



La concordance des pratiques agiles au *Capability Maturity Model Integration*  
(CMMI)

par

Michel Céré

essai présenté au Département d'informatique  
en vue de l'obtention du grade de maître ès sciences  
(M. Sc. en génie logiciel incluant un cheminement de type cours)

FACULTÉ DES SCIENCES  
UNIVERSITÉ DE SHERBROOKE

Longueuil, Québec, Canada, juin 2008

## Sommaire

Une entreprise peut-elle être à la fois agile et se conformer à un modèle de maturité tel que le *Capability Maturity Model Integration* (CMMI)? Une forte perception au sein de l'industrie du développement logiciel semble en douter sans pour autant être en mesure de vraiment appuyer cette prétention.

Cet essai vise à limiter la subjectivité apparente à l'égard de cette perception en proposant une méthode établissant une matrice de concordance entre des pratiques dites agiles et le modèle de maturité CMMI. Les pratiques observées sont celles généralement associées à la méthode Scrum et, dans le but de circonscrire la portée de l'essai, le domaine de processus CMMI étudié est *Project Planning*.

L'analyse des pratiques de la méthode Scrum et des objectifs du domaine de processus CMMI *Project Planning* permet d'identifier d'une part les artefacts que produisent les pratiques et d'autre part des éléments qui composent les objectifs du domaine de processus étudié soit les buts Établir les estimations, Développer un plan de projet et Obtenir l'engagement envers le plan.

La matrice de concordance présente sur un axe les éléments des objectifs CMMI et sur l'autre les pratiques de la méthode Scrum. Les rencontres de ces deux axes sont qualifiées en fonction de la preuve de l'atteinte des éléments des objectifs CMMI.

Tel que représenté dans cet essai, l'analyse permet d'affirmer qu'il existe bel et bien une concordance entre certaines pratiques agiles et les objectifs CMMI.

Il est également possible de cet exercice d'analyse des apprentissages utiles, notamment qu'il faut être prudent de ne pas interpréter sommairement la matrice au risque d'en tirer de fausses conclusions.

Enfin, la méthode proposée pour réaliser la matrice de concordance est une proposition qui doit être améliorée empiriquement soit par une meilleure précision des éléments composants les objectifs CMMI soit par une meilleure qualification de la rencontre des pratiques avec ces éléments.

## **Remerciements**

Je tiens à remercier chaleureusement les personnes suivantes sans qui l'aboutissement de cet essai n'aurait jamais eu lieu. D'abord mon très dévoué directeur académique Patrice Roy pour son soutien constant, ses conseils, son souci du détail, sa collaboration et pour sa générosité de son temps. Également, je souligne l'apport de mon directeur professionnel Bernard Baraby qui m'a permis de mieux orienter la rédaction de cet essai à ses débuts. Enfin, je remercie tout particulièrement ma conjointe, Julie Fortin pour ses conseils, son temps, sa patience, ses encouragements et son soutien indéfectible.

## Table des matières

Introduction.....	1
Chapitre 1 – Modèle de maturité CMMI .....	3
1.1    Historique du modèle.....	3
1.2    Présentation du modèle CMMI.....	5
Chapitre 2 – Méthodes agiles.....	9
2.1    Historique des méthodes de développement agile.....	9
2.2    Terminologie propre aux méthodes agiles.....	12
2.3    Présentation de la méthode Scrum.....	15
Chapitre 3 – Concordances des pratiques .....	17
3.1    Méthode utilisée.....	18
3.2    Méthode d’analyse d’objectifs CMMI.....	19
3.3    Méthode d’analyse des pratiques de la méthode Scrum .....	23
3.4    Analyse d’objectifs du CMMI .....	25
3.4.1    Établir les estimations.....	25
3.4.2    Développer un plan de projet.....	28
3.4.3    Obtenir l’engagement envers le plan .....	31
3.5    Pratiques de la méthode Scrum.....	34
3.5.1    Sprint.....	34
3.5.2    Planification et organisation d’avant-match .....	38
3.5.3    Rencontre de planification du Sprint .....	41
3.5.4    Liste des fonctionnalités du produit.....	42
3.5.5    Équipe Scrum.....	44
3.5.6    Rencontre quotidienne .....	47
3.5.7    Revue du Sprint .....	49

Chapitre 4 – Matrice de concordance .....	52
4.1    Qualification des croisements .....	52
4.2    Réalisation d’une matrice de concordance .....	54
4.3    Constats et apprentissage sur les méthodes d’analyse utilisées .....	57
4.4    Constats et apprentissage sur la matrice .....	60
Conclusion .....	62
Liste des références.....	64
Annexe 1 – Bibliographie .....	68
Annexe 2 – Composante du modèle CMMI .....	71
Annexe 4 – Manifesto for Agile Software Development .....	75
Annexe 5 – Scrum et CMM.....	76

## Liste des tableaux

Tableau 3.1	Questions pour l'analyse des objectifs selon la SQA ([17]). .....	22
Tableau 3.2	Questions pour l'analyse des pratiques selon la SQA ([17]). .....	24
Tableau 4.1	Matrice de concordance CMMI / Scrum.....	55



## Lexique

<i>Backlog</i>	Liste d'activités ou de tâches à réaliser. (Traduction libre)
Buts	Mot réservé pour nommer explicitement l'un des objectifs du CMMI [22].
Client	Personnes ou organisation à qui le projet est destiné. Il peut s'agir des utilisateurs visés par le produit, de la personne ou de l'organisation qui finance le projet ou encore de ceux qui supervisent la réalisation du projet.
Contrainte	Aux fins de cet essai, ce terme représente notamment les risques, problèmes, décisions à prendre, clarifications, <i>etc.</i>
<i>Levels of Concern</i>	L'une des trois perspectives selon lesquelles une méthode ou norme peut être évaluée : « <i>This factor identifies the organizational scope for which the method provides specific guidance</i> » ([4], p. 167) <sup>1</sup> .
<i>Life Cycle Coverage</i>	L'une des trois perspectives selon lesquelles une méthode ou norme peut être évaluée : « <i>This factor identifies the life cycle activities for which the method provides specific guidance</i> » ([4], p. 167) <sup>2</sup>
Mesure	« [...] <i>is the number or symbol assigned to an entity by this mapping in order to characterize an attribute</i> ». ([13], p. 28) <sup>3</sup>
Mesurer	« <i>Measurement is a process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to describe them according to clearly defined rules</i> » ([13], p. 5) <sup>4</sup>
Objectifs	Mot utilisé pour parler des buts du CMMI sans en spécifier un en particulier [22].

---

<sup>1</sup> Niveaux de préoccupation. Mesure qui permet d'identifier la portée d'une organisation et pour laquelle une méthode donne une direction spécifique. (Traduction libre)

<sup>2</sup> Couverture du cycle de vie. Mesure qui permet d'identifier les activités du cycle de vie et pour laquelle une méthode donne une direction spécifique. (Traduction libre)

<sup>3</sup> Une mesure est le nombre ou le symbole utilisé pour caractériser un attribut. (Traduction libre)

<sup>4</sup> Mesurer est le processus par lequel des nombres ou des symboles sont assignés aux attributs d'entités dans le monde réel de façon à les décrire selon des règles définies clairement. (Traduction libre)

Partie prenante	« Partie considérée comme en lien avec une organisation dans l'accomplissement de ses responsabilités, à laquelle on présente l'information pertinente relative à ses opérations ou à ses activités » [14].
<i>Scrum Master</i>	Le gestionnaire responsable de l'équipe et responsable de mettre à jour la liste des fonctionnalités du Sprint ainsi que toutes les données qui y sont reliées. Il doit également veiller à ce qu'il y ait une mise à jour quotidienne de l'information.
<i>Sources of Constraints</i>	L'une des trois perspectives selon lesquelles une méthode ou norme peut être évaluée : « <i>Each method puts constraints on the implementor – the fewer the constraints, the more agile the method can be considered</i> ». ([4], p. 167) <sup>5</sup>
<i>Sprint</i>	Expression empruntée au domaine sportif définissant une période de temps pendant laquelle une équipe effectue des tâches et qui correspond à environ 30 jours dans la méthode Scrum. ([20], p. 50)
<i>Velocity</i>	Vitesse. (Traduction libre) La vitesse est une mesure qui aide à identifier la capacité d'une équipe à réaliser une quantité de travail dans un temps donné ([8], p. 20)

---

<sup>5</sup> Sources des contraintes. Chaque méthode impose des contraintes à ceux qui l'utilisent. Moins il y a de contraintes et plus la méthode peut être considérée agile. (Traduction libre)

## Liste des sigles et acronymes

CMM	<i>Capability Maturity Model</i> : modèle de développement logiciel d'évolution des capacités (Traduction libre). Acronyme également utilisé pour désigner SW-CMM.
CMMI	<i>Capability Maturity Model Integration</i> : modèle intégré de développement logiciel d'évolution des capacités. (Traduction libre)
DoD	<i>US Department of Defense</i> : Département de la défense du gouvernement américain. (Traduction libre)
GDT	Grand dictionnaire terminologique ([14]).
GQM	<i>Goal Question Metric</i> : métrique de questions orientées objectifs. (Traduction libre)
GQ(I)M	<i>Goal Question Indicator Metric</i> : métrique d'indicateur de questions orientées objectifs. (Traduction libre)
P-CMM	<i>People Capability Maturity Model</i> : modèle de développement d'évolution des capacités des personnes. (Traduction libre)

SA-CMM	<i>Software Acquisition Capability Maturity Model</i> : modèle d'évolution des capacités d'acquisition de solution logicielle. (Traduction libre)
SCAMPI	<i>Standard CMMI Appraisal Method for Process Improvement</i> : méthode d'évaluation du CMMI pour l'amélioration des processus. (Traduction libre)
SE-CMM	<i>Systems Engineering Capability Maturity Model</i> : modèle de développement d'évolution des capacités de système d'ingénierie. (Traduction libre)
SPIN	<i>Software Process Improvement Network</i> : Réseau pour l'amélioration de processus logiciel. (Traduction libre)
SQA	<i>Smart Questions Approach</i> : approche par questions intelligentes. (Traduction libre)
SW-CMM	<i>Capability Maturity Model for Software</i> : modèle de développement logiciel d'évolution des capacités. (Traduction libre)

## Introduction

Il existe une forte perception au sein de l'industrie du développement logiciel voulant qu'une entreprise ou une équipe de développement qui doit se conformer à un modèle de maturité ne peut en même temps être agile ([4], p. 50).

L'hypothèse de cet essai est que cette perception est en grande partie erronée. D'une part, car la plupart des pratiques dites agiles n'entrent pas en contradiction ou en conflit avec les objectifs véhiculés par un modèle de maturité tel que le *Capability Maturity Model Integration* (CMMI). D'autre part, avant de déterminer si une pratique est discordante d'un objectif, il faut à tout le moins procéder à une analyse rigoureuse qui limite la subjectivité apparente.

Cet essai vise à déterminer s'il est possible d'établir une concordance entre des objectifs d'un modèle de maturité et des pratiques d'une méthode dite agile. Pour procéder à cet exercice de concordance une méthode d'analyse est proposée pour le domaine de processus CMMI *Project Planning*<sup>1</sup> et certaines pratiques de la méthode Scrum. Le domaine de processus et de la méthode agile sont choisis arbitrairement, notamment en raison de l'accessibilité de l'information et de la connaissance disponible pour réaliser l'analyse proposée.

---

<sup>1</sup> Planification de projet. (Traduction libre)

En effet, l'analyse des objectifs CMMI et des pratiques de la méthode Scrum s'appuie sur de nombreuses monographies rédigées par des experts de l'industrie. La méthode proposée s'appuie quant à elle sur une réflexion faite lors de la lecture du livre de Nadler et Chandon intitulé *Smart Questions* [17].

Afin de déterminer la faisabilité d'établir une telle concordance, un bref historique du CMMI et des méthodes agiles est d'abord présenté. Ensuite, la méthode utilisée pour effectuer l'analyse des objectifs du domaine de processus CMMI *Project Planning* et des pratiques de la méthode Scrum est décrite, préalablement à l'analyse elle-même. Puis, les résultats de l'application de la méthode d'analyse aux objectifs CMMI et aux pratiques de la méthode Scrum sont exposés en utilisant une matrice qui établit un indice de concordance global pour chacun de ces objectifs. Enfin, une réflexion sur les constats et les apprentissages perçus lors de l'analyse et de l'utilisation de la méthode sont également offerts.

Dans un contexte où les entreprises peuvent avoir à être auditées selon différents modèles, normes ou règles, une matrice qui permet d'établir une concordance entre leurs propres pratiques et les objectifs du CMMI peut être particulièrement utile. D'autant plus que la façon dont cette matrice est établie permet à l'entreprise de mieux se questionner sur ses propres objectifs et peut servir à la remise en question de pratiques déjà en place, mais dont les inconvénients ou bénéfices sont inconnus.

# Chapitre 1

## Modèle de maturité CMMI

Tel que son nom l'indique, le *Capability Maturity Model Integration* est un modèle et non une norme ou un standard en tant que tel. Richard Basque indique que ce modèle « constitue un compendium de bonnes pratiques à appliquer dans tout projet qui veut livrer un produit à temps, dans les budgets, à la satisfaction du client et avec une rentabilité intéressante pour le développeur ([3] p. 10). Si le CMMI est un modèle, cela implique qu'il est possible de s'inspirer de celui-ci pour en créer un autre, de l'adapter ou encore de le réduire selon les besoins et les circonstances. Pour ce faire, il convient de bien comprendre la structure du CMMI et ses objectifs (Annexe 2, [23] p. 17).

### 1.1 Historique du modèle

À la fin des années 70, le *US Department of Defense* (DoD) procède à une vaste étude sur les dépenses en sous-traitance de développement logiciel. Donnant suite à cette étude en 1984, le DoD crée le *Software Engineering Institute* (SEI) et confie son hébergement à l'université Carnegie Mellon.

