ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

RAPPORT DE PROJET DE 15 CRÉDITS, PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

DANS LE CADRE DU PROJET DE FIN D'ÉTUDES
DE LA MAÎTRISE EN GESTION DE PROJETS

PAR
Mauricio LOPES

COMPARAISON DE MÉTHODES AGILE ET SES OUTILS DE GESTION DE PROJETS EN LIGNE AVEC LES MÉTHODES ET OUTILS DU DOMAINE DE L'INGÉNIERIE

MONTRÉAL, LE 20 AOÛT 2015

Mauricio Lopes, 2015

Cette licence Creative Commons signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette œuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'œuvre n'ait pas été modifié.

PRÉSENTATION DU JURY

CE RAPPORT DE PROJET A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Alain April, directeur de projet
Département de Génie logiciel et TI à l'École de technologie supérieure

M. Witold Suryn, président du jury
Département de Génie logiciel et T.I. à l'École de technologie supérieure

REMERCIEMENTS

Je remercie le professeur Alain April pour sa patience et son dévouement envers moi et mon projet de recherche. Ensuite, je remercie ma famille, qui même de loin, m'a appuyé et m'a motivé à surmonter ce grand défi. Je remercie surtout ma conjointe, qui m'a soutenu et m'a encouragé, tout en prenant soin de nos enfants. Je les remercie pour leur patience et leur compréhension lorsque j'étais occupé avec ma recherche.

COMPARAISON DE MÉTHODES AGILE ET SES OUTILS DE GESTION DE PROJETS EN LIGNE AVEC LES MÉTHODES ET OUTILS DU DOMAINE DE L'INGÉNIERIE

Mauricio LOPES

RÉSUMÉ

Cette recherche exploratoire vise à étudier et comparer les méthodes et les outils de gestion de projets de développement de logiciels en ligne versus ses vis-à-vis dans le domaine de la gestion de projets d'ingénierie où je travaille depuis quinze ans.

Les méthodes utilisées par les équipes de projet du domaine de l'ingénierie et de la construction, sont guidées par les concepts du « Project Management Book of Knowledge » (PMBOK), créée par le « Project Management Institute » (PMI) et les outils tels que MS-Project et Oracle Primavera P6, sont utilisés pour faire la planification, le suivi et le contrôle de l'échéancier, du budget et des ressources allouées aux projets.

Les outils de gestion de projets de développement de logiciels sont orientés vers l'application de méthodes Agile, récemment apparues sur le marché et disponibles directement sur l'internet, en mode Cloud. Les outils tels que Jira-Agile et VersionOne sont basés sur la méthode Agile et intègrent la méthodologie SCRUM, pour faire la planification du « sprint », la gestion de tâches et de ressources. Ces outils tentent, entre autres choses, de remplacer le tableau accroché au mur avec plusieurs papiers autocollants qui fournissent la planification de ressources et de tâches hebdomadaires.

L'utilisation des outils informatiques, dans les projets Agile du domaine du logiciel, est plus adéquate pour gérer un projet logiciel délocalisé, c'est-à-dire, des projets où les coéquipiers sont localisés à plusieurs endroits. Le support des outils sur le web permet d'avoir un point de convergence et de direction central commun.

Du côté de l'ingénierie et de la construction, étant donné que les projets sont bien localisés sur un site spécifique, tel qu'un bâtiment ou un pont, l'équipe de projet est plutôt localisée à un endroit et les outils sont aussi centralisés. Cependant, il existe de recherches en cours pour incorporer la méthode Agile dans la gestion de projets traditionnelle du domaine de la construction.

Mots-clés : gestion de projets, gestion de projets en construction, gestion de projets logiciels, méthodes Agile, outils de gestion.

AGILE METHOD COMPARISON AND ONLINE PROJECT MANAGEMENT TOOLS WITH METHODS AND TOOLS OF ENGINEERING DOMAIN

Mauricio LOPES

ABSTRACT

This exploratory research aims to study and compare methods and online software development project management tools versus traditional project management tools used by construction and engineering projects where I've been working for fifteen years.

The methods used by the project teams, in the field of engineering and construction, are guided by the concepts of the "Project Management Book of Knowledge" (PMBOK), created by the "Project Management Institute" (PMI). Tools such as MS-Project and Oracle Primavera P6, are commonly used for planning, monitoring and controlling the schedule, budget and project resources in the construction industry.

The current software development project management tools are geared toward the application of Agile methods. They recently appeared on the market and are available directly on the web, in the Cloud. Tools such as Jira-Agile and VersionOne are based on this trend and include an Agile SCRUM methodology for planning the iterative "sprints", to manage tasks and resources. These tools try, among other things, to replace the physical wall and stickers technique that is used for resource planning and weekly tasks allocation.

The use of computer tools, in Agile software projects, is well suited to manage a distributed software project, meaning projects where teammates are distributed across the world. Support tools on the web allow a point of convergence and common central management.

