTS ESSENTIELS OUBLIÉS PAR LE CMM

En plus de laisser les « vrais travailleurs du logiciel » sur leur appétit (l'objet du point ci-

dessus), en plus de ne pas appliquer de façon cohérente les pratiques communes et d'en oublier certaines dans plusieurs secteurs, le CMM avait commis quelques bavures. Ainsi, le CMM demandait de mesurer mais on oubliait de préconiser un alignement intelligent de ces mesures avec les objectifs d'affaire des organisations. Le CMMI a corrigé le tir avec « Measurements and Analysis ». Le CMM abordait les risques mais avec une pudeur qui laissait presque pantois. Le CMMI est plus agressif grâce à « Risk Management ». Le CMM souhaitait des décisions rationnelles et réfléchies quand vient le temps de décider d'acheter ou de développer mais ne le disait pas clairement. Ceci est devenu hyper clair (certains diront presque trop) avec « Decision Analysis and Resolution ».

RAISON 9 : CMMI EST DISPONIBLE EN PLUSIEURS DÉCLINAISONS POUR MIEUX COLLER À LA RÉALITÉ DE CHACUN

On a vu que le CMMI intégrait plusieurs disciplines. Mais il est possible de ne choisir que les disciplines qui collent à la réalité de l'organisation. En ce sens, le CMMI est modulaire et on ne prend que ce dont on a besoin. Par ailleurs, au lieu de trancher entre la vision étagée (utilisée dans le CMM) et la vision continue (préconisée par la norme ISO 15504) pour l'architecture de base, le projet CMMI a rendu un jugement à la Salomon : vous aurez les deux ! Ainsi, le même ensemble de pratiques peut se voir à travers la lunette « Staged » ou la lunette « Continuous » ; on appelle « représentations » ces deux façons de voir les mêmes pratiques. L'une plaira à ceux qui veulent une vision globale et organisationnelle ; l'autre plaira à ceux qui veulent fouiller plus en profondeur et aborder les défis centrés sur les « *process areas* » une à la fois.

RAISON 10 : CMMI DONNE DES INDICATIONS PLUS CLAIRES SUR LES PRODUITS TYPES DE CHAQUE PRATIQUE CIBLE

Sous chaque pratique ou presque, on retrouve, tant dans le CMM que dans le CMMI, au moins quelques lignes et parfois des pages d'explications qui aident à l'interprétation et guident le déploiement. Le projet CMMI a eu la bonne idée de mieux structurer ce texte additionnel. En particulier, il identifie très clairement les produits de travail (ou biens livrables intermédiaires) qu'on s'attend à trouver si on prétend institutionnaliser telle ou telle pratique. On appelle cette rubrique les « Typical Work Products ». Ceci aide grandement une organisation à imaginer de façon très tangible ce qu'on attend d'elle dans le déploiement de chaque pratique : drôlement utile pour se comprendre, pour enseigner, pour illustrer et pour se préparer à une évaluation !

2. CONCLUSION

Ceux qui m'auront suivi jusqu'ici pourront penser que je suis vendu à CMMI. Pourtant, j'ai déjà rédigé quelques propositions d'amélioration pour la prochaine cuvée. Pourtant, je suis un peu inquiet de l'appétit vorace que pourrait avoir le CMMI en implorant, tel un ogre qui aurait perdu toute retenue, des disciplines de plus en plus nombreuses, devenant obèse et parfois inabordable. Pourtant son cartésianisme implacable souffre de quelques lacunes, bien cachées mais néanmoins réelles. Et il y a la discipline IPPD qui me semble mal cadrée, mal définie. Mais dans l'ensemble, je confesse mon enthousiasme certain pour ce grand modèle. George Box disait : « Tous les modèles sont faux. Certains sont utiles » CMM était utile. CMMI l'est encore plus. Le CMM est (presque) mort. Vive le CMMI !

3. RÉFÉRENCE UTILE

Les pages du SEI sur le modèle CMMI d'où on peut, entre autres choses, récupérer les fichiers du modèle CMMI : <http://www.sei.cmu.edu/cmmi/adoption/cmmi-start.html>

On y trouvera aussi tous les détails sur les dates butées pour le support du CMM par le SEI.

L'AUTEUR

Richard Basque (*Richard.Basque@alcyonix.com*) est président et fondateur de la société Alcyonix (*www.alcyonix.com*) basée au Canada qui



offre des services en amélioration de processus. Sa société est partenaire autorisé par le Software Engineering Institute (SEI) pour les services de formation au CMMI et les services d'évaluation SCAMPI. Il est lui-même autorisé par le SEI en tant que chef évaluateur avec les méthodes SCAMPI, CBA IPI et SCE. Il est aussi autorisé par le SEI comme instructeur pour les cours officiels du SEI sur CMMI1.1 et SW-CMM 1.1. De plus, en tant que « SEI Visiting Scientist » à temps partiel, il est occasionnellement invité à former des chefs évaluateurs au SEI.

Il a participé à plus d'une trentaine d'évaluations de processus officielles au Canada, aux États-Unis, en Europe et en Amérique du Sud. Il a piloté la francisation du SW-CMM qui a été adoptée par le SEI comme la version officiellement équivalente à la version originale en anglais. Il prépare présentement, pour les Éditions Dunod un ouvrage en français sur le CMMI.

NOTES

* CMM, CMMI, Capability Maturity Model et Carnegie Mellon sont enregistrés auprès du U.S. Patent and Trademark Office par Carnegie Mellon University.

^{ms} SEI est marque de service de Carnegie Mellon University

ACTES DES 16^{èmes} JOURNÉES INTERNATIONALES ICSSEA

« Génie Logiciel & Ingénierie de Systèmes et leurs Applications »

2-4 décembre 2003

5 volumes, totalisant 900 pages - Publication du CNAM/CMSL - ISSN : 1637-5033

- Architectures
- Conduite de Projets
- Estimation de Projets
- Évolution & Maintenance
- Ingénierie des Connaissances
- Ingénierie de la Qualité
- Ingénierie de la Sécurité
- Ingénierie de Systèmes
- Ingénierie des Exigences
- Modélisation
- Processus
- Qualité du Logiciel
- Réutilisation & Composants
- Revues & Inspections
- Spécifications
- Systèmes d'Information
- Systèmes Embarqués
- Test
- Travail coopératif
- Web, Nomadisme & Ubiquité

Renseignements et Commandes

CNAM/CMSL

C/o Chaire d'Intégration de Systèmes - 292, rue Saint-Martin - 75141 Paris Cedex 03
Tél : 01 40 27 23 83 ou 01 40 27 22 58 - Télécopie : 01 40 27 23 77 - Courriel : rault@cnam.fr