

Scrum et CMMI Niveau 5 : La Potion Magique pour les Guerriers du Code

Jeff Sutherland, Ph.D.
Patientkeeper Inc.
jeff.sutherland@computer.org

Carsten Ruseng Jakobsen
Systematic Software Engineering
crj@systematic.dk

Kent Johnson
AgileDigm Inc.
kent.johnson@agiledigm.com

Résumé

Les projets combinant les méthodes agiles avec CMMI¹ rencontrent plus de succès dans la production d'un logiciel de meilleure qualité qui répond plus efficacement et plus rapidement aux besoins des clients. La société Systematic Software Engineering est CMMI niveau 5 et utilise les principes du Lean pour optimiser les processus logiciels. Les premiers projets pilotes chez Systematic ont démontré une productivité des équipes Scrum quasiment double par rapport à des équipes traditionnelles. Les autres projets ont démontré qu'une approche de développement logiciel pilotée par les tests réduisait de 40% le nombre de défauts constatés au cours des tests finaux.

Nous affirmons que l'usage combiné de Scrum et CMMI apporte une plus grande adaptabilité et prédictibilité que si on les utilisait séparément et nous proposons aux entreprises une manière de les combiner.

1 Introduction

Le développement réussi de logiciels est remis en cause par la capacité du fournisseur à gérer la complexité, l'innovation technologique, et les exigences de changement. Les méthodes Agile et CMMI répondent toutes les deux à ces défis, mais sont très différentes dans l'approche et leurs modes d'application.

La gestion de la complexité requiert une discipline des processus alors que Scrum favorise la capacité d'adaptation. CMMI met en œuvre une maîtrise des processus, Scrum augmente la capacité d'adaptation. Ce document analyse les effets de l'introduction des pratiques Agiles dans une entreprise CMMI niveau 5.

2 Scrum et CMMI : une potion magique

Systematic, une entreprise indépendante dans les systèmes logiciels, a été créé en 1985 et emploie plus de 400 personnes dans le monde entier avec des bureaux au Danemark, aux Etats-Unis et au Royaume-Uni. Les solutions développées par Systematic sont utilisées par des dizaines de milliers de personnes dans la défense, la santé, la

¹ © Capability Maturity Model, CMM and CMMI sont des marques déposées auprès de l'U.S. Patent et du Trademark Office.

construction et les industries de services. Systematic a été évaluée le 11 novembre 2005 avec la méthode SCAMPI^{SM2} et déclarée conforme CMMI niveau 5.

Chez Systematic, les pratiques CMMI de niveau 5 ont réduit de 42% la reprise de code, maintenu l'écart de précision de l'estimation à moins de 10%, et garantit que 92% des jalons étaient livrés en avance ou dans les temps. Parallèlement, les heures supplémentaires ont été sensiblement réduites.

Plus important encore, Systematic a transformé plus de vingt ans d'expérience en un référentiel unifié de processus, utilisé par tous les projets logiciels. Les statistiques projets sont systématiquement collectées et analysées pour disposer en permanence d'une vision de la maturité et de la performance de l'organisation.

L'utilisation d'un référentiel commun partagé facilite le passage d'un projet à un autre et le partage de l'expérience et des leçons entre les projets. La vision du niveau de maturité et de performance des processus permet d'évaluer et comparer la performance des nouveaux processus par rapport à celle des processus existants. C'est la base de l'amélioration continue.

En résumé, Systematic a été en mesure de livrer le client avec ce qu'il avait commandé en respectant les délais, les coûts et la qualité, tout en déployant un effort réduit à 69% par rapport à une entreprise CMMI Niveau 1 [1, 2], ceci avant l'introduction de Scrum.

CMMI Niveau 5 est une exigence de plus en plus forte de la part des clients et une des clés pour obtenir de grands contrats, en particulier au sein de la défense et de la santé. Les clients reconnaissent que CMMI Niveau 5 fournit un haut niveau de prédictibilité et des produits mieux conçus en termes de capacité à monter en charge, maintenabilité, adaptabilité et fiabilité.