On the engineering and construction side, as the projects are located on a specific site, such as a building or a bridge, the project team is mainly located in one place and tools are also centralized. However, there is ongoing research to incorporate Agile methods in the traditional projects management of the construction industry.

Keywords: project management, construction industry project management, software project management, Agile methods, management tools.

TABLE DES MATIÈRES

P.

INTRODUCTION 1

CHAPITRE 1 La gestion de projets.....5
1.1 Gestion de projets traditionnelle 5
1.2 Conclusion 8

CHAPITRE 2 Gestion de projets logiciels 9
2.1 La gestion de projets (secteur développement de logiciel).....9
2.2 Les cycles de vie 9
2.3 Les normes du domaine 12
2.4 Conclusion 13

CHAPITRE 3 Méthode Agile.....14
3.1 La méthode Agile.....14
3.2 Les forces et les faiblesses de la méthode Agile.....14
3.2.1 Les méthodes Agile les plus utilisées 16
3.3 Le SCRUM 16
3.3.1 Définition des étapes et de rôles du cycle de vie SCRUM 19
3.4 Conclusion 20

CHAPITRE 4 La comparaison des approches.....21
4.1 Les outils de gestion.....21
4.2 Comparaison 23
4.2.1 Outils dédies aux projets traditionnels..... 23
4.2.2 Les outils dédiées aux projets logiciel 24
4.2.3 Par rapport l'application 29
4.2.4 Comparaison des outils 30
4.2.5 Le processus de mise à jour de projets (ingénierie versus logiciel)..... 31
4.3 L'impact des outils sur le succès des projets 33
4.3.1 Définition de succès des projets..... 33
4.3.2 Le lien entre les outils de gestion et le succès de projets..... 34
4.3.3 De bonnes pratiques à utiliser 34

CONCLUSION 35

LISTE DE RÉFÉRENCE BIBLIOGRAPHIQUE 37

LISTE DES TABLEAUX

P.

Tableau 3.1 Les méthodes Agiles les plus utilisées par la communauté de développement de logiciels…..……16

Tableau 3.2 Les outils logiciels les plus utilisés pour appuyer la méthode Agile……..18

Tableau 4.1 Comparaison entre les outils de gestion de projets traditionnels versus logiciels…….32

Tableau 4.2 Comparaison entre les données qui doivent être mises à jour dans un projet d'ingénierie versus un projet logiciel……...34

LISTE DES FIGURES

P.

Figure 2.1 L'évolution de méthodes et de cycle de vie……09

Figure 2.2 Cycle de vie du type « Waterfall »……..10

Figure 2.3 Le cycle de vie « Iterative »…….12

Figure 3.1 La vue d'ensemble de la méthode SCRUM……19

Figure 3.2 Exemple de la liste « product backlog »…….19

Figure 4.1 Tableau SCRUM……..22

Figure 4.2 Graphique de Gant……...23

Figure 4.3 Plan de ressources…….24

Figure 4.4 L'utilisation des outils selon le type de localisation de l'équipe….…….25

Figure 4.5 Exemple de tableau SCRUM……...26

Figure 4.6 Exemple de graphique « Burndown »……..27

Figure 4.7 Analyse du graphique « Burndown »……...27

Figure 4.8 Exemple de tableau SCRUM Jira Agile……...28

Figure 4.9 Exemple de graphique « Burndown » Jira Agile…….29

Figure 4.10 Exemple de « Cloud computing »……..30

INTRODUCTION

Avec plusieurs années d'existence, la gestion de projets représente une discipline très importante et est appliquée dans divers secteurs. Les entreprises privées et publiques utilisent de plus en plus le concept de gestion par projets, car la pression pour obtenir de bons résultats, c'est-à-dire, la livraison d'un produit ou d'un service de qualité, dans les délais et les coûts prévus, augmente continuellement. Il faut donc toujours faire plus avec moins d'argent, en utilisant les ressources de manière intelligente, dans un délai de plus en plus court pour les clients et vis-à-vis la compétition.

La méthodologie de gestion de projet décrite dans le Project Management Book of Knowledge (PMBOK), créée par le Project Management Institute (PMI), est la plus populaire actuellement en Amérique du Nord. Cette méthodologie a fait ses preuves, surtout dans des secteurs structurés comme l'ingénierie et la construction. Cependant, il existe toujours de la place pour des améliorations.

Malgré la bonne utilisation de méthodologies, le dévouement des équipes de projets et des autres parties prenantes, il existe encore des projets qui dépassent leurs coûts estimés, le délai accordé et la portée initiale. Selon une étude réalisée par Flyvbjerg et Budzier, en 2011, à l'université de commerce d'Oxford en Angleterre, 1471 projets du secteur des technologies de l'information (TI) ont comporté des dépassements quelconques. Selon cette étude, pour chaque six projets étudiés, un avait soit un échec avec de dépassements de coûts de plus de 200 % ou des dépassements de délai de plus de 70 %. (Flyvbjerg et Budzier, 2011) Cette étude présente aussi, des exemples concrets de situation de dérapage dans l'exécution de projets, par exemple :

> L'utilisation de versions différentes du logiciel de développement du produit par les succursales de Boeing en Allemagne, en Espagne, en Angleterre et en France a causé un délai de six mois pour la première livraison du modèle d'avion A380. Ce dépassement a engendré une réduction de 26% du prix des actions de l'entreprise. (Flyvbjerg et Budzier, 2011)

Lors d'un dérapage de ce genre, il faut absolument être capable de comprendre pourquoi il y a un dépassement, le corriger rapidement et surtout apprendre comment l'éviter à postériori.