D'abord décrite dans le livre *Managing the Software Process* de Watts Humphrey en 1989, une première version du modèle CMMI est publiée officiellement par le SEI sous

l'acronyme SW-CMM pour *Capability Maturity Model for Software*. Le succès du SW-CMM fut tel qu'il engendre un engouement pour la création d'autres modèles tel que le *Systems Engineering Capability Maturity Model (SE-CMM)*, *Software Acquisition Capability Maturity Model (SA-CMM)*, *People Capability Maturity Model (P-CMM)*.

Une deuxième version du SW-CMM est mise en chantier en 1995. Cependant, en 1998, le besoin d'intégrer ensemble les différents modèles sous la bannière du CMMI se fait sentir. C'est donc en 2000 que la première version intégrée du modèle CMMI voit le jour. Enfin, une première mise à jour apparaît en 2003 et une deuxième en 2006.

À l'origine, le SEI est créé pour répondre à la crise du logiciel qui sévit depuis le milieu des années soixante. Cette crise est caractérisée notamment par le fait que les logiciels ne sont pas livrés à temps, qu'ils coûtent plus cher que prévu, qu'ils ne répondent pas aux besoins des clients ou qu'ils ne voient tout simplement pas le jour. C'est dans ce contexte que le SEI doit fournir au DoD une grille d'évaluation des fournisseurs de services informatiques.

Le CMMI est évidemment bien plus qu'une grille d'évaluation. C'est d'abord un modèle d'amélioration continue dont les différentes étapes ou échelles servent de point de repère et de guide. De plus en plus, les gouvernements américain et canadien utilisent la grille du CMMI dans l'attribution de contrats de services informatiques. Ainsi, les entreprises qui désirent soumettre leur candidature doivent démontrer qu'elles ont un niveau suffisant de



maturité. Le niveau de maturité d'une entreprise se définit comme étant l'étape à laquelle elle se situe dans un processus d'amélioration continu de ses processus. La connaissance du niveau de maturité d'une organisation permet de prédire les performances futures de celle-ci quant à une tâche donnée ([22] p.10).

Or, l'étude même d'une entreprise n'est pertinente que dans un contexte et à un moment donné; elle ne vaut pas pour l'avenir. Elle sert d'indicateur pour un projet, une équipe, un département ou une entreprise donnée. Ainsi, une entreprise peut soumettre une équipe à une évaluation, obtenir le niveau de maturité souhaité et se déclarer satisfaite. Son objectif n'est donc plus l'amélioration continue, mais l'obtention d'un niveau quelconque de maturité pour obtenir des contrats.

Lorsque l'objectif n'est plus l'amélioration mais la conformité, une entreprise ne peut que viser la mise en place de processus qui visent à satisfaire un éventuel auditeur plutôt que de répondre à ses propres besoins d'affaire ou de développement.

## **1.2 Présentation du modèle CMMI**

Tel que vu dans la section précédente, l'origine du CMMI, ses motivations ou encore son utilisation peuvent faire en sorte de s'éloigner de ses objectifs. C'est pourquoi il importe de mettre l'accent non pas sur les pratiques énumérées par le CMMI, mais bien sur les

objectifs qu'il vise à accomplir. Une brève étude de la terminologie propre au modèle CMMI, notamment la maturité, les niveaux d'aptitude, le domaine de processus, les buts et les meilleures pratiques aident à mieux comprendre ces objectifs.

Le modèle permet d'établir la maturité selon une perspective étagée ou continue. Selon la perspective utilisée, la maturité d'une entreprise est établie selon différents niveaux d'aptitude. Ainsi, l'amélioration continue permet de mesurer le niveau d'aptitude par processus, ce qui est particulièrement utile s'il faut évaluer uniquement certains processus. Les niveaux d'aptitude s'échelonnent de zéro à cinq, alors que la perspective étagée utilise une échelle d'un à cinq s'il faut mesurer un niveau d'aptitude de façon plus globale.

Cette différence de niveau entre les deux perspectives s'explique par le fait que l'amélioration continue est « plus ciblée et ne porte que sur un processus à la fois, la représentation continue doit pouvoir exprimer la situation où la réalisation des activités de base d'un processus n'est tout simplement pas constatée » ([3] p. 34).

Peu importe la perspective utilisée, le modèle est structuré par domaine de processus. Par exemple, la version 1.1, présente un total de 25 domaines de processus ([22]). Chacun de ces domaines contient des objectifs qui peuvent être génériques à tous les domaines ou leur

être spécifiques. Chaque objectif<sup>2</sup> est composé d'un ensemble de pratiques qui peuvent générer des extrants et des sous-pratiques.

Les meilleures pratiques ont été choisies par un panel d'experts et elles sont reconnues comme étant représentatives des meilleures pratiques de l'industrie au moment de la rédaction du texte. De nouvelles pratiques peuvent ne pas être reconnues à titre de « meilleures pratiques » au sens du modèle. Elles peuvent néanmoins être tout aussi valables selon le contexte.

Dans le cadre de cet essai, la matrice de concordance du modèle se limite uniquement au domaine de processus « Planification de projet »<sup>3</sup>. Richard Basque définit ce domaine comme suit : « L'intention du domaine de processus Planification de projet est d'établir et maintenir les plans qui définissent les activités de projet » ([3], p. 64). Le domaine de processus Planification de projet de la catégorie Gestion de projet (*Project Management*) est constitué de cinq objectifs, soit (a) *Establish Estimates*, (b) *Develop a Project Plan*, (c) *Obtain Commitment to the Plan*, (d) *Institutionize a Managed Process* et (e) *Institutionalize a Defined Process* dont la traduction s'inspire librement de celle de Richard Basque par (a)

---

<sup>2</sup> Les mots *goal* et *objectives*, traduits librement par but et objectifs, sont des synonymes dans le texte comme dans le CMMI ([22], p. 26). Lorsqu'un objectif est nommé expressément le mot « but » est alors utilisé pour le désigner.

<sup>3</sup> À titre de convention dans le texte, les objectifs, éléments ou pratiques sont entre guillemets lorsqu'elles sont nommées explicitement pour une première fois dans le texte. Par la suite, elles ne conservent que la majuscule.

« Établir les estimations », (b) « Développer un plan de projet », (c) « Obtenir l'engagement envers le plan », (d) « Institutionnaliser un processus géré » et (e) « Institutionnaliser un processus défini » [3].

Le domaine de processus Planification de projet est principalement retenu parce qu'à sa face même, il s'agit du domaine qui s'apparente le mieux avec certaines des pratiques de la méthode Scrum.

## Chapitre 2

### Méthodes agiles

Malgré les efforts du SEI pour améliorer les pratiques de l'industrie (avec la publication du CMMI) une problématique n'a pas été résolue. En effet, il semble y avoir l'apparence d'un formalisme lourd qui émerge des entreprises ayant adopté le modèle : « *If one has strong discipline without agility, the result is bureaucracy and stagnation* » ([4], p. 2). De nouvelles méthodes de travail émergent et font contrepoids à cette tendance. Celles-ci sont regroupées sous les méthodes agiles. Il apparaît également évident que les pratiques et méthodes dites traditionnelles sont remises en question par l'essence même des méthodes agiles, soit le développement itératif et incrémental.

#### 2.1 Historique des méthodes de développement agile

Malgré une apparente évolution des pratiques de développement logiciel et d'une prise de conscience du besoin d'améliorer la maturité des organisations, un triste constat émerge en 1994 lors de la publication du *Chaos Report* par *The Standish Group* [28]. L'étude du *Standish Group* révèle que les projets complétés à temps, conformes au budget et aux devis ne représentent que 16.2 % de l'ensemble des cas étudiés alors que les projets qui ont été complétés, mais en retard, dont les coûts ont été dépassés et dont les fonctionnalités n'ont

pas toutes été livrées en représentent 52.7 %. Les autres projets représentent 31.1 % des projets qui ont été annulés à un moment donné durant le cycle de développement.

Selon Craig Larman, les méthodes utilisées, dites traditionnelles, sont au cœur des problèmes rencontrés dans le développement logiciel ([16], p. 74). Malgré tout, elles continuent d'être utilisées encore même aujourd'hui, en particulier le *Waterfall* ([16], p. 103).

Une partie de cette problématique vient du fait que plusieurs normes ou standards s'inspirent du standard DOD-STD-2167. Ce standard a d'abord été créé par le DoD et a servi d'inspiration aux autres départements de ce même gouvernement, ainsi qu'ailleurs dans le monde. Cependant, lorsque le DoD met à jour son standard en 1994, notamment pour y introduire le développement itératif et incrémental, les autres ne suivent pas en omettant de mettre leurs standards à jour. Également, Craig Larman soutient que peu de personnes ont lu l'article à l'origine du *Waterfall* et qu'ils ignorent que Royce est en fait un promoteur du développement itératif et incrémental : « *If the computer program in question is being developed for the first time, arrange matters so that the version finally delivered to the customer for operational deployment is actually the second version* »<sup>4</sup> ([18], p. 334).

---

<sup>4</sup> Si le programme en question est développé pour une première fois, faites en sorte que la version finale livrée au client pour le déploiement opérationnel soit en fait la seconde version. (Traduction libre)

Puisque cette méthode est facile à expliquer et à intégrer, Larman suggère qu'elle a beaucoup plus de succès qu'un développement itératif et incrémental qui est plus complexe à comprendre et à décrire. Par exemple, les projets de développement logiciels sont souvent comparés à tort au paradigme de projets de développement manufacturiers prévisibles. Larman suggère même que le *Waterfall* est favorisé par certains gestionnaires puisqu'il donne l'illusion d'un processus prévisible, en ordre, mesurable et imputable qui possède des jalons orientés vers la documentation. Larman soutient également que plusieurs certificateurs CMM sont influencés par les valeurs du *Waterfall*. ([16], pp. 105-106).

En 2001, des personnalités influentes (Annexe 3) du développement logiciel se rencontrent pour discuter et émettre un manifeste des méthodes agiles (Annexe 4). Plusieurs méthodes ont émergé ou se sont réunies autour de ce manifeste. Les éléments contenus dans le manifeste sont plus amplement abordés à la prochaine section. Par la suite, les méthodes de développement agile ont connu une plus grande notoriété, d'abord par la publication de nombreuses monographies, et ensuite par une très grande visibilité lors de conférences ou séminaires. Des organisations tel que *Agile Alliance* ont même été créées pour promouvoir des méthodes agiles. Enfin, plusieurs groupes de discussions existent dans Internet vantant les succès et les méthodes utilisées. Par exemple, à la Standard Life du Canada, l'expérience a démontré qu'en situation critique la méthode peut contribuer au succès du projet, tel que le rapporte Johanne Danis-Préfontaine lors d'une conférence au SPIN qui s'est tenue à Montréal le 5 novembre 2006 [25].

## 2.2 Terminologie propre aux méthodes agiles

Contrairement au CMMI, les méthodes agiles ne proposent pas de modèles en particulier; ils énoncent cependant un ensemble de principes, de valeurs et de pratiques qui guident la réalisation d'un développement logiciel. Alors que le CMMI est une approche normative, en ce sens où une solution générique est appliquée à un problème donné, les méthodes agiles se basent sur une approche empirique où, selon l'expérience et les résultats obtenus, les processus sont constamment en amélioration. Dans la deuxième édition de son livre *Agile Software Development*, Alistair Cockburn, constate qu'il y a deux distinctions fondamentales entre les méthodes agiles et CMMI : d'une part, que le « *CMMI is based on a statistical process assumption rejected by the authors of the agile manifesto* »<sup>5</sup>, et d'autre part que « *The CMMI places optimisation of the process at the final level, level 5. Agile teams (teams using Crystal, in particular) start optimising the process immediately* »<sup>6</sup> ([7], p. 2).

En effet, l'optimisation du processus peut se réaliser par une approche de développement itératif et incrémental d'un projet. Par exemple, le développement d'une application qui doit prendre 12 mois à réaliser peut être constitué de 12 petites itérations d'un mois au

---

<sup>5</sup> Le CMMI est basé sur des présomptions statistiques des processus qui ont été rejetées par les auteurs du manifeste agile. (Traduction libre)

<sup>6</sup> Le CMMI place l'optimisation des processus au dernier niveau, le niveau 5. Les équipes agiles (en particulier celle qui utilise Crystal) commencent à optimiser leurs processus immédiatement. (Traduction libre)



cours desquelles certaines actions sont portées. Chacune de ces itérations est constituée d'une part de construction d'éléments d'un logiciel utile et fonctionnel, puis d'une remise en question des méthodes de fonctionnement selon les résultats obtenus. Ainsi, l'amélioration se fait au niveau de l'équipe de façon continue tout au long du projet, contrairement au CMMI qui vise l'amélioration au niveau institutionnel.

Les principes agiles s'orchestrent autour de quatre thèmes principaux : *Individuals and interaction over processes and tools*, *Working software over comprehensive documentation*, *Customer collaboration over contract negotiation* et *Responding to change over following a plan* ([1]).

Le premier de ces thèmes est traduit librement par : les individus et les interactions doivent primer sur les procédures et les outils. Il est conçu selon la prémisse qui veut que la programmation soit d'abord et avant tout une activité humaine de création ([5], p. 202). Ainsi, tout processus de développement doit tendre vers l'amélioration de l'organisation des humains, de la communication et des façons de travailler.

Le deuxième thème est traduit librement par : le développement logiciel doit primer sur la documentation exhaustive. Sans nier qu'une documentation est parfois nécessaire et même obligatoire pour certains organismes réglementaires (par exemple un design ou des spécifications), il est trop souvent possible de constater que l'ensemble de la documentation

rédigée dans le cadre d'un projet est soit inutile, soit inutilisable. Ceci peut parfois s'expliquer par le fait que l'information ainsi documentée peut être éphémère et n'a pas besoin d'être mise à jour tout au long du projet. Également, la documentation n'est rédigée qu'au début d'un projet, mais pas mise à jour, la rendant désuète et par le fait même inutilisable en cours de projet. Bref, le principe énonce tout simplement qu'il est parfois préférable d'avoir un logiciel fonctionnel plutôt que de miser sur de la documentation qui demande beaucoup d'énergie et de temps à construire et à maintenir. Alistair Cockburn indique que s'il faut absolument documenter la réalisation d'un développement logiciel, il faut en faire juste assez et pas plus ([6], p. 176).