CMMI fournit une vision processus sur ce qu'il est nécessaire de faire pour maintenir la maturité et la **discipline** d'une organisation capable de prévoir et améliorer la performance de ses projets. La méthode Scrum est un guide pour améliorer l'efficacité des projets en offrant une très grande souplesse et **adaptabilité**.

Lorsqu'on combine les deux, une potion magique apparaît, dans laquelle l'ingrédient Scrum garantit que les processus sont mis en œuvre efficacement

² SM Capability Maturity Model Integration et SCAMPI sont des marques déposées par la Carnegie Mellon University.

tout en encourageant le changement et l'ingrédient CMMI garantit que tous les processus pertinents sont pris en compte. Scrum diminue aujourd'hui de 50% la part de travail (traitement des défauts, reprise de code, charge totale et pilotage) par rapport à notre précédente implémentation CMMI Niveau 5, tout en maintenant le même niveau de discipline dans l'application des processus.

Séparément, CMMI et Scrum présentent des avantages réels, mais aussi des pièges. Une entreprise Agile peut correctement appliquer Scrum, mais aussi ne pas y parvenir par manque de structure (voir chapitre 3) ou par une gestion/exécution des processus insuffisante voire incohérente. CMMI peut aider les entreprises Agiles à définir un cadre d'application des méthodes agiles plus cohérent et identifier les processus concernés.

Une entreprise peut être conforme à CMMI, mais ne pas parvenir à atteindre une performance optimale en raison d'une mise en œuvre inadéquate des processus. Scrum et les autres méthodologies Agile peuvent aider de telles entreprises à mettre en œuvre efficacement les exigences et processus CMMI.

2.1 Expérience Lean chez Systematic

Systematic a pris la décision stratégique d'utiliser les principes du Lean pour ses futures améliorations après l'atteinte du Niveau 5 de CMMI. L'usage du Lean a démontré des résultats remarquables depuis de nombreuses années dans des domaines tels que la construction automobile, et en raison de sa popularité, a été adapté à d'autres domaines, y compris le développement de logiciels. Systematic a identifié le développement logiciel en mode Lean comme étant le plus pertinent pour Systematic.

Appliqué le Lean dans le développement logiciel, en tant que fil conducteur pour les améliorations futures dans une entreprise évaluée CMMI Niveau

5, repose sur l'adoption d'une structure Lean et Agile dans la mise en œuvre des processus CMMI ; Systematic a mis l'accent sur la conduite du changement Lean dans l'esprit du Manifeste Agile.

Les compétences Lean ont été acquises en distribuant les photocopies de livres, des formations formelles et informelles et des activités d'échanges. Nos Chefs de Projets ont été formés au développement de logiciel en mode Lean, et Marie Poppendieck est venu tenir un séminaire sur le sujet chez Systematic.

Ce séminaire a permis de comprendre la mentalité Lean et Agile. Les dépendances entre les principes et les outils de développement logiciel en mode Lean ont été analysés par Carsten Jakobsen, responsable de la conduite du changement Lean, et a abouti au modèle schématisé dans le Tableau 1. Le modèle croise les outils (T) et les principes (P) du développement logiciel en mode Lean selon leurs dépendances, les éléments de la droite dépendant d'un ou plusieurs éléments de la gauche. Le modèle a permis de prioriser notre travail sur les outils. La plupart des outils ont été considérés comme de bons candidats pour commencer.