Afin de tenter de réduire les dérapages, et de livrer des logiciels plus rapidement et qui satisferont les besoins du client, le «Manifesto Agile» a été lancé en 2001. (Kent Beck et al, 2001) Cette nouvelle méthodologie de cycle de vie de logiciel propose une rupture avec la gestion de projets traditionnelle qui se concentre davantage sur la documentation contractuelle et le processus au lieu de développer une relation avec le client, tout en livrant des produits finis à chaque itération de développement. Le Manifesto Agile propose aussi des valeurs et des principes, mais ne décrit pas précisément comment gérer un projet logiciel au jour le jour. C'est à partir de ce Manifesto que plusieurs méthodologies Agile sont issues. Des exemples de ces méthodologies seront présentées dans ce rapport.

Les entreprises doivent choisir une méthode de développement de logiciel, par exemple une méthode Agile, à laquelle elles s'identifient, tout en prenant en considération leurs particularités: l'équipe et le type de projet. En conséquence, une entreprise peut utiliser plus d'une méthode pour combler des besoins différents.

En 2011, un sondage a été commandé par l'entreprise de télécommunication Ericsson afin de connaître quelle méthode Agile et quels outils étaient les plus utilisés parmi les entreprises du même secteur. Les responsables du sondage ont reçu 121 réponses provenant de 120 entreprises réparties parmi 35 pays. Les résultats de ce sondage ont démontrés que la méthodologie Agile SCRUM était la plus utilisée et que le mur et les papiers autocollants étaient aussi une technique très populaire. (Azizyan, Magarian et Kajko-Matsson, 2011) Les données complètes de ce sondage seront présentées à la section 3.3 de ce rapport.

Un autre sondage, mené par un fournisseur d'outil de gestion de projets en ligne sur le web, confirme que le SCRUM est la méthode Agile la plus utilisée. Le mur et les papiers autocollants sont aussi utilisés par la majorité de répondants. (VersionOne Inc, 2014)

L'objectif principal de cette recherche est de comparer deux approches de gestion de projets : 1) une approche qui utilise les outils développés exclusivement pour la gestion de projets traditionnels, tels que les secteurs de l'ingénierie et de la construction, et 2) les outils logiciels émergents, disponibles sur le web, du domaine de la gestion de projet du logiciel qui appuient l'approche SCRUM.

Cette recherche tente de répondre à la question suivante : quels sont les outils choisis par les entreprises pour la gestion de projets selon la localisation et la taille de l'équipe de projet? Les recherches, à ce jour, ne sont pas concluantes concernant l'impact, d'une ou l'autre des techniques sur le facteur de succès d'un projet.

CHAPITRE 1 La gestion de projets

1.1 Gestion de projets traditionnelle

La discipline de gestion de projets, connue depuis plus de 64 ans, vit une véritable croissance auprès des entreprises privées et publiques, grâce à plusieurs facteurs tels que: les changements dans l'économie mondiale, les demandes plus exigeantes du marché et leur besoin d'être profitable.

Les années 1950 marquent le début de l'utilisation systématique de la gestion de projets, notamment, par l'armée navale américaine et dans les années suivantes par le département de défense des États-Unis (DoD). (Carayannis, Kwak et Anbari, 2005)

À cette époque, l'utilisation de la gestion de projets s'avérait de plus en plus nécessaire dans tous les domaines, car la certification des gestionnaires de projets devenait de plus en plus demandée. Les outils disponibles pour gérer un projet et calculer les échéanciers étaient principalement les techniques du Gantt, du chemin critique (CPM) et du Program Evaluation and Review Technique (PERT). (Carayannis, Kwak et Anbari, 2005)

Dans les années 1960, suite à l'introduction et la popularité grandissante des microordinateurs, ces outils ont pu se déployer davantage, en permettant la réalisation de calculs de plus en plus complexes, car ils étaient automatisables. Ces techniques, aujourd'hui, sont intégrées à des outils logiciels de plus en plus puissants et plus complets tels que : MS-Project et Primavera P6, pour ne nommer que ceux-ci. Ils permettent de calculer plusieurs paramètres critiques d'un projet, comme par exemple, l'allocation de ressources. (Microsoft Corporation, 2015)