Le troisième thème est traduit librement par : la collaboration avec le client doit primer sur la négociation contractuelle. Ceci implique qu'il doit y avoir une forme ou une autre de rétroaction entre l'équipe qui réalise le produit et le client. En effet, sans cette collaboration, les développeurs ne peuvent être en mesure de bien mesurer les attentes et de s'assurer que l'accent est mis sur des fonctionnalités utiles et ayant une valeur certaine pour le client.

Enfin, le quatrième thème est traduit librement par : l'ouverture au changement doit primer sur le suivi d'un plan rigide. Il se veut un reflet d'une réalité selon laquelle environ 25 % des besoins changent au cours de la réalisation d'un projet en développement logiciel ([16], p. 51). En effet, si les besoins ou requis d'un projet changent de façon aussi importante,

vaut mieux mettre en place une structure qui permet d'absorber ces changements plutôt que suivre un plan qui, en fin de compte, ne répond plus aux exigences d'un client.

### 2.3 Présentation de la méthode Scrum

Plusieurs méthodes agiles, dont Scrum, évoluent autour de quatre principes énoncés dans le manifeste agile. Dans leur livre sur la méthode Scrum, Ken Schwaber et Mike Beedle expliquent ainsi la méthode : « *First used to describe hyper-productive development in 1987 by Ikujiro Nonaka and Hirotaka Takeuchi, Scrum refers to the strategy used in rugby for getting an out-of-play ball back into play* » ([20], p. 1).

La liste des pratiques de la méthode Scrum provient essentiellement des auteurs de la méthode Ken Schwaber et Mike Beedle. Ceux-ci identifient six pratiques distinctes, soit : « Équipe Scrum » (*Scrum Teams*), « Liste des fonctionnalités du produit » (*Product Backlog*), « Rencontre quotidienne » (*Daily Scrum Meeting*), « Rencontre de planification du Sprint » (*Sprint Planning Meeting*), « Sprint » et « Revue du Sprint » (*Sprint Review*). De son côté, Craig Larman ([16], pp. 117-120) fait également un inventaire assez intéressant des pratiques de la méthode Scrum et ajoute à celles mentionnées par Schwaber les pratiques : « Planification et organisation d'avant-match » (*Pre-game Planning and Staging*), « Équipe autodirigée et auto-organisée » (*Self-Directed and Self-Organizing Teams*), « Aucun ajout pendant l'itération » (*Don't Add to Iteration*), « Pare-feu » (*Scrum*

*master firewall*), « Décision en une heure » (*Decision in One Hour*), « Éléments bloquants éliminés en un jour » (*Blocks Gone in One Day*), « Poulets et cochons » (*Chickens and Pigs*), « Équipe de sept » (*Teams of Seven*), « Salle commune » (*Common Room (preferred)*) et « Compilation quotidienne » (*Daily Build*).

L'un des objectifs des méthodes agiles est d'améliorer la réalisation d'un projet et non pas l'organisation au grand complet tel que peut le prétendre le CMMI. Le défi d'utiliser ces méthodes dans une organisation CMMI est de démontrer que les objectifs des deux méthodes ne sont pas contradictoires, mais qu'ils sont complémentaires. Évidemment, la question est de déterminer comment il est possible de démontrer que les méthodes agiles, en l'occurrence certaines pratiques de la méthode Scrum, sont conformes à l'esprit des objectifs du CMMI.

## Chapitre 3

### Concordances des pratiques

Les auteurs Barry Boehm et Richard Turner ([4], p. 166) proposent de mesurer certaines méthodes, normes et modèles selon trois perspectives soit les niveaux d'implication (*Levels of Concern*), la couverture du cycle de vie (*Life-Cycle Coverage*) et les contraintes (*Sources of Constraints*). Cet exercice compare entre elles les différentes méthodes seulement, et ne cherche pas à vérifier le niveau de conformité d'une méthode par rapport à une norme ou à un modèle. Cette comparaison est utile tant aux praticiens et aux spécialistes des méthodes qu'aux auditeurs, puisqu'elle permet notamment d'adapter une méthode à un contexte tout en demeurant conforme à un certain niveau de maturité.

Cet essai propose non pas d'établir une concordance complète, mais plutôt une méthode pour la concevoir en prenant en exemple le modèle CMMI portant sur le développement logiciel et la méthode Scrum. Ainsi, la concordance proposée vise d'une part à déterminer si les pratiques proposées par une méthode dite agile sont en mesure de se conformer aux exigences du modèle CMMI, et sert d'autre part à qualifier le niveau de satisfaction de ces exigences. Par exemple, une pratique peut répondre à une exigence complètement, partiellement ou pas du tout.

Ken Schwaber établit une concordance de la méthode Scrum avec CMM<sup>7</sup> au niveau des domaines de processus et, hormis l'exemple qu'il donne pour le domaine de processus Gestion des besoins, peu d'explications sont fournies pour justifier l'évaluation qu'il fait de la concordance des domaines de processus avec la méthode Scrum (Annexe 5).

L'inconvénient de cette matrice, c'est de ne pas permettre d'apprécier ou de critiquer l'analyse en raison d'un manque d'information et de justification. C'est en raison de l'absence d'information relative à la façon dont est effectuée la matrice de Schwaber qu'il est essentiel de proposer une méthode visant la mise en place d'une concordance des pratiques avec des objectifs.

### **3.1 Méthode utilisée**

Tel que le suggère Hillel Glazer, chef de la direction de la compagnie Entinex et auteur de nombreux articles sur le CMMI et les méthodes agiles, tenter de remplacer les pratiques proposées par le CMMI par des pratiques agiles peut être utile, mais n'est pas nécessaire ([15]). Ce qui est essentiel selon Glazer est simplement de démontrer comment les pratiques agiles permettent d'obtenir des résultats similaires à ceux promis par le modèle CMMI.

---

<sup>7</sup> Lorsque Ken Schwaber parle de CMM, il ne s'agit probablement pas du CMMI mais du SW-CMM qu'il ne cite qu'avec l'acronyme CMM.

Les auteurs Dennis M. Ahern, Aaron Clouse et Richard Turner abondent également dans le même sens lorsqu'ils affirment que « *The sole required component of the CMMI models is the "goals"* »<sup>8</sup> ([2], p. 61). Il est donc possible d'en déduire que c'est par l'analyse des éléments nécessaires à l'atteinte des objectifs qu'une concordance est possible.

### **3.2 Méthode d'analyse d'objectifs CMMI**

L'analyse des objectifs passe par trois étapes. Tout d'abord, il faut établir la liste des objectifs du modèle CMMI, plus précisément ceux du domaine de processus Planification de projet. Cette étape est particulièrement facile à réaliser puisque tous les objectifs sont clairement identifiés dans le modèle.

En guise de seconde étape, pour chacun de ces objectifs, une analyse sommaire de la terminologie doit être effectuée lorsque les termes sont ambiguës. Au besoin, cette analyse débute d'abord par une recherche de définitions fournies par le modèle CMMI. Lorsque les termes utilisés pour nommer ou décrire les objectifs ne sont pas définis dans le modèle, il faut les interpréter selon la définition généralement reconnue dans la communauté scientifique ou professionnelle, selon une publication ou, en dernier lieu, selon le contexte. Ainsi, l'interprétation d'un terme doit avoir un sens dans son contexte.

---

<sup>8</sup> La seule composante requise des modèles du CMMI est celle des buts. (Traduction libre)

La dernière étape consiste à décrire les éléments qui servent à mesurer les objectifs. Pour déterminer comment il est possible de mesurer un objectif, une adaptation de l'approche métrique de questions orientées objectifs (GQM) ([13]) et de l'approche métrique d'indicateur de questions orientées objectifs (GQ(I)M) est utilisée ([24]). Ainsi, pour chacun des objectifs établis du domaine de processus Planification de projet, il faut d'une part identifier les questions qui permettent de déterminer si l'objectif a été atteint et, d'autre part, déterminer si celles-ci répondent adéquatement aux objectifs.

Les questions qui servent à circonscrire les objectifs et à en vérifier l'atteinte s'inspirent à la fois de ceux du GQM ou du GQ(I)M et, plus particulièrement, des concepts élaborés par Gerald Nadler et William J. Chandon dans le livre *Smart Questions* ([17]). Lorsqu'orienté vers la résolution de problèmes, rien n'interdit une adaptation de l'approche par questions intelligentes (SQA) ([17]) pour aider à l'analyse des objectifs. Cette approche propose d'établir une série de questions rassemblées autour de trois aspects : l'unicité, l'information et le système. Le premier aspect vise à identifier les éléments qui font d'un problème (ou plus précisément, dans le cadre de cet essai, d'un objectif) qu'il est unique, c'est-à-dire en quoi un objectif est différent des autres ou en quoi il se démarque significativement. La réponse aux questions de cet aspect permet d'identifier les éléments qui servent de base à la mesure de l'objectif lors de l'établissement d'une matrice de concordance. Par exemple et tel qu'il est plus amplement exprimé plus loin dans ce chapitre, l'unicité du but Établir les



estimations est le fait de constater ou non la présence d'élément prouvant que des évaluations ont été faites.

Le deuxième aspect, l'information, vise à déterminer quelles sont les données nécessaires à l'atteinte d'un objectif. Pour y parvenir, il faut avoir préalablement trouvé les raisons qui justifient cet objectif, donc qui permettent de comprendre à quoi il sert. Par exemple, il peut être nécessaire pour le but Établir les estimations d'avoir une liste des objectifs du projet à accomplir. Ensuite de déterminer la capacité de production d'une équipe, le temps et les ressources disponibles et requises. Enfin de spécifier comment s'effectue le travail.

Le dernier aspect, le système, vise à s'assurer que l'information est pertinente et permet d'atteindre l'objectif sans nuire aux autres, donc à s'assurer que le système en place protège l'information.

Le tableau 3.1 propose, pour chacun des trois aspects, un ensemble de questions non exhaustives et adaptables qui constitue une analyse des objectifs afin d'établir une matrice de concordance entre les objectifs CMMI et les pratiques de la méthode Scrum.

**Tableau 3.1 Questions pour l'analyse des objectifs selon la SQA ([17]).**

<b>Unicité</b> <ul style="list-style-type: none"><li>- En quoi cet objectif est-il unique par rapport aux autres?</li><li>- En quoi cet objectif est-il plus ou moins important que les autres?</li><li>- Quelles sont les circonstances particulières du projet ou de l'entreprise affectant cet objectif?</li><li>- En quoi cet objectif est-il essentiel?</li></ul>
<b>Information pertinente</b> <ul style="list-style-type: none"><li>- Quelle est l'information essentielle à la réalisation de cet objectif, qui doit la fournir et à qui?</li><li>- À quelle fréquence et pendant combien de temps doit-elle être communiquée?</li><li>- Quelle est la pérennité de cette information?</li><li>- Quel est le résultat engendré par cette pratique?</li></ul>
<b>Système</b> <ul style="list-style-type: none"><li>- Qu'est-ce qui peut mettre en péril la qualité de l'information fournie ou son accès?</li><li>- Qu'est-ce qui peut mettre en péril la réalisation de l'objectif?</li><li>- Comment est mesurée l'atteinte de l'objectif?</li></ul>

Ce tableau n'a pas la prétention d'être complet. L'important ici est que l'analyse d'un objectif soit précédée de l'établissement d'une liste de questions aidant à identifier les éléments les plus pertinents ou même essentiels. Il en est d'ailleurs question dans la prochaine section relative à l'analyse des pratiques de la méthode Scrum.

### 3.3 Méthode d'analyse des pratiques de la méthode Scrum

L'analyse des pratiques de la méthode Scrum s'effectue en trois étapes de la même façon que l'analyse des objectifs du modèle CMMI. D'abord, il importe d'obtenir l'inventaire des pratiques de la méthode choisie, soit Scrum ([20]). Ensuite il faut effectuer une analyse de la terminologie utilisée dans les pratiques. Enfin, pour chacune des pratiques, il convient d'établir la liste des éléments essentiels.

La concordance n'est possible que par une analyse des pratiques, notamment en établissant d'abord une liste de pratiques selon la méthode Scrum. Suit ensuite la tentative de rapprocher la pratique et un objectif de façon préliminaire. C'est d'abord grâce à l'analyse de la syntaxe propre à la pratique, puis en identifiant les caractéristiques clés de celle-ci, que l'analyse débute.

L'établissement de la liste des pratiques de la méthode Scrum est relativement facile à obtenir soit par la documentation disponible ou encore par une énumération *ad hoc* sur le site Web qui en fait la promotion ([10]).

Malheureusement et contrairement au modèle CMMI, il peut exister plusieurs définitions et descriptions d'une même pratique. L'analyse d'une pratique provenant d'une méthode dite agile est donc plus ardue et exige une connaissance plus approfondie, que ce soit par le

biais d'une documentation exhaustive de la littérature ou, préférablement, par l'exercice de ces pratiques dans un contexte professionnel. Pour cet essai, l'analyse s'appuie d'abord sur la description simple et détaillée des pratiques de la méthode Scrum par Craig Larman ([16]), par la lecture des textes des créateurs de cette méthode ([ControlChaos], [21], [20]) et des monographies complémentaires, et enfin par l'expérience de l'auteur de cet essai.

La dernière étape consiste à trouver les éléments essentiels des pratiques de la méthode Scrum pour comprendre le fonctionnement de la pratique en recherchant notamment le ou les objectifs de la pratique, les informations essentielles, les résultats obtenus et finalement les limites rencontrées par cette pratique.