La plus importante contribution à une première sélection d'outils Lean est une analyse montrant les possibilités d'amélioration avec un bon rapport coût-bénéfice. Des études internes chez Systematic montrent que le coût de résolution d'un défaut augmentait de 1,6 heures lorsqu'il était détecté dans la phase de codage, de 12 heures lorsqu'il était détecté dans la phase de test et de 23,7 heures lorsqu'il était détecté dans la phase de maintenance. Par conséquent, les améliorations qui pourraient mener à éliminer ou détecter tout défaut dès les phases amonts présentent un effet de levier intéressant. Nous avons également constaté que notre vision de la qualité, nous a progressivement conduit à allonger les cycles de tests.

	Valeur	Flux de valeur	Tiré par la demande	Perfection
Ingénierie logicielle	<u>P6 Intégrité</u> T19 Refactoring T20 Test	<u>P2 Développer</u> <u>L'Apprentissage</u> T5 Synchronisation T4 Itérations	<u>P2 Développer</u> <u>L'Apprentissage</u> T3 Feedback T6 Bases de développement	<u>P6 Intégrité</u> T18 Conceptuelle T17 Perçue
Management	<u>P1 Création de Valeur</u> T1 Identifier le gaspillage T2 Flux de Valeur	<u>P4 Livrer Rapidement</u> T11 Théorie des files d'attente T12 Coût du retard	<u>P7 Voir l'Ensemble</u> T22 Contrats T21 Mesures T10 Tiré par la demande	<u>P3 Différer l'Engagement</u> T7 Options T8 Différer l'engagement T9 Prise de décision
Équipe	<u>P5 Equipe autonome</u> T16 Expertise	<u>P5 Equipe autonome</u> T14 Motivation	<u>P5 Equipe autonome</u> T15 Leadership	<u>P5 Equipe autonome</u> T13 Autodétermination

Tableau 1 - Développement logiciel en mode Lean

2.2 Expérience de Systematic à partir des projets pilotes

L'analyse ci-dessus des opportunités d'amélioration et des relations de dépendance avec le Lean, menée chez Systematic, a conduit à la décision de rechercher des améliorations sur la base des principes de développement Lean que sont l'Accroissement de l'Intégrité, le Développement de l'Apprentissage, la Rapidité des Livraisons. Ces outils Lean nous ont donné l'inspiration pour examiner et expérimenter la méthode Scrum. Sur une période d'environ 4 mois, deux petits et deux grands projets pilotés en mode Scrum et basés sur des tests le plus tôt possible des stories.

2.2.1 Scrum

Le premier projet pilote a été lancé suite à un appel d'offres, lorsque Systematic – inspiré par les principes du Lean – a proposé un planning de livraison bihebdomadaire et répondu explicitement aux demandes d'engagement et de réactivité du client. Le projet avait une équipe de 4 personnes et concernait un client du gouvernement Danois.

L'une des principales raisons de l'attribution du contrat à Systematic, a été son engagement de livrer un code qui fonctionne bi-hebdomadairement et donc d'être extrêmement transparent vis-à-vis du client. Au cours du projet, un grand niveau de communication a été maintenu entre l'équipe, le client et les utilisateurs. Cela a été identifié comme l'une des principales raisons pour obtenir un niveau élevé de satisfaction de la part du client.

Le planning de livraison et l'engagement du client ont permis de détecter très tôt les problèmes techniques. Si une approche traditionnelle avait été utilisée, ces problèmes auraient été identifiés beaucoup plus tard avec des impacts négatifs sur le coût et le calendrier.

Pourtant, la productivité de ce petit projet est bien restée conforme à la productivité de référence pour les petits projets. Un autre petit projet, avec une équipe de 5 personnes pour un client du secteur de la Défense, a également utilisé Scrum et a obtenu une productivité, des indicateurs qualité et de satisfaction du client identiques.

Chez Systematic, la productivité d'un projet est définie comme le nombre total de lignes de code produites divisé par la charge totale du projet en heures. Les données sont corrélées avec des informations relatives au langage de programmation, le type de code : nouveau, réutilisation ou test.