Les organisations cherchaient une nouvelle méthodologie qui comportait des techniques et une structure organisationnelle plus formelle, afin de gérer des projets qui permettent de livrer le produit selon le budget, le coût et tenant compte de l'envergure croissante des projets. La « National Aeronautics and Space Administration » (NASA) a créé alors son bureau du

programme Apollo avec la mission de gérer et planifier les missions Apollo en utilisant le PERT; gérer l'approvisionnement et l'octroi des fournisseurs tel que « General Electric» (GE); mettre en place un système de gestion pour mesurer la performance des projets, et mettre en place une gestion centralisée du programme Apollo. (Carayannis, Kwak et Anbari, 2005)

Pour consolider la croissance et l'évolution de la gestion de projets, deux organismes internationaux ont alors été fondés. En1967, l'International Project Management Association (IPMA) en Europe, et deux années après, en 1969, suit la création du Project Management Institute (PMI) aux États-Unis. Ces deux organisations fournissent des méthodologies de gestion de projets. Elles proposent aussi des certifications professionnelles pour leurs membres qui doivent rencontrer une série d'exigences et qui doivent réussir un examen de certification des connaissances. La certification « Project Management Professional » PMP est très reconnue en Amérique du Nord et depuis 2011, le PMI offre aussi une certification Agile « Agile Certified Practitioner » PMI-ACP.

Dans le cadre de la gestion de projets PMBOK (c'est-à-dire le PMBOK 5e édition) la gestion d'un projet est découpée en dix domaines de compétences, cinq groupes de processus et 47 processus de gestion de projets. Le concept derrière cette méthodologie est de simuler l'exécution détaillée du projet avant de le faire réellement, autrement dit, « *Every moment spent planning saves three or four in execution* ». (Wysocki, 2014)

Une autre définition proposée par le PMBOK concerne le cycle de vie d'un projet qui possède quatre étapes (Project Management Institute, 2013):

- Démarrage du projet;
- Organisation et préparation;
- Réalisation des travaux;
- Clôture du projet.

Typiquement, dans le domaine de l'ingénierie et de la construction, le cycle de vie est découpé en cinq étapes: faisabilité, planification, design, exécution et mise en service.

Contrairement au cycle de vie d'un projet, la méthodologie de gestion de projets sert à gérer le projet. Ces deux concepts de gestion de projet peuvent porter à confusion, car les deux sont nécessaires et utiles. La différence entre eux est que ce dernier décrit comment gérer un projet et le cycle de vie sert à décrire, de manière détaillée, le travail à faire. (Mulcahy, 2005) La méthodologie de gestion de projets vise donc la planification, l'organisation, la gestion de ressources humaines et des parties prenantes, entre autres.

Les recommandations du PMBOK orientent les chargés de projets, qui normalement sont des professionnels de la gestion de projets (parfois même certifiés par le PMI), à initier un projet seulement quand tous les plans de gestion sont définis et approuvés par toutes les parties prenantes du projet, ces plans sont:

- Plan de gestion de la communication;
- Plan de gestion de coûts;
- Plan de gestion de ressources humaines;
- Plan de gestion de l'approvisionnement;
- Plan d'amélioration;
- Plan de gestion la qualité;
- Plan de gestion de besoins;
- Plan de gestion de risques;
- Plan de gestion de l'échéancier;
- Plan de gestion de la portée;
- Plan de gestion de parties prenantes;
- La référence des coûts;
- La référence de l'échéancier;
- La référence de la portée;
- Plan de gestion de la mise à jour;

La gestion de projet n'est pas une tâche facile à réaliser dans un monde qui demande des actions, des progrès et des résultats de plus en plus rapides.
La réussite des projets dépend de la bonne application de la méthode, mais aussi d'autres facteurs très importants comme la composition de l'équipe du projet, l'expérience du chargé de projet, une stratégie de communication efficace, l'identification et la gestion de risques, et bien d'autres facteurs.

1.2 Conclusion

Cette première revue littéraire a présenté une synthèse de l'évolution de la gestion de projets, plus traditionnelle, au fil des ans et a aussi présenté quelques concepts importants pour comprendre cette discipline. Le chapitre suivant fait la synthèse de l'évolution de la gestion de projets dans le domaine du développement des logiciels.

CHAPITRE 2 Gestion de projets logiciels

2.1 La gestion de projets (secteur développement de logiciels)

Cette deuxième revue littéraire est réalisée du point de vue de quelqu'un expérimenté dans la gestion de projets traditionnels qui vise établir une comparaison avec les nouvelles méthodes Agile et ses outils de gestion de projets en ligne dont il n'a aucune connaissance préalable. Il s'agit donc d'une comparaison avec les méthodes et les outils de gestion de projets du domaine de l'ingénierie qui sont plus maitrisées par l'auteur de ce rapport.

Au tout début de l'industrie du logiciel, la méthode utilisée pour développer un nouveau logiciel était chaotique et non structurée. Ce n'est qu'en 1930, que monsieur Willian Edwards Deming a publié la méthode « Plan-Do-Study-Act » (PDSA), qui par la suite, est devenue très connue dans le domaine de la qualité. C'est à cette même époque que les cycles de vie sont apparus et ont été appliqués pour développer des logiciels. (Larman et Basili, 2003)

La Figure 2.1 présente l'évolution de méthodes et de cycles de vie du développement de logiciel de 1986 à 2001.