**Tableau 3.2 Questions pour l'analyse des pratiques selon la SQA ([17]).**

<p><b>Unicité</b></p> <ul style="list-style-type: none"> <li>- En quoi cette pratique est-elle différente des autres?</li> <li>- En quoi cette pratique est-elle essentielle à la méthode?</li> <li>- Quelles sont les circonstances du projet ou de l'entreprise affectant cette pratique?</li> <li>- Qu'est-ce que cette pratique vise à accomplir?</li> </ul>
<p><b>Information pertinente</b></p> <ul style="list-style-type: none"> <li>- Quelle information est essentielle à la réalisation de cette pratique, qui doit la fournir et à qui?</li> <li>- À quelle fréquence et pendant combien de temps doit-elle être fournie?</li> <li>- Quelle est la pérennité de cette information?</li> <li>- Quel est le résultat engendré par cette pratique?</li> </ul>
<p><b>Système</b></p> <ul style="list-style-type: none"> <li>- Qu'est-ce qui peut mettre en péril la qualité ou l'accessibilité de l'information fournie?</li> <li>- Qu'est-ce qui peut mettre en péril l'exécution de cette pratique?</li> <li>- Qui doit réaliser ou participer à cette pratique?</li> <li>- Quels sont les éléments négatifs liés au recours à cette pratique?</li> <li>- Quelles sont les limites de cette pratique?</li> </ul>

Le tableau 3.2 propose, pour chacun des trois thèmes énumérés dans la section précédente, un ensemble non exhaustif et adaptable de questions permettant l'analyse d'une pratique afin d'en établir la concordance avec les objectifs du modèle CMMI.

### **3.4 Analyse d'objectifs du CMMI**

Il est proposé dans la section Terminologie propre aux méthodes agiles que les pratiques choisies de la méthode Scrum ne visent pas l'organisation, mais bel et bien le projet. C'est pourquoi les objectifs Institutionnaliser un processus géré et Institutionnaliser un processus défini du domaine de processus Planification de projet qui revêtent un caractère organisationnel ne sont pas analysés. Ainsi, l'analyse porte uniquement sur les objectifs Établir les estimations, Développer un plan de projet et Obtenir l'engagement envers le plan.

#### **3.4.1 Établir les estimations**

Afin d'avoir la meilleure interprétation possible des termes, leurs définitions sont cherchées dans la langue d'origine. Ainsi, c'est la définition du *Webster* qui est utilisée pour définir *Estimate* en tant que « *a general calculation of size, value, etc.; esp., an approximate*

*computation of the probable cost of a piece of work made by a person undertaking to do the work* »<sup>9</sup> ([29]).

Le but Établir les estimations est décrit dans le modèle CMMI de la façon suivante : « *Estimates of project planning parameters are established and maintained* »<sup>10</sup>. Le verbe *établir*, lorsqu'il est conforme à la formulation « *establish and maintain* », doit être lu de façon à ce qu'il « *connotes a meaning beyond the component terms ; it includes documentation and usage* » ([22], p. 22)<sup>11</sup>.

Une estimation, selon l'une des définitions du Grand dictionnaire terminologique (GDT), peut être interprétée comme étant un devis, une évaluation ([14]). Certains gestionnaires peuvent considérer que les estimations sont des évaluations, donc une promesse de rendement, une garantie. D'autres peuvent être plus pragmatiques et les considérer en tant que devis qui peut changer, au fur et à mesure que le contexte de réalisation se précise. Pour les fins de cet essai, le terme « estimation » prend le sens d'une évaluation approximative des travaux à réaliser, que ce soit en temps, en ressources et en coûts.

Après avoir défini les termes, il convient d'analyser l'objectif selon les trois aspects cités plus tôt soit son unicité, l'information pertinente et le système.

---

<sup>9</sup> Évaluation de la valeur, de la quantité, de la qualité ([25]). (Traduction libre du mot estimation)

<sup>10</sup> L'évaluation des paramètres du plan de projet est établie et entretenue. (Traduction libre)

<sup>11</sup> Donne une signification au-delà des termes; cela inclut la documentation et l'utilisation. (Traduction libre)

L'unicité de l'objectif s'explique par le fait qu'il s'agit du seul objectif où il est question de prévision budgétaire, d'approximation du temps et des coûts du travail à effectuer. Si le but Établir les estimations contribue parfois à déterminer la faisabilité d'un projet, il n'est cependant pas plus important que les autres objectifs. En effet, l'effort déployé pour réaliser l'estimation doit être proportionnel au risque associé au développement du projet. Également, cet effort ne doit pas mettre en péril la réalisation du projet, que ce soit en raison de son inexactitude ou parce que sa réalisation nécessite un effort démesuré. Ainsi, l'objectif peut être affecté par la complexité d'un projet, par la nouveauté technique, par l'ajout de nouvelles ressources de même que par l'expertise disponible pour réaliser le projet et pour l'estimer. Enfin, l'objectif est essentiel puisqu'il permet d'établir en partie la faisabilité du projet.

En tant qu'information pertinente, il est essentiel de connaître le domaine d'affaire du projet, les besoins du client et les contraintes technologiques, quelles qu'elles soient. Les ressources disponibles pour la réalisation du projet qu'elles soient matérielles ou humaines, de même que le niveau d'expertise de l'équipe ainsi que toute information pouvant affecter l'objectif, sont autant de données essentielles devant être transmises à l'équipe par les gestionnaires et par les clients, régulièrement et jusqu'à la fin de la réalisation du projet. Établir les estimations apporte donc à l'équipe qui réalise le projet, au client et aux gestionnaires, une idée approximative du délai de réalisation, d'une échelle de coûts, des efforts nécessaires et de l'expertise requise.

Enfin, en ce qui concerne le système, la qualité de l'information fournie est mise en péril par l'incompréhension du contexte ou de l'importance des objectifs du projet. Elle l'est également par l'incompréhension du domaine d'affaires ou des priorités. C'est-à-dire que même si l'information est juste, son utilisation peut être erronée s'il y a mésestime sur les données clés du projet. En fait, la réalisation de l'objectif peut être mise en péril s'il y a incompréhension de ces données. Si, par exemple, le client n'est pas suffisamment disponible, alors l'absence de données justes sur le système peut avoir le même effet qu'une modification des données quant aux ressources disponibles.

L'atteinte de l'objectif se mesure par la présence d'artefacts exposant des estimations sur les éléments clés cités plus tôt. Ces artefacts existent pendant tout le cycle de vie du projet et servent éventuellement d'éléments d'information pour l'amélioration de l'équipe et de l'organisation. Par exemple, une rétrospective peut révéler qu'il y a eu des erreurs d'estimation liées à une méconnaissance d'une technologie. Cette constatation peut inciter l'organisation à réorganiser ses équipes en fonction des projets ultérieurs.

### **3.4.2 Développer un plan de projet**

Dans l'expression Développer un plan de projet, un plan est un ensemble structuré d'objectifs ainsi que des moyens utilisés pour les atteindre [14]. Un projet est une réalisation unique, limitée dans le temps et comportant un ensemble de tâches cohérentes,



utilisant des ressources humaines, matérielles et financières en vue d'atteindre ces objectifs, en fonction de contraintes spécifiées [14]. Mike Cohn définit l'action de planifier ou, tel que le suggère l'objectif, de Développer un plan comme étant une tentative de répondre à la question « qu'est-ce qui doit être construit? » ([9], p. 5). Il ajoute que pour répondre à cette question, les équipes doivent notamment prendre en considération les éléments à développer, les ressources disponibles et le délai à l'intérieur duquel il faut réaliser le tout. Il avertit finalement que cette question ne peut être résolue d'un seul coup et qu'elle doit plutôt l'être de façon itérative et incrémentale.

Après l'analyse des termes, il convient d'établir l'unicité, l'information pertinente ainsi que le système du but Développer un plan de projet. Tel que l'analyse terminologique nous l'indique, cet objectif donne un aperçu général du déroulement du projet et y associe les éléments qui servent à le réaliser. Un plan de projet est essentiel pour être en mesure de fournir une estimation. Il sert de guide pour l'équipe et d'outils de mesure sur ce qui est en cours de réalisation. Il donne une direction approximative et permet d'établir les balises du projet et une orientation pour l'équipe de développement. Sans plan, il est difficile, voire impossible, d'estimer ou d'aller dans une direction de façon cohérente.

La complexité du projet et la volatilité des besoins sont des éléments qui affectent l'objectif. En effet, la maintenance d'un plan dont les éléments sont modifiés fréquemment consomme beaucoup d'énergie, souvent sans que ce ne soit vraiment utile.

La réalisation de l'objectif nécessite de connaître les besoins du client, et plus précisément les éléments qui lui sont critiques et prioritaires. Le client doit donc à chaque changement, ou dès qu'une information pouvant affecter l'objectif est disponible, informer l'équipe, et ce, tout au long du projet.

Un échéancier de réalisation, une liste d'objectifs priorisés, un calendrier de réalisation ainsi qu'une liste des dépendances entre les objectifs et les contraintes et risques du projet résultent de la réalisation de l'objectif. Également, des données relatives à la supervision du projet et à la fréquence des contrôles, de même que la façon dont sont mesurés les projets, doivent être indiquées ainsi que la façon dont sont gérés les changements.

La qualité de l'information peut être mise en péril pour les mêmes motifs que ceux exprimés pour le but Établir les estimations et doit, elle aussi, être divulguée régulièrement ou lorsque survient un changement, et ce, tout au long du projet.

Une planification trop détaillée peut mettre en péril la réalisation de l'objectif. En effet, s'il faut mettre à jour régulièrement un plan détaillé, les efforts à cet effet peuvent démobiliser l'équipe qui réalise le projet tout en ajoutant un risque d'erreur à chaque manipulation.

À l’opposé, l’atteinte progressive du résultat peut être mesurée par une planification réellement utilisée, par les prises de décisions qui en découlent et par la mise à jour régulière de l’information utile.

### **3.4.3 Obtenir l’engagement envers le plan**

Le terme *Commitment* est souvent traduit par « engagement » [14]. L’une des définitions proposées par le Webster définit ainsi le terme: « *a dedication to a long-term course of action* » ([29], p. 281). Puisqu’il n’est précisément pas établi qui s’engage, étant donné que ce n’est précisé nulle part, il est supposé que l’engagement est multiple, c’est-à-dire qu’il s’adresse tant à ceux qui réalisent le plan de projet, qu’à ceux qui attendent les résultats et enfin qu’à ceux qui doivent fournir tous les éléments permettant la réalisation du plan de projet.

Considérant que le plan de projet est basé sur des approximations et des suppositions, il est hasardeux de proposer que l’engagement en vise le respect intégral. Cela enlève également du sens à la possibilité de mettre à jour le plan de projet. Il est peu probable de devoir ou de pouvoir mettre à jour un plan de projet qui doit être respecté à la lettre. Une modification des orientations de l’entreprise, une réaffectation de ressources, un changement dans le plan ou l’arrivée d’information modifiant certaines hypothèses dans l’élaboration du plan peuvent affecter l’engagement envers le plan de projet.

Par conséquent, l'engagement de la part des gestionnaires est l'obligation de fournir les ressources nécessaires à la réalisation du projet et de valider la partie réalisable du plan de projet en participant à sa compréhension et à son élaboration. De la part de l'équipe, l'engagement doit être celui de mettre les efforts nécessaires à l'aboutissement de la partie réalisable du plan de projet. L'engagement peut également se limiter à une petite partie du plan ou à la méthode de travail utilisée, à l'information à recueillir et à fournir, ou enfin aux artefacts à livrer au terme d'une période donnée.

L'objectif d'engagement est unique en ce qui a trait à la recherche d'un accord tacite et explicite entre les différentes parties participant au projet, qu'il s'agisse du client, des gestionnaires ou encore de l'équipe qui réalise le projet.

Cet objectif est essentiel en ce sens qu'il ne peut être réalisé sans l'implication réelle des parties prenantes<sup>12</sup>, des gestionnaires et de l'équipe. Il ne s'agit pas que d'une signature en bas d'un contrat, mais d'un engagement moral de part et d'autre à tout mettre en œuvre pour assurer le succès du projet selon le plan de projet proposé.

Le plan de projet ainsi que les estimations sont des données essentielles pour réaliser l'objectif puisque l'engagement se fait sur la base de ces données. Ainsi, toutes

---

<sup>12</sup> « Partie considérée comme en lien avec une organisation dans l'accomplissement de ses responsabilités, à laquelle on présente l'information pertinente relative à ses opérations ou à ses activités » ([14])

modifications au plan et aux estimations doivent être communiquées à tous les intervenants le plus rapidement possible, et ce, tout au long du projet.

La réalisation de l'objectif d'engagement fait en sorte que les ressources soient mises à la disposition de l'équipe, que l'information circule entre tous les intervenants et, surtout, que les parties prenantes sont en mesure de qualifier l'avancement du projet selon le plan. Évidemment, ce qui met en péril la qualité de l'information fournie ou son accessibilité est constitué de tous les éléments mettant en péril la réalisation des buts Établir les estimations et Développer un plan de projet, exposés précédemment.

Par contre, des éléments tels qu'une modification au contexte de réalisation du projet, qu'une rigidité du plan de projet ou que des attentes irréalistes, sont autant d'aspects qui peuvent mettre en péril l'engagement envers le plan de projet. Par exemple, si les gestionnaires ou les clients considèrent que les estimations ne sont pas des approximations mais bien une garantie de résultat, cela peut, dans un contexte de développement logiciel, affecter grandement l'engagement d'une équipe vis-à-vis le plan de projet.

Enfin, l'atteinte de l'objectif peut se mesurer tant par la disponibilité et la fluidité de l'information que par la participation des parties prenantes au projet de façon régulière et continue.

### **3.5 Pratiques de la méthode Scrum**

Après l'analyse des objectifs CMMI liés au domaine de processus Planification de projet, il convient d'analyser les pratiques de la méthode Scrum.

En proposant les pratiques Équipe autodirigée et auto-organisée, Aucun ajout pendant l'itération et Pare-feu, Larman fait en sorte de raffiner des pratiques proposées par Schwaber. Cependant, la seule pratique de Larman analysée dans le cadre d'une matrice de concordance est Planification et organisation d'avant-match, puisque les autres pratiques constituent des raffinements aux pratiques Schwaber, ou ne s'appliquent pas à première vue au domaine de processus Planification de projet du CMMI.