Systematic a créé et maintient un référentiel de productivité (PPB~Productivity Performance Baseline) à partir de projets terminés et basée sur une taille estimée en heures [4]. Les données montrent que la productivité est élevée sur de petits projets et diminue avec la taille du projet. Le

référentiel de productivité de Systematic est divisé en deux groupes : les petits projets de moins de 4000 heures et les grands projets de plus de 4000 heures. La productivité des petits projets représente 181% de la productivité des grands projets.

Lorsque l'on compare les projets utilisant Scrum à la productivité de référence, on constate qu'elle ne change pas significativement ; mais la productivité des grands projets montre une augmentation de 201%. Comme mentionné ci-dessus, la conduite de grands projets s'est améliorée, et il n'est donc pas possible d'attribuer uniquement ce gain à Scrum.

Toutefois, les personnes concernées sont toutes d'accord pour dire que Scrum représente une partie importante de cette amélioration.

A l'avenir, chez Systematic, nous pressentons que les grands projets utilisant Scrum doubleront leur productivité. Chez Systematic, les petits projets montrent déjà une forte productivité. Nous pensons que c'est parce que les petits projets avaient toujours été gérés d'une manière similaire à Scrum chez Systematic. Cependant la qualité et la satisfaction du client semblent s'être améliorée et nous pensons que c'est parce que Scrum a permis de mieux comprendre comment gérer de manière efficace les petits projets.

2.2.2 Les tests au plus tôt

Un grand projet avec une équipe de 10 personnes travaillant sur un système de messagerie militaire. Ce projet s'est inspiré du principe Lean de l'Accroissement de l'Intégrité pour trouver un moyen de tester au plus tôt ; le résultat a été que l'équipe a mis en œuvre une approche basée sur les tests au plus tôt des stories. Le nom « Basé sur des stories » est inspiré de XP, mais notre approche a inclus de nouveaux aspects tels que : contributions incrémentales à court terme, inspections et piloté par les fonctionnalités.

L'idée d'un développement basé sur des stories était de découper les fonctionnalités – généralement estimée à des centaines d'heures de travail – en petites stories représentant 20 à 40 heures de travail. La mise en œuvre d'une story a conduit à une nouvelle procédure, où la première activité a été de décider comment la story pourrait être testée avant qu'un début de code ait été écrit. Ce test pourrait alors être utilisé comme critère de validation du développement de la story.

La procédure comporte quelques points de contrôle pour vérifier le travail produit, et décider si oui ou non le développeur peut passer à l'activité suivante de la procédure. Ces inspections sont légères, et peuvent généralement être réalisées en moins de 5 minutes

De nombreux avantages immédiats apparaissent suite au développement basé sur des stories. La combinaison d'une bonne définition du moment où une story est terminée, et des tests incrémentaux au plus tôt, ont fourni un aperçu très rapide de l'état

d'avancement du projet pour l'équipe et les autres parties prenantes.

Le développement d'une série de petites stories plutôt que de parties d'une grande fonctionnalité donne une meilleure vision pour compléter une fonctionnalité jusqu'à ce que tous les critères permettent de dire qu'elle est terminée.

Ce projet s'est terminé plus tôt, et le nombre de d'anomalies de codage détectées lors des tests finaux a été réduit de 38% par rapport aux processus antérieurs.

Un autre projet avec une équipe de 19 personnes travaillant sur le module d'un système d'enregistrement de patients, a également travaillé selon un mode tests au plus tôt. Ce qui a permis de garantir que les activités de tests étaient intégrées au développement, tout en insistant sur le fait d'avoir une « vision globale » et de comprendre comment la solution s'intègre dans le métier du client. Pour chaque semaine, le projet a défini un objectif à atteindre. Le projet a garanti que les tests et les experts métiers étaient localisés au même endroit que les développeurs. Ceci a généré des discussions et réflexions entre testeurs, développeurs, utilisateurs expérimentés et architectes logiciels, avant ou très tôt dans le développement d'une nouvelle fonctionnalité. Du coup, le nombre de d'anomalies de codage résiduelles détectées dans les tests finaux a été réduit de 42% par rapport au processus précédent.