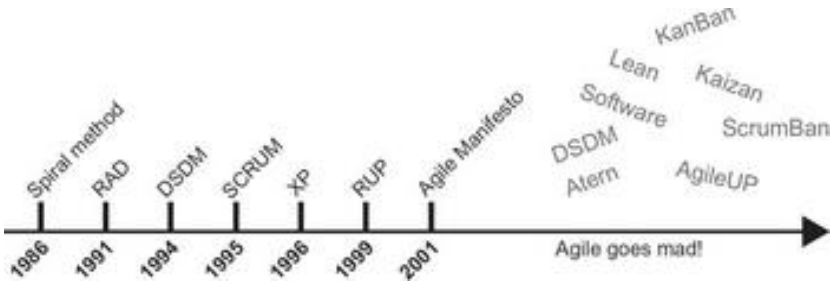

Figure 2.1 L'évolution de méthodes et de cycles de vie (Cadle, 2014)

2.2 Les cycles de vie

Les cycles de vie de développement de logiciels sont principalement divisés en deux groupes : 1) les cycles de vie linéaires et 2) les cycles de vie évolutionnaires.

Les cycles de vie linéaires sont, entre autres, le « Waterfall », l'« Incremental », et le cycle de vie « en V ». Ces approches visent la réalisation d'une étape après l'autre. (Khan et Beg, 2013)

La Figure 2.2 décrit un exemple du cycle de vie « Waterfall », utilisée par l'industrie du logiciel.

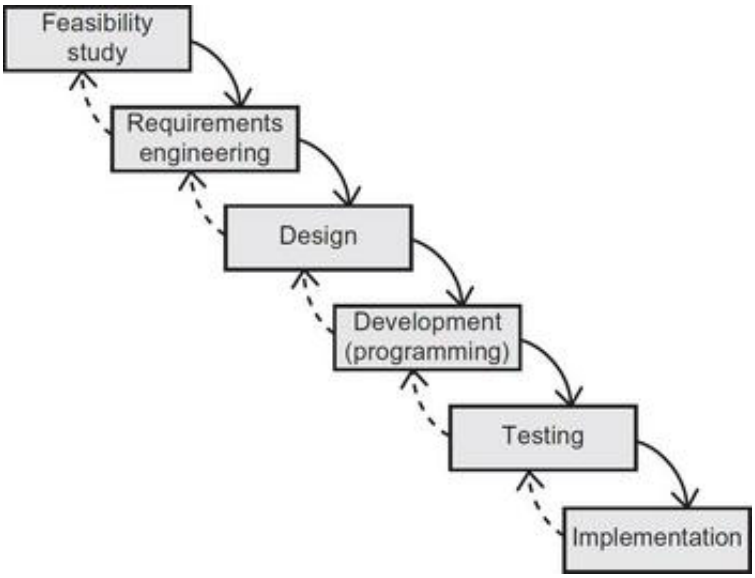

Figure 2.2 Cycle de vie du type «Waterfall »
(Cadle, 2014)

Ce cycle de vie s'applique particulièrement bien à de projets où les besoins et les requis sont connus dès le début et la probabilité de changements est réduite. Avec l'utilisation de ce cycle de vie, la livraison et l'essai se feront à la fin seulement, ce qui pourrait être trop tard pour certains clients ou avoir changé entretemps.

L'utilisation de méthode linéaire vise les projets plus stables, tels que l'ingénierie civile, qui utilise souvent la méthode: Ingénierie, Approvisionnement et Gestion de la construction (IAGC), qui se ressemble beaucoup au « Waterfall ».

Ces méthodes linéaires, de développement, ont été mises en question au début des années 70, surtout à cause de la rigidité de la séquence des étapes. À cette époque,

Winston Royce partage ses expériences concernant l'utilisation de la méthode « Waterfall » et propose des améliorations comme l' «Interactive and Incremental Development life cycle method» (IID). (Larman et Basili, 2003)

À l'époque, un des grands donneurs d'ordre de cette industrie était le Département de la Défense (DoD) des États-Unis, qui avait comme norme recommandée la méthode de développement de logiciels en cascade c'est-à-dire le «Waterfall», ce qui a ralenti considérablement l'utilisation et l'introduction de nouvelles approches. (Larman et Basili, 2003)

Le résultat de l'utilisation de cette norme durant les années 80, aboutit, selon Larman et Basili, en plusieurs échecs, par exemple : le produit final ne reflétait plus les besoins des utilisateurs, conséquemment il était devenu nécessaire de refaire une grande partie du logiciel. C'est seulement en 1987 que le DoD a autorisé l'utilisation d'autres cycles de vie de développement logiciel comme l'IID. En 1999, le DoD exprime dans un rapport que « d'un total de $ 37 milliards de dollars US, 75 % des projets logiciels ont échoué ou n'ont été jamais utilisés… ». (Larman et Basili, 2003)

Les cycles de vie évolutionnaires sont, entre autres, « Iterative », « Spiral », SCRUM et « Agile Unified Process » (Agile UP). (Cadle, 2014) Ce type de cycle de vie vise la livraison fonctionnelle de plusieurs versions du produit après chaque itération. La Figure 2.3 décrit le cycle de vie itératif, introduit vers les années 1980.