#### **3.5.1 Sprint**

Le mot Sprint, emprunté au domaine sportif, est défini par Schwaber et Beedle comme étant le travail que doit effectuer une équipe pendant une période de temps prédéfinie ([20], p. 50). Elle se distingue des autres pratiques, car elle encadre les règles de travail pendant une période donnée au cours de laquelle une portion du produit est effectivement réalisée, testée et produite dans un état permettant sa livraison.

Cette pratique est au cœur de la méthode Scrum puisque c'est par elle que le travail est accompli de façon itérative et incrémentale. L'objectif est d'obtenir de petites victoires afin d'améliorer la productivité de l'équipe ([20], p. 50) car ces victoires permettent de construire les équipes ([06], p. 104).

Tout changement affectant les ressources impliquées dans le Sprint va l'impacter manifestement puisque les ressources utilisées ne sont pas interchangeables. Il est possible d'utiliser l'analogie de Brooks lorsqu'il commente le fait de comparer les personnes et le temps requis pour effectuer un travail : « *Men and months are interchangeable commodities only when a task can be partitioned among many workers with no communication among them* » ([5], p. 16)<sup>13</sup>.

Cette pratique vise à limiter le travail dans le temps en utilisant la technique du *Time Box* ([20] p. 143), et à favoriser l'accès à une rétroaction à tous les 30 jours environ en produisant un artefact fonctionnel ayant de la valeur pour le client. Avec un rythme de travail de 30 jours, il devient alors possible d'évaluer la vitesse (*velocity*) d'une équipe, c'est-à-dire sa capacité à réaliser une quantité de travail dans un temps donné ([8], p. 20). Mesurer quotidiennement l'avancement du travail permet d'obtenir des données sur les parties critiques et sur les risques du projet. Le rythme de 30 jours assure aux clients et

---

<sup>13</sup> Les personnes et les mois sont interchangeables seulement lorsqu'une tâche peut être distribuée entre des travailleurs sans qu'il y ait besoin de communication entre eux. (Traduction libre)

gestionnaires que l'équipe travaille sur ses priorités et que toute information critique est communiquée de façon à éviter que le projet ne s'enlise pendant des mois : « *How does a project get to be a year late? ... One day at a time* » ([5], p. 153)<sup>14</sup>. Enfin, cela permet, en cas de problème pendant un Sprint, de faire table rase, d'apprendre et de recommencer sur de nouvelles bases.

Les principales données et les principaux artefacts utilisés par la pratique ou produits par elle sont la liste des fonctionnalités une fois transformée en produit fonctionnel, la vitesse de l'équipe et les estimations établies par les membres des équipes techniques.

Pour réaliser la pratique du Sprint, il importe que la pratique Rencontre de planification du Sprint ait été réalisée puisque son issue détermine comment est réalisé le Sprint. Ainsi, il est essentiel que les parties prenantes et les gestionnaires identifient, dans la Liste des fonctionnalités du produit, quelles fonctionnalités ils doivent réaliser au cours de la prochaine période de 30 jours, et ce, dès le début du Sprint, et quels sont les principaux risques et contraintes découlant de ce choix. Pendant la réalisation de la pratique du Sprint, l'équipe doit soulever quotidiennement les contraintes qu'elle rencontre et informer de l'avancement du travail notamment par le biais d'une graphique démontrant le travail accompli (*Burndown Chart*).

---

<sup>14</sup> Comment un projet arrive-t-il un an en retard ? Un jour à la fois. (Traduction libre)



La mauvaise communication et le mauvais climat à l'intérieur d'une équipe peuvent mettre en péril la qualité et l'accessibilité de l'information. En effet, le gestionnaire responsable de l'équipe (*Scrum Master*) est responsable de mettre à jour la liste des fonctionnalités du Sprint ainsi que toutes les données qui y sont reliées. Il doit également veiller à ce qu'il y ait une mise à jour quotidienne de l'information. Cette visibilité donne la possibilité aux gestionnaires et aux parties prenantes d'évaluer la progression et de permettre l'allocation de ressources supplémentaires pour enlever tout élément perturbateur du projet.

Puisque la pratique du Sprint permet d'évaluer la vitesse d'une équipe, il n'est pas possible d'ajouter au cours d'un Sprint d'autres fonctionnalités afin de favoriser l'émergence de la responsabilisation de l'équipe. Il n'est également pas possible d'interrompre un Sprint sauf pour des circonstances exceptionnelles. Les équipes ne doivent pas être dérangées et les gestionnaires, tel que les parties prenantes, ne peuvent leur dire comment procéder. Pendant le Sprint, l'équipe est relativement en contrôle sur la façon d'atteindre les buts fixés. Évidemment, plusieurs organisations ont de la difficulté à donner ce genre de liberté à une équipe et doivent considérer qu'il s'agit d'un risque limité à 30 jours ([20], p. 51).

La pratique Sprint limite donc les risques par la réalisation et la mesure de petits progrès. Il n'est pas recommandé d'aborder un Sprint comportant une tâche qui s'échelonne sur plusieurs mois, ce qui est de toute manière une mauvaise pratique dans n'importe quelle

gestion de projet, agile ou non. Cette tâche doit être décomposée de façon à ce qu'il soit possible d'en réaliser des portions dans un Sprint.

### **3.5.2 Planification et organisation d'avant-match**

Les auteurs de la méthode Scrum utilisent, tel que mentionné au deuxième chapitre, une terminologie empruntée au rugby ([20], p. 121). Le mot Scrum fait lui-même référence à un moment précis de la pratique du sport du rugby. Puisqu'aucune autre référence que la brève description fournie par Larman n'est disponible ([16], p. 117), il est possible de spéculer sur l'expression *Pre-game* qui fait référence au moment où les joueurs d'une équipe se rencontrent avant une partie et discutent de la stratégie et des tactiques à employer. Il s'agit donc de planifier le match ou, autrement dit, le projet, selon la métaphore utilisée par Schwaber et Beedle. D'ailleurs, ceux-ci ne sont pas les seuls à comparer le développement logiciel à un jeu puisque Cockburn considère lui aussi que le développement logiciel est « *a cooperative game of invention and communication* » ([16], p. 31)<sup>15</sup>.

La pratique Planification et organisation d'avant-match, bien qu'elle ne soit pas explicitement définie par les auteurs de la méthode Scrum, constitue la première planification réelle du projet. Elle se distingue également de la pratique Rencontre de

---

<sup>15</sup> Le développement logiciel est un jeu coopératif alliant inventivité et communication. Le premier objectif de ce jeu est de livrer un logiciel fonctionnel et utile. Le second objectif, le résidu du premier objectif, est de préparer le prochain match. (Traduction libre)

planification du Sprint par le fait qu'il s'agit souvent de la première rencontre avec les parties prenantes au projet. C'est enfin lors de cette rencontre que les rôles des parties prenantes sont identifiés et qu'une liste de fonctionnalités est produite.

Cette pratique est essentielle à la méthode puisque toutes les parties prenantes participent à la rencontre, les rôles sont identifiés, et une liste de fonctionnalité est créée incluant des corrections, des améliorations ou des ajouts en quantité suffisante pour être en mesure de réaliser un premier Sprint ([16], p. 117). La culture d'entreprise, l'absence d'une partie prenante et l'expertise de l'équipe sont autant d'éléments affectant la pratique Planification et organisation d'avant-match. Cette pratique vise également à prioriser les fonctionnalités, à établir un premier échéancier de réalisation (*Release Backlog*) ([16], p. 117) et à permettre à l'équipe de s'entendre sur les bases de fonctionnement de la méthode de développement et sur les bases du projet. Enfin, elle vise à établir un contact avec les parties prenantes et à établir une liste de contraintes, technologiques ou non, affectant le projet.

Parmi les informations essentielles à la pratique, il est possible d'identifier notamment les fonctionnalités à développer, leurs priorités, la liste des parties prenantes et des membres de l'équipe, les contraintes du projet, qu'elles soient d'ordre technologique ou non, ainsi qu'un estimé d'effort pour ces fonctionnalités. Ces données proviennent autant de l'équipe et des parties prenantes que des gestionnaires. Puisque cette pratique ne s'exerce qu'une seule fois

au début du projet, l'information recueillie pour la réaliser n'est pertinente qu'une seule fois et n'est utile que pour la durée du Sprint. Cependant, plusieurs artefacts résultent de la réalisation de cette pratique, en particulier la liste des fonctionnalités, l'échéancier de réalisation, la liste de contacts, de contraintes techniques ou non, la définition des rôles et la liste des fonctionnalités sur lesquelles l'équipe s'est engagée à livrer pour la date de révision du Sprint. ([21], p. 142)

Si les bonnes parties prenantes sont absentes lors de cette pratique, cela peut remettre en question la qualité même de l'information qui y est consignée. C'est pour cette raison précisément que les gestionnaires du projet, les membres de l'équipe et les parties prenantes doivent y participer. La qualité de l'information peut également être affectée si les parties prenantes présentes ne détiennent pas toutes les données nécessaires pour l'équipe ([26], p. 122). Il en va de même si certaines données sont confidentielles et ne peuvent être communiquées à l'équipe. La qualité de l'information peut même mettre en péril l'exécution de la pratique, de même que l'incompréhension du travail à réaliser, du domaine d'affaires ou des besoins du client, ainsi que la tergiversation des clients sur les priorités.

La pratique Planification et organisation d'avant-match, comme toutes les autres pratiques de la méthode Scrum, est difficile à utiliser dans des organisations hiérarchiques et autoritaires. Pour ces organisations, la liberté de parole nécessaire à la pratique et la

méthode peuvent constituer un frein bien réel ([20], p. 153). L'autre élément négatif est le peu de temps habituellement consacré à cette pratique, ce qui peut faire en sorte qu'il soit possible de prendre une mauvaise direction et de devoir changer ou recommencer certaines fonctionnalités, ce que peut exceptionnellement permettre la pratique Sprint ([20], p. 53).

### **3.5.3 Rencontre de planification du Sprint**

La méthode Scrum définit la pratique Rencontre de planification du Sprint comme étant une rencontre d'une durée de huit heures au début de chaque Sprint et qui a pour but de prioriser la Liste des fonctionnalités du produit, d'établir lesquelles sont livrées à la fin du prochain Sprint et d'obtenir l'engagement de l'équipe à cet égard ([21], p. 142). À la différence de la pratique précédente, celle-ci vise plutôt à rajuster la liste des fonctionnalités selon l'information obtenue par la réalisation du Sprint précédent. Cette pratique est essentielle à la méthode puisqu'elle sert à déterminer le but et les fonctionnalités pour chaque Sprint subséquent. C'est par cette pratique également que l'équipe divise les objectifs en tâches individuelles ([20], p. 47).

En plus d'être affectée par tous les éléments mentionnés dans la pratique Planification et organisation d'avant-match, la pratique Rencontre de planification du Sprint est également affectée par l'atteinte ou non des objectifs fixés dans le Sprint précédent. Cette pratique vise principalement à permettre à l'équipe et au projet de s'adapter en fonction des données

récemment acquises. Cette adaptation se reflète par une révision des estimations, des priorités, des objectifs et des contraintes liés au projet.

Les réponses aux questions visant l'information pertinente et le système, hormis ce qui est mentionné précédemment, sont essentiellement les mêmes que pour la pratique Planification et organisation d'avant-match à cette nuance près qu'elles sont récurrentes à tous les mois et qu'elles dépendent de l'historique de réalisation du Sprint précédent.

#### **3.5.4 Liste des fonctionnalités du produit**

La pratique Liste des fonctionnalités du produit est à la fois un artefact et une activité. D'une part, il s'agit d'un document qui contient une liste de fonctionnalités priorisées (ce qui inclut les améliorations ou correction d'un produit). D'autre part, il s'agit d'une activité visant à construire cette liste et à rassembler plusieurs données à l'intérieur du document.

Cette pratique se distingue des autres puisque l'assemblage et les modifications des données constituant le document s'effectuent une fois au début de chaque Sprint avec la participation de l'équipe, des gestionnaires et autres parties prenantes. La spécificité de cette pratique est que les parties prenantes doivent s'impliquer pour établir la priorité des

fonctionnalités. La pratique Liste des fonctionnalités du produit est donc essentielle à la méthode puisqu'elle constitue la liste et l'ordre du travail à accomplir.

L'implication des parties prenantes, la complexité du développement et la volatilité des besoins sont des circonstances particulières pouvant affecter la pratique Liste des fonctionnalités du produit. Cette pratique vise à définir les objectifs à atteindre, à les prioriser, à identifier les objectifs à réaliser lors d'un prochain Sprint et à donner un aperçu de l'ampleur de chacun des objectifs. En bref, elle permet à l'équipe, aux gestionnaires et aux parties prenantes de s'entendre et de se commettre sur la charge de travail à réaliser.

La réalisation de la Liste des fonctionnalités du produit s'effectue avec les besoins et priorités clairement exprimées du client et avec l'expérience de l'équipe lors du Sprint précédent. Ces données alimentent la pratique au moins une fois par Sprint pendant toute la durée du projet. Elles servent notamment à permettre aux intervenants du projet de prendre des décisions éclairées sur les prochaines orientations du projet. Un sous-produit de cette pratique est la liste des fonctionnalités du Sprint (*Sprint Backlog*). Cette liste est constituée d'un extrait de la Liste des fonctionnalités du produit et ne vise qu'un seul Sprint. Elle contient les données d'origine en plus de contenir de l'information sur l'évolution quotidienne des fonctionnalités et tâches en effort.

La pratique Liste des fonctionnalités du produit définit cependant mal les dépendances entre les objectifs. La légèreté et la facilité à modifier des priorités et à ajouter des objectifs peuvent également nuire à une certaine stabilité si aucun effort n'est investi pour bien comprendre l'impact de ces changements.