Sur la base de ces deux projets, il a été conclu que les activités de tests devaient être une activité intégrée au cycle de vie d'un projet, sachant que Scrum le propose de fait avec des équipes multifonctionnelles et des livraisons fréquentes au client. En outre, il a été décidé que la méthode de développement basé sur les stories serait la méthode de développement logiciel recommandée par défaut pour les projets.

2.2.3 Besoins réels

Un client a envoyé un appel d'offres avec un ensemble d'exigences fixé. Lorsque Systematic a répondu, nous avons exprimé notre inquiétude sur le fait que le périmètre et le contenu des exigences exprimées dans l'expression des besoins étaient au-delà des besoins réels du client.

Systematic a donc décidé de partager son estimation interne et détaillée des besoins de façon transparente avec le client, ceci afin de réduire le périmètre en supprimant des exigences non nécessaires ou trop chères par rapport au budget du client. Le client a accepté de revoir son expression des besoins ; le résultat obtenu est que les exigences et les prix ont été réduits de 50%.

Cette expérience confirme les résultats du Standish Group Study présentés par Jim Johnson au XP2002, et montrant que 64% des fonctionnalités d'un

contrat au forfait ne sont jamais ou rarement utilisées par les utilisateurs finaux.

Nous pensons que cela montre combien il est important d'avoir une discussion franche et ouverte avec le client, afin de connaître ses besoins réels. Le succès n'est pas atteint en faisant le plus grand projet, mais en faisant le projet qui fournit le plus de valeur au client, ce qui laisse le temps pour les développeurs de logiciels de travailler sur des besoins réels avec les autres clients. Cette stratégie est fortement préconisée par Scrum.

2.3 Adoption des méthodes Agiles

Le résultat des projets pilotes a été de deux ordres : d'une part, il a confirmé l'idée générale d'utiliser les principes du Lean en tant que moyen d'identification de nouvelles améliorations, et d'autre part, il a fourni deux améliorations spécifiques que sont Scrum et les tests au plus tôt basés sur les stories, en montrant comment les méthodes agiles pouvaient être adoptées, tout en maintenant la conformité avec CMMI. Un point important pour Systematic était que l'adoption de ces méthodes agiles ne requièrent que des petits ajustements des processus existants.

La principale différence fut d'adopter des principes lean et agile dans l'interprétation des processus existants.

L'évaluation des résultats des projets pilotes a conduit à la décision d'adopter Scrum et les tests au plus tôt basé sur des stories. Ces méthodes sont maintenant les choix par défaut pour les nouveaux projets, et sont intégrés dans les descriptions de processus chez Systematic.

3 Guide pour combiner CMMI et Agile

3.1 CMMI peut améliorer l'Agile

Nous mettons l'accent sur la façon dont CMMI peut aider une organisation à institutionnaliser les méthodes Agile. Nous avons tous entendu parler des Méthodes Agiles par les uns comme une méthode de gestion de projet indisciplinée et par les autres qui prétendent être Agiles juste parce qu'ils « ne documentent rien ». Nous estimons que la valeur des méthodes Agiles ne peut être obtenue que grâce à une utilisation disciplinée. CMMI a un concept d'*Institutionnalisation* qui peut aider à mettre en œuvre ce besoin de discipline.

L'*Institutionnalisation* est défini dans CMMI comme « la manière fondamentale de faire des affaires qu'une organisation applique régulièrement dans le cadre de sa culture d'entreprise ». D'autres ont décrit l'institutionnalisation comme étant simplement « la manière dont nous faisons les choses ici ». Notez que l'institutionnalisation est un concept au niveau de l'organisation qui s'applique à des projets multiples.