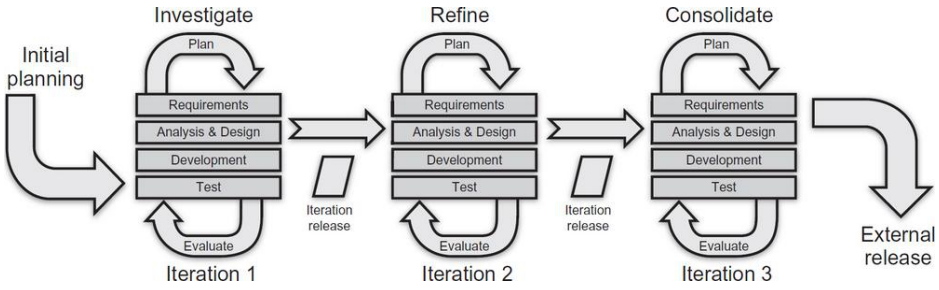

Figure 2.3 Le cycle de vie itératif (Cadle, 2014)

À la figure 2.3 nous remarquons que l'itération 1 ou autrement dit l'investigation est utilisée pour réduire les risques. L'itération 2 ou le raffinement est utilisé pour définir les besoins et il peut se répéter plusieurs fois et l'itération 3 ou la consolidation est utilisée pour assurer que toutes les fonctionnalités ont été intégrées avant la livraison externe du projet. Ce cycle de vie est normalement combiné avec l'incrémental afin d'éviter une seule livraison à la fin. (Cadle, 2014)

2.3 Les normes du domaine

Le domaine du développement de logiciel est guidé par plusieurs normes internationales. Deux normes ont été choisies pour leurs liens avec le sujet de recherche :

1) La norme ISO/IEC 12207-2008 établit « un cadre commun pour le processus de cycle de vie du logiciel, avec une terminologie bien définie, qui peut être référencée par l'industrie du logiciel ». (International Organization for Standardization, 2008)

2) Plus récemment, en 2011, une autre norme a été créée dans le but d'adapter la norme ISO/IEC 12207 pour les petites entreprises du domaine de développement de logiciel, il s'agit de la norme ISO/IEC 29110. On peut penser ici à une version Agile de la norme de cycles de vie du logiciel. L'idée est de créer une norme qui permettra de standardiser la documentation et la communication au sein des entreprises, sans l'attacher obligatoirement à une méthodologie Agile spécifique. (International Organization for Standardization, 2011)

Malgré la possibilité d'obtenir de bons résultats suite à l'adoption de ces cycles de vies, l'acceptation, par les entreprises commerciales, est encore faible à cause de la méconnaissance et du coût perçu de l'implantation de ces normes. Possiblement, les entrepreneurs ne perçoivent pas le bénéfice à long terme, mais, seulement l'investissement qu'ils doivent faire à court terme. En effet, l'histoire se répète de la même façon lorsque les normes ISO sont sorties vers les années 80. Au départ, seu-

lement les grandes entreprises avaient le moyen pour investir dans l'assurance qualité, au fil des ans, avec la popularisation et la disponibilité de ressources le nombre de petites et moyennes entreprises certifiées a beaucoup augmenté.

2.4 Conclusion

Dans ce chapitre, nous avons fait une courte synthèse de la littérature concernant l'historique de la gestion de projets logiciel, l'importance des cycles de vie pour ce type de projet et deux normes du domaine. Le prochain chapitre présente une synthèse d'une méthode de développement Agile, la méthode SCRUM qui est la plus utilisée actuellement dans l'industrie du logiciel.

CHAPITRE 3 Méthode Agile

3.1 La méthode Agile

Les méthodes de développement Agile sont devenues populaires dès 2001, lors de la rencontre d'experts du domaine, qui visait la promotion de l'utilisation de ces méthodes. (Larman et Basili, 2003)

Cette rencontre produit le Manifesto Agile qui préconise les valeurs suivantes (Kent Beck et al, 2001)

- Les individus et leurs interactions plus que les processus et les outils;
- Des logiciels opérationnels plus qu'une documentation exhaustive;
- La collaboration avec les clients plus que la négociation contractuelle;
- L'adaptation au changement plus que le suivi d'un plan.