### **3.5.5 Équipe Scrum**

L'autonomie vis-à-vis la répartition des tâches et l'organisation du travail constitue le principal objectif de la pratique Équipe Scrum ([20], p. 28). Cette auto-organisation et cette auto-assignation constituent les éléments essentiels de l'engagement de l'équipe envers la réalisation du Sprint. L'équipe est composée de toutes les personnes nécessaires pour atteindre l'objectif du Sprint. Il s'agit généralement d'une équipe de cinq à sept personnes ayant des expertises diverses et complémentaires ([20], pp. 36-37). Puisque les tâches ne sont pas encore assignées lors de la formation d'une Équipe Scrum, il est de la responsabilité de chacun des membres de l'équipe d'identifier à qui sont assignés chaque tâche et l'état d'avancement de celles-ci. L'équipe doit également prendre en compte, lors de l'auto-assignation des tâches, les priorités établies préalablement.

La liste des fonctionnalités qui sont réalisées dans le cadre du Sprint rassemble les données essentielles de l'équipe, car c'est avec celles-ci qu'elle est en mesure de déterminer le travail à réaliser et d'identifier les contraintes. Ces données, remises par les parties



prenantes et par les gestionnaires, sont utiles tout au long du Sprint et sont mises à jour à la fin de celui-ci.

Les données que l'équipe récupère du Sprint précédent, incluant celles de la « Revue du Sprint », abordé dans une prochaine section, sont également essentielles à l'équipe puisqu'elles servent à mieux déterminer sa capacité de travail de même que pour améliorer leur méthode de réalisation du travail et ce pendant toute la durée du projet.

La pratique Équipe Scrum permet d'identifier le *Scrum Master* de l'équipe, l'assignation des tâches et leur état, l'identification des contraintes et des problèmes et, enfin, l'engagement des membres de l'équipe sur la réalisation des objectifs pour un Sprint donné ([20], p. 31).

La pratique Équipe Scrum tient compte du fait que ce sont des humains qui créent des applications et qu'à cet égard, la méthode Scrum doit favoriser le travail d'équipe. L'un des éléments qui nuisent à l'équipe est ce que DeMarco et Lister appellent la gestion défensive ([11], p. 133), c'est-à-dire ce qui se produit quand les gestionnaires ne veulent courir aucun risque avec leur équipe et tentent de contrôler chacune des variables. Par exemple, certains gestionnaires peuvent assigner eux-mêmes les tâches et leurs priorités pour tenter de maximiser la productivité de chaque personne selon son expertise. La pratique Équipe Scrum vise donc à favoriser la complicité d'une équipe pour que les engagements ne soient

pas pris sur une base individuelle mais en fonction du groupe. L'échec de l'un devient alors l'échec de tous. De Marco et Lister ajoutent même que les projets sur lesquels les gestionnaires n'ont fait aucune pression de quelque façon que ce soit sont ceux qui ont eu le plus haut taux de productivité ([11], p. 29).

La culture d'une équipe ou de l'entreprise peut nuire à la pratique Équipe Scrum, par exemple lorsque la culture de l'équipe ou de l'organisation est fortement hiérarchique ([6], p. 106). Il en est de même pour l'hyper spécialisation des ressources ou encore pour l'absence des ressources essentielles à la réalisation de l'objectif.

DeMarco et Lister considèrent que la raison d'être d'une équipe n'est pas d'atteindre un but, mais de s'aligner vers ce but ([11], p. 126). Dans le cadre de la pratique Équipe Scrum, le but est également de diriger les gens vers un objectif commun, et ce, en fournissant aux personnes un environnement dans lequel elles peuvent travailler de façon plus productive ([20], p. 36). Dans cet environnement, la définition des rôles doit être clairement établie dès le début du projet. La méthode Scrum identifie et définit les rôles de *Scrum Master* et de parties prenantes ([16], p. 119).

La qualité de l'information est liée de façon intrinsèque à d'autres pratiques de la méthode Scrum. Plus spécifiquement, la pratique Équipe Scrum est limitée par les règles et standards qui entourent l'organisation. Par exemple, malgré toute l'autonomie dont l'équipe croit

disposer, elle ne peut, de son propre chef, transgresser les règles d'opération et d'exploitation en place dans l'entreprise; certains interdits demeurent.

Après l'analyse de la pratique Équipe Scrum, il est possible d'identifier en tant qu'éléments essentiels l'engagement de l'équipe, l'auto-organisation et l'auto-assignation du travail. La prochaine section présente comment l'organisation et l'assignation sont étroitement liées à la pratique Rencontre quotidienne.

### **3.5.6 Rencontre quotidienne**

Cette pratique est différente des autres pratiques puisqu'elle sert principalement à synchroniser les efforts de l'équipe vis-à-vis la réalisation des objectifs du Sprint.

Pour la méthode Scrum, c'est par cette pratique que l'organisation de l'équipe peut s'effectuer de façon quotidienne. L'agilité de l'équipe provient de la communication quotidienne d'information. Les données recueillies durant la journée permettent d'identifier les dépendances entre les tâches et les contraintes de façon à ce que, rapidement, le *Scrum Master* règle ces difficultés. La visibilité que procurent les rencontres quotidiennes aide notamment à renforcer un climat amical et à augmenter la confiance dans l'équipe et vis-à-vis elle ([6], p. 105).

Pour être efficaces, tous les membres de l'équipe doivent participer à cette pratique. Les entreprises qui permettent aux employés d'être sur les lieux du travail à des heures variables affectent cependant cette pratique, tel que la dispersion des membres de l'équipe sur différents fuseaux horaires d'ailleurs.

Tel que mentionné, l'objectif principal de la pratique Rencontre quotidienne est de synchroniser les efforts de l'équipe. Cette synchronisation s'effectue par la mise à jour des données sur le travail en cours, sur le travail effectué et sur l'identification de tout risque ou problème pouvant nuire à l'avancement du travail. Indirectement, cette pratique permet également de prendre le pouls de l'équipe et de s'assurer qu'elle travaille sur les bons objectifs selon les priorités établies.

L'estimation du temps de travail qui reste à accomplir sur une tâche est une donnée essentielle à la pratique puisqu'elle permet aux gestionnaires de mettre à jour la Liste des fonctionnalités du Sprint afin de déterminer la vitesse de l'équipe. Selon ces données, il est possible d'évaluer si l'équipe est en mesure de réaliser ses objectifs. Ces données, de même que la liste des contraintes, sont fournies quotidiennement par l'équipe au *Scrum Master* et sont utiles jusqu'à la prochaine rencontre ou jusqu'au retrait de la contrainte.

La liste de contraintes et la mise à jour de la « Liste de fonctionnalités du Sprint » sont les principaux artefacts affectés par cette pratique. La pratique est limitée dans le temps et sur

le contenu. D'une part, la pratique doit s'exercer en 15 minutes environ et ne doit pas excéder 30 minutes. D'autre part, il ne s'agit pas de discuter de tous les éléments, mais de répondre à trois questions relatives à la réalisation des activités des 24 heures précédentes, à celles qui sont réalisés au cours des 24 heures de travail subséquentes et des contraintes ou obstacles rencontrés ou susceptibles de l'être. Seuls les membres de l'équipe participent à la rencontre. Les gestionnaires et parties prenantes peuvent y assister, mais ils ne peuvent participer, la seule exception permise étant dans le cas où l'équipe travaille sur d'autres objectifs que ceux fixés au début du Sprint.

L'étude de la pratique « Rencontres journalière » permet de mettre en relief deux éléments soit un compte-rendu de l'avancement des travaux et une réévaluation quotidienne des estimations.

### **3.5.7 Revue du Sprint**

La pratique « Revue du Sprint » est en quelque sorte une rétrospective vers laquelle convergent les résultats de presque toutes les autres pratiques. C'est lors de son exercice que la démonstration de l'atteinte des objectifs de l'équipe est faite. C'est également à ce moment que les parties prenantes et les gestionnaires font notamment un retour sur ce qui a été réalisé.

C'est essentiellement par le biais de cette pratique que l'équipe peut améliorer d'une part le produit et d'autre part son processus de réalisation du travail depuis la dernière itération. C'est au moment de faire le point sur le travail accompli que l'équipe peut évaluer sa performance. C'est également par la réalisation de cette pratique que la pratique « Revue de planification du Sprint » est possible.

L'un des objectifs de la pratique « Revue du Sprint » est d'exposer ce qui doit être amélioré au cours du prochain Sprint. Cependant, la culture d'entreprise peut affecter cette pratique. Par exemple, lorsqu'une entreprise tolère peu l'échec, un dialogue franc sur les problèmes rencontrés et leurs causes peut être difficile.

La pratique « Revue du Sprint » vise non seulement à démontrer l'atteinte des objectifs, mais également à recueillir la rétroaction de l'équipe sur le déroulement du Sprint, des parties prenantes sur la satisfaction par rapport à la réalisation des objectifs ou tout autre commentaire visant à améliorer le produit. Elle vise également obtenir toute information permettant de mieux estimer les fonctionnalités restantes, les risques et les contraintes. À cet égard, la « Liste des fonctionnalités du Sprint » et le graphique exprimant la vitesse de l'équipe sont des informations qui servent notamment à établir les prochaines estimations. Cette pratique ne se réalise qu'à la fin d'un Sprint, tout au long du projet.

En plus, la « Revue du Sprint », qui ne dure qu'une demi-journée environ, doit permettre d'obtenir une liste des leçons apprises par l'équipe. Cette liste peut constituer en une série d'éléments de risque à tenir compte lors du prochain Sprint. Également, cette pratique doit permettre l'acceptation des travaux tels quels ou, s'il y a lieu, avec modifications.

L'absence de gestionnaires, des parties prenantes ou de certains membres de l'équipe affecte la qualité et la pertinence de l'information communiquée lors de la pratique « Revue du Sprint ». Le formalisme entourant cette rencontre peut nuire aux échanges où la rétroaction de l'équipe, des gestionnaires et des parties prenantes est essentielle ([20], p. 56). Ce qu'il faut retenir de cette pratique c'est d'une part qu'il s'agit d'une étape de validation et d'amélioration et, d'autre part, que l'information présentée peut servir tant à identifier les risques qu'à mieux estimer les prochaines fonctionnalités.

## **Chapitre 4**

### **Matrice de concordance**

Après avoir procédé à l'analyse à la fois de trois buts du CMMI et de certaines des pratiques de la méthode Scrum, il convient d'une part de réaliser une matrice de concordance entre les buts CMMI et les pratiques de la méthode Scrum en statuant sur les éléments qui ne sont pas explicitement concordants ainsi que sur les raisons qui justifient l'établissement d'une telle concordance. D'autre part, il s'agit de soutirer de l'exercice d'analyse réalisé dans le chapitre précédent et de l'établissement d'une matrice de concordance des constats et des apprentissages.

#### **4.1 Qualification des croisements**

La réalisation du croisement entre les buts CMMI et les pratiques de la méthode Scrum, notamment par le biais d'une matrice, aide à déterminer s'il est possible d'affirmer qu'il y a une concordance entre les buts CMMI analysés et certaines pratiques de la méthode Scrum. Le résultat de ces croisements est alors qualifié par la façon dont il est possible de prouver l'atteinte ou la réalisation des buts CMMI. Ce croisement peut comporter une inscription qualifiant la concordance. L'absence d'inscription signifie qu'il n'existe pas de concordance directe ou indirecte.



Une inscription de type D indique qu'une preuve directe permet de prouver qu'il est possible d'atteindre ou de réaliser l'un des éléments d'un but CMMI. La production d'un artefact résultant de la réalisation d'une pratique de la méthode Scrum constitue une telle preuve directe. Lorsque l'inscription est de type I, cette preuve peut être établie indirectement par le biais d'autres artefacts dont la finalité n'est pas nécessairement l'atteinte de l'élément du but CMMI. Enfin, lorsque l'existence de la rencontre de l'élément identifié ne peut se prouver que par le biais de témoignage, l'inscription est de type A. L'inscription, si elle est en majuscule, indique que l'élément est totalement rencontré par la pratique; si elle est en minuscule, c'est qu'elle n'est que partiellement rencontrée.

Lorsqu'il est établi qu'un des éléments peut être rencontré par une preuve de type D, les autres types ne sont alors pas considérés. Également, lorsqu'il s'agit d'une preuve de type I, il n'est pas nécessaire de spécifier si une preuve de type A existe également. Enfin, en présence de preuves partielles et complètes de types différents, les preuves partielles ne sont pas considérées.

Le tableau présenté dans la prochaine section représente un extrait d'une matrice de concordance des buts CMMI et des pratiques de la méthode Scrum. Les colonnes présentent tous les éléments identifiés précédemment lors de l'analyse des buts CMMI. Les rangées présentent chacune des pratiques de la méthode Scrum également analysées dans le chapitre précédent. La dernière rangée permet d'obtenir un indice de concordance global

pour chacun des éléments qui est qualifié de la même façon que les croisements de la matrice et selon les mêmes règles.

## **4.2 Réalisation d'une matrice de concordance**

Afin de faciliter l'observation des concordances, l'utilisation d'une matrice est privilégiée. Dans un premier temps, les éléments sont identifiés pour chacun des trois buts du CMMI analysé soit : « Délais de réalisation », « Échelle de coûts », « Efforts nécessaires » et « Expertise requise » pour le but Établir les estimations. Les éléments « Échéancier de réalisation », « Listes d'objectifs priorisés », « Calendrier de réalisation », « Liste des dépendances entre les objectifs », « Liste des contraintes », « Risques du projet », « Méthode de supervision des projets », « Fréquence des contrôles » et « Méthode de gestion des changements » sont liés au but Développer un plan de projet. Finalement, les éléments « Disponibilité des ressources », « Circulation de l'information » et « Participation des parties prenantes pour qualifier l'avancement du projet » sont liés au but Obtenir l'engagement envers le plan.

Aux fins du présent essai, chacune des qualifications n'est pas justifiée outre mesure puisque l'attention se porte principalement sur les éléments dont le croisement avec les pratiques ne comporte aucune preuve directe. Puisqu'il s'agit d'une interprétation plutôt

subjective, il convient alors de la justifier pour, à tout le moins, permettre l'ouverture d'un débat à cet égard lors d'études complémentaires.

Le tableau 4.1 vise à présenter une matrice de concordance entre les pratiques de la méthode Scrum et trois des objectifs du domaine de processus CMMI *Project Planning*.