CMMI appuie l'institutionnalisation par le biais des Pratiques Génériques (GP) associés à l'ensemble des processus. Pour les besoins de notre discussion, nous nous pencherons sur les 12 pratiques génériques associées aux niveaux de maturité 2 et 3 de CMMI [5] et la façon dont ils pourraient aider une organisation à utiliser les méthodes Agiles.

3.1.1 Établir et maintenir une politique d'organisation pour planifier et déployer des méthodes Agiles (GP 2.1)

La première étape vers l'institutionnalisation des méthodes Agiles est d'établir comment et quand elles doivent être utilisées dans l'organisation. Une organisation pourrait déterminer que les méthodes Agiles sont utilisées sur tous les projets ou un sous-ensemble de projets selon la taille, le type de produits, la technologie ou d'autres facteurs. Cette politique est un moyen de communiquer clairement sur l'intention de l'organisation concernant l'usage des méthodes Agile. Conformément au Principe Agile du face-à-face, une participation à "toutes les réunions plénières" ou la visite d'un haut responsable au cours d'une réunion de lancement d'un projet pourraient être utilisés pour communiquer sur cette politique.

3.1.2 Établir et maintenir le planning de déploiement des Méthodes Agiles (GP 2.2)

Le but est d'éviter que les méthodes agiles ne dérivent en une pratique indisciplinée. On s'attend à ce que le déploiement des Méthodes Agiles soit planifié et piloté. Cela devrait inclure une série d'étapes pour définir le minimum nécessaire à décrire ce qu'un projet doit faire. Ainsi que les aspects essentiels de mise en œuvre des 10 autres pratiques génériques dans un projet. Dans Scrum, certaines tâches du planning devraient être gérées dans un backlog produit et/ou un backlog de sprint, plus probablement dans un outil que dans un document.

3.1.3 Prévoir des ressources suffisantes pour déployer les Méthodes Agiles (GP 2.3)

Chaque projet veut, a besoin et attend des professionnels compétents, un financement adéquat, des moyens et des outils appropriés. Mettre en œuvre une activité pour gérer explicitement ces différents besoins, s'est avérée utile. En Scrum, par exemple, ces besoins peuvent être examinées et traitées lors de la réunion de planification du sprint et reconsidérées lorsque des changements importants se produisent.

3.1.4 Attribuer à quelqu'un la responsabilité et l'autorité du déploiement des Méthodes Agiles (GP 2.4)

Pour qu'un projet réussisse, un niveau de responsabilité et d'autorité claire doit être clairement défini. Habituellement, cela met en jeu une combinaison de descriptions de rôles et d'affectations. La définition de ces rôles mentionne un niveau de responsabilité et d'autorité. Par exemple, un projet en Scrum affectera à un ou plusieurs individus les rôles de Product Owner, de ScrumMaster et d'Equipe. L'Expertise dans l'Equipe est susceptible d'inclure un mélange d'experts du domaine, des ingénieurs systèmes, des ingénieurs logiciels, des architectes, des programmeurs, des analystes, des experts de l'Assurance Qualité, des testeurs, des ergonomes, ... Scrum attribue à l'équipe dans son ensemble, la responsabilité de la réalisation du logiciel. Le Product Owner est responsable de la spécification et de la priorisation du travail. Le ScrumMaster est responsable du suivi du respect du process Scrum. Le Management est responsable de la fourniture du bon niveau d'expertise à l'équipe.

3.1.5 Former les personnes effectuant le déploiement des méthodes Agiles (GP 2.5)

Le bon cursus de formation peut augmenter la performance des professionnels compétents et soutenir l'introduction de nouvelles méthodes dans une organisation. L'institutionnalisation de la Méthode Agile utilisée nécessite une formation cohérente. Cette pratique implique d'identifier les personnes à former, de définir exactement le cursus de formation nécessaire, et de réaliser cette formation. La formation peut être assurée au travers de différentes approches, formation sur le tas, monitorat, formations formelles. Il est important de définir un mécanisme d'évaluation pour s'assurer que la formation a eu lieu et est bénéfique.