Les éléments principaux qui différentient l'approche Agile de l'approche linéaire sont la livraison fonctionnelle du produit à chaque itération, la participation du client, en tout temps pendant la réalisation du projet, la force de l'équipe et de techniques utilisées par la méthode SCRUM. (Agile Alliance, 2015)

3.2 Les forces et les faiblesses de la méthode Agile

Les forces de ces méthodes ou les raisons le plus fortes pour l'adopter, selon les résultats d'une enquête menée en 2013, sont notamment le temps réduit pour arriver au marché (c.-à-d. délai de livraison) (32 %), la facilité de changer les priorités (27 %), un meilleur alignement entre la technologie de l'information (TI) et les affaires (23 %) et l'augmentation de la productivité (19 %). Ce sondage permettait aux répondants de choisir différentes réponses. (VersionOne Inc, 2014)

Ce sondage démontre aussi que les entreprises, qui utilisent une approche Agile, ont améliorés leur gestion du changement, leur productivité, leur visibilité des projets et le moral de leurs équipes de projet. (VersionOne Inc, 2014)

En contrepartie, les faiblesses observées, sont : l'absence d'une planification préliminaire (30 %), la perte de contrôle (30 %), la résistance à l'utilisation d'une nouvelle méthode (28 %) et la faible documentation produite (24 %). (VersionOne Inc, 2014)

Concernant la planification préliminaire (c'est-à-dire, l'estimation en heures et en argent), il est rapporté qu'il est possible de la faire en tenant compte des éléments connus et en ajoutant une marge de sécurité pour les changements. La gestion de cette marge de sécurité reste entre les mains du chargé de projets et/ou du commanditaire du projet.

Concernant la perte de contrôle et l'opposition à la méthode, cette problématique vient, surtout des gestionnaires qui ne sont pas impliqués dans le projet, car l'équipe de projets, a un grand pouvoir sur les tâches à accomplir entre les itérations.

Sachant qu'un des plus gros problèmes qui mène les projets de logiciels à un échec provient des définitions de besoins du projet, un des prérequis de la méthodologie Agile est la présence constante du « product owner » qui est le représentant des utilisateurs, ce qui demande beaucoup d'implication de la part du client.

L'impact de l'adoption de méthodologies Agile, se fait ressentir partout dans l'entreprise, car même les départements qui ne font pas partie du projet doivent se familiariser avec les nouveaux principes de la méthodologie pour expliquer aux clients externes, par exemple dans le cas du département de ventes, cela demandera un changement de culture.

3.2.1 Les méthodes Agile les plus utilisées

Les méthodes Agile les plus populaires pour le développement de logiciels sont actuellement le SCRUM, le XP et le Custom hybride. (VersionOne Inc, 2014) Cela a été confirmé par le résultat d'un sondage réalisé en 2011. Dans ce sondage, la méthodologie SCRUM a été la plus citée de toutes les méthodes Agiles. C'est donc celle qui est la plus utilisée par la communauté de développement de logiciels. (Azizyan, Magarian et Kajko-Matsson, 2011) C'est pour cette raison que ce rapport se basera sur cette méthode.

3.3 Le SCRUM

L'histoire du SCRUM a commencé en 1986 avec Messieurs Takeuchi et Nonaka suite à la publication de l'article «The New New Product Developement Game ». Par la suite, en 1990, Jeff Sutherland et Ken Schwaber ont publié l'article « SCRUM Software Developement Process ».(Schwaber et Sutherland, 2013)

Le SCRUM se définit comme: « un cadre de travail permettant de répondre à des problèmes complexes et changeants, tout en livrant de manière productive et créative des produits de la plus grande valeur possible. » (Schwaber et Sutherland, 2013)

Les résultats obtenus en 2011 par l'équipe de la firme Ericsson concernant les méthodes et les outils les plus utilisés par cette industrie sont présents aux tableaux suivants.

Tableau 3.1 – Les méthodes Agiles les plus utilisées par la communauté de développement de logiciels. (Azizyan, Magarian et Kajko-Matsson, 2011)

Méthodologie Agile	Pourcentage des répondants
SCRUM	54%
SCRUM – XP	32%
XP	11%

Kanban	9%
TDD	2,5%
Crystal	2,5%
DSDM	2%
FDD	2%

Les auteurs de ce sondage ont observé que la somme des pourcentages était de 114 %. Cela confirme que certaines entreprises sondées utilisaient plus d'une méthode Agile pour combler leurs besoins de gestion de projets. (Azizyan, Magarian et Kajko-Matsson, 2011)

Chaque méthode possède son orientation propre envers le développement d'un produit logiciel. Par exemple, l'« Extreme Programming » (XP) est orienté vers les pratiques de programmation, pendant que l'approche SCRUM est orientée vers la gestion d'itérations successives d'un projet au complet. En effet, on peut voir que ces deux méthodes peuvent être complémentaires, car le SCRUM n'aborde pas les pratiques de programmation directement.

Les méthodes Agiles ont aussi une orientation selon la grandeur et la complexité du projet à réaliser. Par exemple, un des moyens de mesurer la taille d'un projet est via la quantité de « stories points » qui reflètent le temps (ou l'effort) estimé par l'équipe du projet pour réaliser une « story » qui représente une (ou plusieurs regroupées) fonctionnalité demandée par le client. De plus, la méthode joue un rôle très important selon la taille de l'équipe de développement et sa localisation.