**Tableau 4.1 Matrice de concordance CMMI / Scrum**

<b>Pratiques de la méthode Scrum</b> \ <b>Buts CMMI</b>	Délais de réalisation	Échelle de coûts	Efforts nécessaires	Expertise requise	Échéancier de réalisation	Listes d'objectifs priorités	Calendrier de réalisation	Liste des dépendances entre les objectifs	Contraintes	Risques du projet	Méthode de supervision des projets	Fréquences des contrôles	Méthode de gestion des changements	Disponibilité des ressources	Circulation de l'information	Participation des parties prenantes pour qualifier l'avancement du projet
Planification et organisation d'avant-match	D		D	i	D	D	D		D					i	A	
Rencontre de planification du Sprint						D	D	i		I	I	I	D		A	
Liste des fonctionnalités du produit		I	D			D	I		I		D	D	D	I	I	A
Sprint	D	I	D		D	D	D				D	D	I		D	A
Équipe Scrum				A										A	A	
Rencontre quotidienne			D			D			D	A	A	A		A	D	A
Revue du Sprint	D	D	D	i	D	D	D	i	D	D	D	D	D		A	I
Qualification maximale obtenue	D	D	D	A	D	D	D	i	D	D	D	D	D	I	D	I

La lecture de ce tableau suggère que l'élément « Expertise requise » peut être prouvé partiellement par des artefacts indirects ou complètement par le biais de témoignage. En effet, c'est n'est qu'en questionnant les membres de l'équipe qu'il est possible de déterminer si celle-ci possède toute l'expertise requise en considérant que la réalisation de la pratique Équipe Scrum demande que l'équipe soit composée de toutes les personnes nécessaires pour atteindre l'objectif. Bien que la pratique Planification et organisation d'avant-match ne propose pas directement une liste de l'expertise requise, il est possible d'en déduire une partie par l'identification des rôles. La « Revue de Sprint » permet également de réaliser la même chose si l'un des risques ou problèmes rencontrés dans le Sprint précédent est lié à l'expertise de l'équipe.

Également, l'élément « Liste des dépendances entre les objectifs » ne semble pas être en mesure d'être prouvé complètement et que de manières indirectes. Tant la pratique « Revue du Sprint », Rencontre de planification du Sprint et Rencontre quotidienne ne font qu'établir partiellement une liste de contraintes technologiques ou non ou encore de risques auxquels ont doit tenir compte au cours d'un Sprint subséquent. Aucune dépendance n'est directement établie entre les objectifs, mais de Sprint en Sprint, ces dépendances émergent et peuvent être considérées comme des contraintes ou des priorités.

Indirectement, Disponibilité des ressources se prouve par la Liste des fonctionnalités du produit puisque cette pratique vise à ce que tous s'entendent sur la charge de travail à

accomplir et, surtout, à ce que tous s'engagent à cet égard. Par conséquent, l'engagement implique que les ressources sont disponibles. Toujours indirectement, mais partiellement, l'identification des rôles lors de la Planification et organisation d'avant-match permet de connaître la disponibilité de certaines ressources. Également, que ce soit par le biais des pratiques Équipe Scrum et Rencontre quotidienne, il est possible de connaître réellement l'implication des ressources, que ce soit par leur présence effective ou par leur apport aux tâches.

Enfin, l'élément Participation des parties prenantes pour qualifier l'avancement du projet ne semble pas être démontrable autrement que par l'affirmation des personnes impliquées dans les pratiques Liste des fonctionnalités du produit, Sprint, Rencontre quotidienne et Revue du Sprint. Pour chacune de ces pratiques, l'implication des parties prenantes est essentielle, que ce soit pour s'entendre sur la charge de travail à effectuer, les fonctionnalités à développer, dans quelles priorités et les résultats à obtenir à la fin d'un Sprint. Ce n'est que par la satisfaction de la réalisation des objectifs que se mesure réellement l'avancement du projet.

### **4.3 Constats et apprentissage sur les méthodes d'analyse utilisées**

Après l'analyse des buts CMMI et de certaines pratiques de la méthode Scrum, il est possible de constater que les méthodes d'analyse décrites aux sections Méthode utilisée

(3.1), Méthode d'analyse d'objectifs CMMI (3.2) et Méthode d'analyse des pratiques de la méthode Scrum (3.3) souffrent de faiblesses.

D'abord, une première ébauche des questions a révélé que certaines des réponses, notamment celles visant les circonstances pouvant affecter le projet ou l'entreprise, se retrouvent dans presque toutes les pratiques de la méthode Scrum. Dans le cadre d'une prochaine analyse, l'établissement d'une liste de question s'adressant spécifiquement à la méthode dans son ensemble peut être envisagé, un peu comme pour les objectifs généraux du CMMI. Il peut s'agir, par exemple, d'une série de questions génériques lorsque les réponses recourent plusieurs pratiques.

De plus, il arrive qu'il soit difficile de répondre à certaines questions. Par exemple, lors de l'analyse de la pratique Équipe Scrum, il est difficile de déterminer ce qui doit résulter de la réalisation de cette pratique. Ce dont il faut se souvenir, c'est que les questions servent de guide, d'aide visant à identifier les éléments qui sont utilisés pour établir la concordance. C'est pourquoi les questions posées ne sont pas des incontournables et laissent place à amélioration. L'objectif est de trouver les éléments essentiels et uniques à chacune des pratiques de façon à les mettre en relief vis-à-vis les objectifs du CMMI.

Évidemment, c'est au moment de répondre aux questions qu'il est possible de constater ou non si elles sont bien formulées. Si des questions différentes engendrent deux ou même

trois fois les mêmes réponses, c'est qu'elles se ressemblent trop, que le nombre de questions est trop élevé pour qu'il soit possible de nuancer efficacement les réponses, ou encore que les questions laissent trop de place à des réponses ambiguës. Par exemple, la question « Quels sont les éléments positifs liés à l'utilisation de cette pratique? » cherche les mêmes éléments de réponse que « En quoi cette pratique est essentielle à la méthode? » ou encore « Qu'est-ce que cette pratique vise à accomplir? ». La difficulté peut également provenir du fait que l'essentiel de l'analyse repose sur la traduction des buts et des pratiques rendant ainsi possible une possibilité d'erreur, de mauvaise interprétation ou traduction contaminant ainsi la source d'une matrice de concordance.

Il arrive également que les questions s'adressent à la méthode et non aux pratiques. Dans ce cas, il est difficile d'obtenir une réponse adéquate puisque l'aspect méthodologique n'est pas analysé dans le cadre de la concordance. Par exemple, la question sur la priorité de la pratique « Quelles sont les priorités de la pratique par rapport aux autres? » est difficile à répondre en raison l'osmose de certaines pratiques de la méthode Scrum. Il semble également inutile de demander le rôle et les responsabilités de la pratique : l'objectif est d'identifier les rôles et responsabilités associés à la méthode et non pas à la pratique.

Les questions doivent servir à rechercher toute l'information nécessaire à l'établissement de la concordance. Une question ne visant pas cette fin est inutile et les données qui en résultent constituent un fardeau supplémentaire par exemple la question « Quelle

information est essentielle à la compréhension de cette pratique? ». Cette information peut sembler pertinente, mais n'aide pas à établir une matrice. Également, les références et la question visant à savoir qui utilise ces pratiques peuvent être utiles, mais n'aident pas nécessairement à établir la concordance ni à déterminer si une pratique est dynamique dans le sens où elle continue d'évoluer et ne reste pas statique. Par ailleurs, même si la question relative à l'importance d'un objectif CMMI a été conservée, il est possible de remettre en question sa pertinence puisque son apport à l'analyse est limité.

Parfois il peut être utile de regrouper certaines questions quand les réponses sont étroitement liées par exemple : « Qu'est-ce qui peut mettre en péril la qualité de l'information fournie? », « Qu'est-ce qui peut mettre en péril l'accessibilité à l'information? » ou « Quelle information est essentielle à la réalisation de cette pratique, qui doit la fournir et à qui? ». La question qui vise à déterminer quelles sont les activités ou quels sont les artefacts essentiels équivaut à demander quelle information est nécessaire puisque celle-ci est très souvent transmise via une activité tel que Rencontre quotidienne ou un artefact comme un *Backlog*.

#### **4.4 Constats et apprentissage sur la matrice**

Dans un premier temps, le tableau a été établi de façon intuitive en comparant les buts aux pratiques sans aller plus en détail, et c'est lors de la rédaction des justifications que



plusieurs résultats des croisements ont été modifiés. Par conséquent, il faut faire attention aux premières impressions qu'il est tentant de faire lors de l'établissement d'une concordance.

De plus, lorsque vient le temps de justifier le résultat des croisements, il faut déterminer en quoi chacune des pratiques rencontre ou non l'élément. Or, cette justification doit se s'effectuer par objectif et non par pratique, car ce sont ces objectifs qu'il nous faut atteindre.

Le principal inconvénient de cette matrice de concordance est l'interprétation qu'il est tentant de faire en l'observant. En effet, en observant uniquement le tableau, il est tentant d'affirmer que seule la pratique « Revue de Sprint » est utile, car elle couvre d'une façon ou d'une autre presque tous les éléments des trois buts CMMI. Évidemment, une telle affirmation est fautive puisqu'elle ne prend pas en considération l'interdépendance entre les pratiques. Sans doute faut-il ajouter à la matrice de concordance une mise en garde visant à ne pas interpréter de façon indépendante chaque pratique, mais plutôt de les considérer interdépendantes les unes des autres.

## Conclusion

Malgré ses faiblesses et ses lacunes, l'analyse des objectifs et des pratiques à l'aide de questions intelligentes (tableaux 3.1 et 3.2) facilite l'identification de métriques visant à déterminer comment ces objectifs sont atteints. Ces mêmes questions permettent également d'identifier des éléments des pratiques de la méthode Scrum permettant d'établir et de mesurer la concordance avec ces objectifs. Il importe que l'analyse d'un objectif soit précédée de l'établissement d'une liste de questions aidant à identifier les éléments les plus pertinents ou même essentiels et ce, même si ces questions restent à parfaire.

La mesure de l'atteinte d'un objectif s'établit par la qualification de la rencontre des éléments essentiels avec la production des artefacts de la pratique et par la capacité de prouver cette atteinte. Une fois la mesure de l'atteinte de tous les objectifs établie, l'identification pour chacun d'eux des éléments qui n'ont pas rencontré par l'ensemble des pratiques de la méthode Scrum permet d'apprécier la qualité de la concordance avec les objectifs CMMI.

La méthode d'analyse proposée permet d'affirmer qu'il existe une concordance entre certaines pratiques de la méthode Scrum et les buts Établir les estimations, Développer un plan de projet et Obtenir l'engagement envers le plan, et ce, même si l'un des éléments du

deuxième but, soit la Liste des dépendances entre les objectifs, n'est que partiellement et indirectement rencontré.

L'exercice même de la réalisation d'une matrice est utile et bénéfique. Il est possible de l'utiliser pour établir une concordance entre les buts CMMI et les pratiques de la méthode Scrum, s'il est entendu que ces pratiques fonctionnent en symbioses et qu'une interdépendance des pratiques entre elles reste à établir afin de déterminer de façon absolue que seules l'une ou l'autre des pratiques ainsi exposées est réellement nécessaire pour atteindre les buts CMMI. En l'occurrence, il n'est pas possible de prétendre qu'une pratique a plus d'importance qu'une autre ou qu'elle n'est pas essentielle pour atteindre l'un des buts CMMI.

Que ce soit par les définitions ou la réponse aux questions posées, ou même par le choix des questions, l'avantage de la méthode proposée dans le cadre de cet essai est qu'elle s'expose aux critiques et à l'amélioration. La définition des termes peut être ainsi débattue. L'accent est mis sur l'élimination des malentendus et sur la clarification des ambiguïtés qui sont la source de nombreuses frustrations et interprétations douteuses. Le tout dans le but d'améliorer la méthode proposée.

## Liste des références

- [1] *Agile Manifesto*. Site web : <http://agilemanifesto.org/>, 2008.
  
- [2] Ahern, D. M., Couse, A. et Turne R., *CMMI Distilled*, 2<sup>e</sup> édition, Addison-Wesley, Boston, 2003, 305 p.
  
- [3] Basque, R., *Un itinéraire fléché vers le Capability Maturity Model Integration*, Dunod, Paris, 2004, 198 p.
  
- [4] Boehm, B. et Turner, R., *Balancing Agility and Discipline*, Addison-Wesley, Boston, 2004, 266 p.
  
- [5] Brooks, F. P. JR., *The Mythical Man-Month*, Anniversary Edition, Addison-Wesley, Boston, 1995, 322 p.
  
- [6] Cockburn, A., *Agile Software Development*, Addison-Wesley, Boston, 2002, 278 p.
  
- [7] Cockburn, A., *Agile Software Development*, 2e édition, prépublication, copyright 2006, extrait.
  
- [8] Cockburn, A., *Crystal Clear*, Addison-Wesley, Boston, 2004, 311 p.

- [9] Cohn, M., *Agile Estimating and Planning*, Prentice Hall, Upper Saddle River, 2006, 330 p.
- [10] Control Chaos. Site Web: <http://www.controlchaos.com>, 2008.
- [11] DeMarco, T. et Lister, T., *Peopleware: Productive Projects and Teams*, 2<sup>e</sup> édition, Dorset House, New York, 1999, 248 p.
- [12] De Villier, M.-E., *Multi dictionnaire de la langue française*, 4<sup>e</sup> édition, Québec Amérique, Montréal, 2003, 1542 p.
- [13] Fenton, N. E. et Pfleeger, S. L., *Software Metrics*, 2<sup>e</sup> édition, PWS Publishing Company, Boston, 1997, 638 p.
- [14] Grand dictionnaire terminologique. Office québécois de la langue française. Site Web: <http://www.granddictionnaire.com>, 2008.
- [15] Correspondance électronique du 9 juillet 2006 avec Hillel Glazer, chef de direction de la compagnie Entinex Inc., auditeur SCAMPI et instructeur du cours d'introduction au CMMI, 2006.