3.1.6 Associer au produit un niveau approprié de gestion de la configuration (GP 2.6)

Le but du projet est de produire des produits livrables. Chaque produit est un ensemble de produits intermédiaires (code, manuels, fichiers de build, ...). Chaque produit intermédiaire a une valeur en traversant une série d'étapes qui augmentent leur valeur. Le concept de gestion de la configuration a pour objectif de protéger ces produits intermédiaires en définissant le niveau de contrôle adéquat, par exemple une version voire plusieurs versions de référence à utiliser dans le cadre du projet.

3.1.7 Identifier et impliquer réellement les parties prenantes (GP 2.7)

Impliquer le client en tant que réelle partie prenante constitue une force des Méthodes Agiles. Cette pratique identifie de plus le besoin de garantir un niveau suffisant d'implication des parties prenantes. Par exemple, si le projet dépend de la réactivité du client à chaque build ou sprint, et que le niveau d'implication ne répond pas aux attentes, il est alors nécessaire de communiquer au niveau approprié, individu ou groupe dans l'organisation, pour mener une action corrective si tant est que l'action corrective est du ressort de l'équipe projet. Dans des mises en œuvre très avancées de Scrum, ceci est souvent formalisé par des MetaScrum [6] où les parties prenantes servent de conseil d'administration pour le Product Owner.

3.1.8 Piloter et contrôler le déploiement des Méthodes Agiles et prendre des mesures correctives appropriées (GP 2.8)

Cette pratique consiste à mesurer l'état d'avancement réel du planning de déploiement et à prendre les mesures correctives nécessaires. Un pilotage au jour le jour en direct est un des points forts de la réunion Scrum quotidienne ; le Release Burndown Chart montre le reste à faire au début de chaque sprint, le Sprint Burndown Chart montre le nombre d'heures de travail restantes sur les activités. Scrum améliore l'efficacité du planning en permettant au Product Owner d'inspecter et d'adapter afin de maximiser le retour sur investissement, plutôt que de s'assurer du respect exact du planning.

3.1.9 Évaluer objectivement l'adhérence aux Méthodes Agiles et traiter la non-conformité (GP 2.9)

Cette pratique est basée sur le fait de disposer d'une personne qui n'est pas directement responsable de la gestion ou de réalisation du projet et qui évalue les activités réelles du projet. Certaines organisations mettent en œuvre cette pratique au travers, à la fois, d'une activité d'assurance qualité et d'une activité de coaching. Le concept de coaching correspond à de nombreuses Méthodes Agiles. Le ScrumMaster a la responsabilité principale de l'adhésion aux pratiques Scrum, le suivi des progrès, la suppression des obstacles, la résolution des problèmes personnels, et n'est généralement pas engagé dans la mise en œuvre de tâches du projet. Le Product Owner a la responsabilité principale de s'assurer que le logiciel répond aux exigences et a un niveau de qualité élevée.

3.1.10 Examen des activités et des résultats des Méthodes Agiles avec le top management et résolution des problèmes (GP2.10)

Le but de cette pratique est de faire en sorte que le top management ait une visibilité satisfaisante des activités du projet. Les différents managers ont des besoins différents d'information. Les Méthodes Agiles ont un niveau élevé d'interaction ; par exemple, Scrum a la réunion de planification de Sprint, les réunions quotidiennes, la réunion de revue de sprint et la réunion de rétrospective du sprint. Les besoins du management sont satisfaits par le niveau de transparence offert par les données produites sur les Scrum Burndown Chart ainsi que la liste des défauts. Les responsabilités du management sont de (1) fournir une vision stratégique, la stratégie métier, et des ressources, (2) supprimer les obstacles remontés par les équipes Scrum et que les équipes ne peuvent supprimer elles-mêmes, (3) assurer la croissance et la carrière des individus, et (4) motiver les équipes Scrum pour sortir de la médiocrité et se dépasser.