- [16] Larman, C., *Agile & Iterative Development: A Manager's Guide*, Addison-Wesley, Boston, 2004, 342 p.
- [17] Nadler G. et Chandon. W.J., *Smart Questions*, Jossey-Bass, San Francisco, 2004, 297 p.
- [18] Royce, W. *Managing the Development of Large Software Systems*, Proceedings, IEEE Westcon, 1970, pp 328-338.
- [19] Schrage, M., *The Struggle to Define Agility*, Site Web :  
<http://www.cio.com/archive/081504/schrage.html>, 2004.
- [20] Schwaber, K. et Beedle, M., *Agile Software Development with Scrum*, Prentice-Hall, Upper Saddle River, 2001, 158 p.
- [21] Schwaber, K., *Agile Project Management with Scrum*, Microsoft Press, Redmond, 2003, 165 p.
- [22] Software Engineering Institute, *Capability Maturity Model Integration (CMMI), Version 1.1, Staged Representation*, Carnegie Mellon, Pittsburgh, 2002, 624 p.

- [23] Software Engineering Institute, *Capability Maturity Model Integration (CMMI), Version 1.2, for Development*, Carnegie Mellon, Pittsburgh, 2006, 561 p.
- [24] Software Engineering Institute (Park, E. R., Goethert, W.B et Florac, W. A.), *Goal-Driven Software Measurement*, SEI, Carnegie Mellon, Pittsburgh, 1996, 170 p.
- [25] SPIN de Montréal. Site Web : <http://www.spin-montreal.org/index.php?module=EventReg&func=displayevent&eventid=17>, 2008.
- [26] Stephens, M. et Rosenberg, D., *Extreme Programming Refactored: The Case Against XP*, Apress, New York, 2003, 400 p.
- [27] Takeuchi, H. et Nonaka, I. *The New New Product Development Game*, Harvard Business Review, vol. 64, n° 1, 1986, pp. 137-146.
- [28] The Standish Group. Site Web : [http://www.standishgroup.com/sample\\_research/chaos\\_1994\\_1.php](http://www.standishgroup.com/sample_research/chaos_1994_1.php), 2008.
- [29] *Webster's New World College Dictionary*, 3<sup>e</sup> édition, Macmillan, New York, 1997, 1588 p.

## Annexe 1 – Bibliographie

Augustine, S., *Managing Agile Projects*, Prentice-Hall, Upper Saddle River, 2006, 229 p.

Beck, K., *Test Driven Development by Example*, Addison-Wesley, Boston, 2003, 220 p.

Beck, K., *Extreme Programming Explained. Embrace Change*, deuxième édition, Addison-Wesley, Boston, 2004, 189 p.

Boehm, B. et Turner, R., *Rebalancing Your Organization's Agility and Discipline*, site Web: <http://www.agileprojectmgt.com/docs/Boehm20.pdf>, 2003.

Boehm B., *Software Engineering Economics*, Prentice-Hall, Englewood Cliffs, 1981, 764 p.

Cajolet-Laganière, H., Collingue P. et Laganière, G., *Rédaction technique, administrative et scientifique*, 3<sup>e</sup> édition, Édition Laganière, 1997, 468 p.

Cattan, M., Idrissi, N et Knockaert P., *Maîtriser les processus de l'entreprise*, 4<sup>e</sup> édition, Édition d'Organisation, Paris, 2003, 304 p.

DeMarco, T. et Lister, T., *Waltzing with Bears*, Dorset House, New York, 2003, 196 p.



Fowler, M., *Refactoring. Improving The Design Of Existing Code*, Addison-Wesley, Reading Massachusetts, 1999, 431 p.

Glass, R. L., *Facts and Fallacies of Software Engineering*, Addison-Wesley, Boston, 2003, 195 p.

Harrison, N., *Pattern Languages of Program Design 4*, Addison-Wesley Professional, Boston, 1999, 784 p.

Highsmith, J., *Agile Project Management*, Addison-Wesley, Boston, 2004, 277 p.

Hunt A., et Thomas, D., *The Pragmatic Programmer*, Addison-Wesley, Reading Massachusetts, 1999, 321 p.

Poppendieck, M. et Poppendieck, T., *Lean Software Development: An Agile Toolkit*, Addison-Wesley, Boston, 2003, 203 p.

Potter, N. S., et Sakry, M., *Making Process Improvement Work*, Addison-Wesley, Boston, 2002, 169 p.

Rüping, A., *Agile Documentation*, John Wiley & Sons LTD, West Sussex, 2003, 226 p.

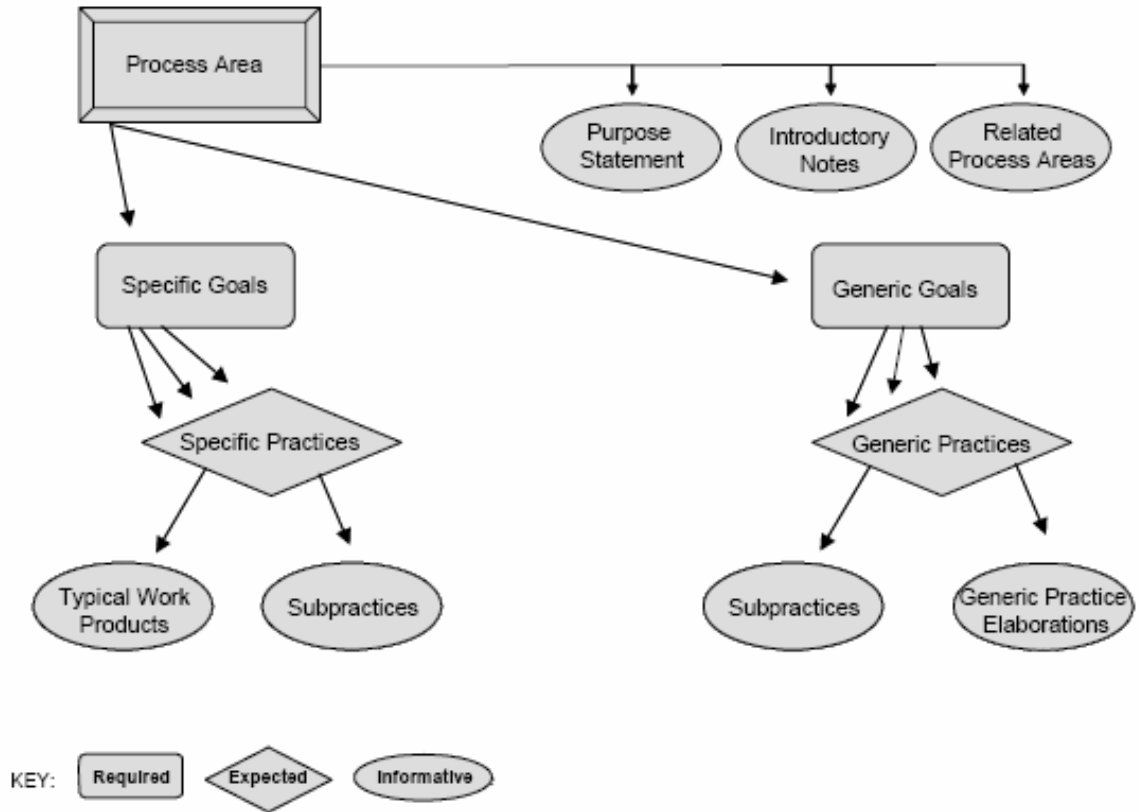
Tate K. *Sustainable Software Development*, Addison-Wesley, Upper Saddle River, 2006,  
226 p.

Turner, R., *Using CMMI to Balance Agile and Plan-driven Methods*, site Web:  
<http://www.dtic.mil/ndia/2003CMMI/Turner.ppt>, 2003.

<http://www.agilealliance.org/>

<http://www.agileprojectmgt.com/>

## Annexe 2 – Composante du modèle CMMI



### Annexe 3 – Auteurs et signataires du manifeste agile

Kent Beck: Créateur de la méthode *Extreme Programming* avec Ron Jeffries et Ward Cunningham, et auteur de plusieurs livres dans le domaine logiciel notamment *Extreme Programming Explained: Embrace Change* et *Test-Driven Development: By Example*.

Mike Beedle: Créateur de la méthode agile Scrum avec Ken Schwaber et Jeff Sutherland, et auteur de nombreuses publications liées au *Design Pattern* et au développement logiciel.

Arie van Bennekum: Membre du conseil d'administration du *Dynamic Systems Development Method (DSDM)* et auteurs d'articles sur DSDM.

Alistair Cockburn: Détenteur d'un doctorat de l'université d'Oslo, auteur et créateur de la famille de méthodes *Crystal*. Auteur de nombreux ouvrages dont *Agile Software Development: The Cooperative Game*, *Crystal Clear : A Human-Powered Methodology for Small Teams* et *Writing Effective Use Cases*.

Ward Cunningham: Créateur de la méthode *Extreme Programming* avec Kent Beck et Ron Jeffries, développeur du premier wiki et auteur d'un livre sur le sujet.

Martin Fowler: Auteurs de nombreux livres dont *Analysis Patterns: Reusable Object Models*, *Refactoring: Improving the Design of Existing Code* et *UML Distilled: A Brief Guide to the Standard Object Modeling Language*.

James Grenning: Auteur de nombreux articles publiés notamment dans la revue *IEEE software* et *C++ Report*.

Jim Highsmith: Créateur de la méthode *Adaptive Software Development*, récipiendaire du Jolt Awards en 2000 et auteur de nombreux livres dont *Agile Project Management: Creating Innovative Products*, *Agile Software Development Ecosystems* et *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*.

Andrew Hunt: Avec Dave Thomas, il est l'auteur du livre *The Pragmatic Programmer* et dirige la maison d'édition *The Pragmatic Bookshelf*.

Ron Jeffries: Créateur de la méthode *Extreme Programming* avec Kent Beck et Ward Cunningham et auteur des livres *Extreme Programming Installed* et *Extreme Programming Adventures in C#*.

Jon Kern: Auteur du livre *Java Design* et de nombreux articles sur la programmation.

Brian Marick, Auteur des livres *The Craft of Software Testing: Subsystem Testing Including Object-based and Object-oriented Testing* et *Everyday Scripting with Ruby: For Teams, Testers, and You*.

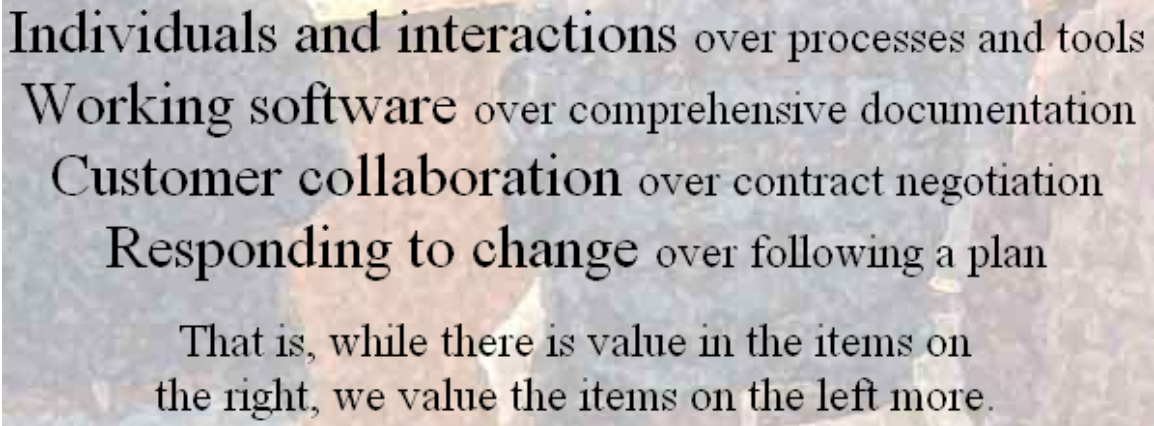
Robert C. Martin: Éditeur en chef du périodique *C++ Report* de 1996 à 1999 et auteur des livres *Agile Software Development: Principles, Patterns, and Practices*, *Designing Object-Oriented C++ Applications using the Booch Method* et *Agile Principles, Patterns, and Practices in C#*.

Ken Schwaber: Créateur de la méthode Scrum avec Jeff Sutherland et Mike Beedle, et auteur de livres sur la méthode dont *Agile Project Management with Scrum* et *Agile Software Development with Scrum*.

Jeff Sutherland: Co-créateur de la méthode Scrum avec Ken Schwaber et Mike Beedle.

Dave Thomas: Avec Andrew Hunt, il est l'auteur du livre *The Pragmatic Programmer* et dirige la maison d'édition *The Pragmatic Bookshelf*.

## Annexe 4 – Manifesto for Agile Software Development



**Individuals and interactions** over processes and tools  
**Working software** over comprehensive documentation  
**Customer collaboration** over contract negotiation  
**Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Traduction libre : « Manifeste agile : Les individus et les interactions doivent primer sur les procédures et les outils. Le développement logiciel doit primer sur la documentation exhaustive. La collaboration avec le client doit primer sur la négociation contractuelle. L'ouverture au changement doit primer sur le suivi d'un plan rigide. Ceci étant dit, nous considérons que les éléments de droite ont une valeur certaine. Cependant, nous valorisons encore plus les éléments de gauche » [1].

## Annexe 5 – Scrum et CMM

Level	Key Practice Area	Rating
2	Requirements management	✓✓
2	Software project planning	✓✓
2	Software project tracking and oversight	✓✓
2	Software subcontract management	
2	Software quality assurance	✓✓
2	Software configuration management	✓
3	Organization process focus	✓
3	Organization process definition	✓
3	Training program	✓
3	Integrated Software management	✓
3	Software product engineering	✓✓
3	Intergroup coordination	✓✓
3	Peer review	✓✓

Dans la matrice de concordance proposée par Ken Schwaber ([21], p. 153, les doubles crochets signifient que Scrum est totalement conforme (*Compliant*) au domaine de processus tandis qu'un simple crochet veut dire qu'il est majoritairement conforme à *CMM*.