3.1.11 Établir et maintenir la description des Méthodes Agiles (GP 3.1)

Cette pratique est un raffinement de la pratique GP2.2 décrite ci-dessus. La seule véritable différence est que la description des Méthodes Agiles dans cette pratique devrait être établie à l'échelle de l'organisation et non pas pour un seul projet. Le résultat est que les variations d'application des Méthodes Agiles sont réduites à l'échelle de l'organisation ; par conséquent, davantage d'échanges entre les personnes des projets, les outils, les informations et les produits peuvent être assurés.

3.1.12 Recueillir les résultats de l'utilisation des Méthodes Agiles pour préparer une utilisation future et améliorer l'approche de l'organisation vis-à-vis des Méthodes Agiles (GP 3.2).

Cette pratique a pour objectif l'apprentissage à travers tous les projets par la collecte des résultats de chacun des projets. En Scrum, la réunion de rétrospective du sprint pourrait être utilisée dans le cadre de cette pratique.

Toutes ces pratiques génériques ont été utiles dans des organisations mettant en œuvre d'autres processus. Nous avons vu qu'un certain nombre de ces pratiques génériques sont au moins partiellement traitées par Scrum ou d'autres Méthodes Agiles. Nous pensons que la mise en œuvre de ces pratiques peut aider à établir la discipline nécessaire à toute Méthode Agile.

4 Conclusion

L'utilisation combinée de CMMI et Scrum donne des résultats significatifs en améliorant considérablement les performances tout en maintenant le niveau de conformité CMMI. Les projets pilotes Scrum ont remonté des gains significatifs de productivité et de qualité supérieurs aux méthodes traditionnelles. Ces résultats permettent de prendre une décision basée sur le retour sur investissement pour introduire plus largement Scrum et envisager l'usage des autres pratiques Agiles chez Systemic. Scrum diminue aujourd'hui de 50% la part de travail (traitement des défauts, reprise de code, charge totale et pilotage).

Pour les entreprises Agiles, nous avons décrit comment les Pratiques Génériques de CMMI pouvaient être utilisés pour institutionnaliser les pratiques Agiles et nous avons présenté le principe de développement logiciel en mode Lean comme un outil opérationnel permettant d'identifier des opportunités d'amélioration dans une entreprise CMMI 5.

Les entreprises dans la défense, l'aérospatiale et d'autres industries qui requièrent un haut degré de maturité des processus, devraient soigneusement envisager l'introduction des pratiques Agiles et toutes les entreprises logicielles devraient envisager l'introduction de pratiques CMMI.

Notre recommandation à la communauté Agile est d'utiliser les pratiques génériques du Niveau 3 de CMMI pour amplifier les avantages de l'usage des Méthodes Agiles. Notre recommandation à la communauté CMMI est que les Méthodes Agiles peuvent s'insérer dans votre référentiel CMMI et offrir de passionnants axes d'améliorations à votre organisation.

5 Références

- [1] Krasner and Houston, "Using the Cost of Quality Approach for Software," *CrossTalk*, November 1998.
- [2] M. Diaz and J. King, "How CMM Impacts Quality, Productivity, Rework, and the Bottom Line," *CrossTalk*, March 2002.
- [3] M. Poppendieck and T. Poppendieck, *Lean Software Development: An Implementation Guide*: Addison-Wesley, 2006.
- [4] M. K. Kulpa and K. A. Johnson, *Interpreting the CMMI: A Process Improvement Approach*. Boca Raton: Auerbach Publications, 2003.
- [5] M. B. Chrissis, Konrad, and Shrum, *CMMI – guideline for process integration and product improvement*, 2002.
- [6] J. Sutherland, "Future of Scrum: Parallel Pipelining of Sprints in Complex Projects," in *AGILE 2005 Conference*, Denver, CO, 2005.