Ce même sondage permet d'avoir une idée des outils de gestion de projets logiciels les plus utilisés, pour appuyer la méthode Agile choisie par une entreprise.

Tableau 3.2 – Les outils logiciels les plus utilisés pour appuyer la méthode Agile (Azizyan, Magarian et Kajko-Matsson, 2011)

Outil	Pourcentage
Le mur et les autocollants	26%
Le chiffrier électronique	23%
MS-Project	8%
Outils maison	5%
Rally	5%
Trac	4%
Mingle	3%
Scrumworks	2%
VersionOne	2%
Jira	2%
MS Team Foundation Server	2%
XPlanner	2%
Assembla	1%
Autres	12%

La somme des pourcentages des neuf outils dédiés aux projets logiciels résulte en 23 %. Cela démontre que le marché est en train d'expérimenter ces outils tout en gardant les outils qui sont déjà connus des équipes comme le mur et les autocollants (26 %) et le chiffrier électronique (23 %). Concernant le MS-Project (8 %), il se présente comme une option en parallèle, car il est plus utilisé par les projets traditionnels. Les outils maison (5 %) et les autres outils (12 %) représentent 20 utilisateurs, sur les 121 qui ont répondu au sondage, que pour de raisons particulières utilisent des outils peu connus sur le marché.

En fonction de la disponibilité de données et de références du marché, ce rapport basera sa comparaison sur les outils Jira-Agile et VersionOne.

3.3.1 Définition des étapes et de rôles du cycle de vie SCRUM

La Figure 3.1 décrit un exemple du cycle de vie SCRUM :

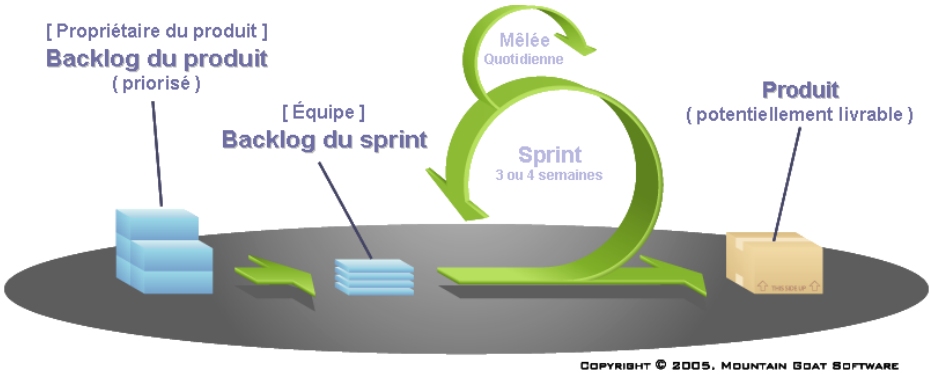

Figure 3.1 La vue d'ensemble de la méthode SCRUM (Wikipédia l'encyclopédie libre, 2015)

Voici comment fonctionne ce cycle de vie. Au début du projet, un analyste de besoin d'affaires rencontre le client pour comprendre ses besoins. Par la suite, il documente cette information à l'aide d'un outil informatisé et cette information servira de « product backlog » qui est la liste d'exigences du projet. La Figure 3.2 présente un exemple de « product backlog » :

ToDo List

ID	Story	Estimation	Priority
7	As an unauthorized User I want to create a new account	3	1
1	As an unauthorized User I want to login	1	2
10	As an authorized User I want to logout	1	3
9	Create script to purge database	1	4
2	As an authorized User I want to see the list of items so that I can select one	2	5
4	As an authorized User I want to add a new item so that it appears in the list	5	6
3	As an authorized User I want to delete the selected item	2	7
5	As an authorized User I want to edit the selected item	5	8
6	As an authorized User I want to set a reminder for a selected item so that I am reminded when item is due	8	9
8	As an administrator I want to see the list of accounts on login	2	10
Total		**30**	

Figure 3.2 Exemple de la liste « product backlog » (International Scrum Institute, 2015)

Le propriétaire du produit doit prioriser cette liste afin de créer le « backlog du sprint » en lien avec la capacité et la performance de l'équipe. Dès que la liste est acceptée par l'équipe,

elle ne change plus jusqu'à la fin du « sprint » qui est un cycle de développement d'environ 7 à 10 jours. Des rencontres journalières sont tenues, sous la direction du « SCRUM Master », qui a comme rôle la facilitation des activités du projet.

3.4 Conclusion

Ce chapitre a présenté une synthèse de la littérature concernant les méthodologies Agiles, leurs forces et faiblesses, ainsi qu'une introduction à la méthode SCRUM. Le chapitre suivant, qui est central au projet de recherche, présente la comparaison entre deux approches de gestion de projets : 1) une approche qui utilise les outils développés pour la gestion de projets traditionnels, comme l'ingénierie et de la construction, et 2) les outils logiciels émergents, disponibles sur le web, du domaine de la gestion de projet du logiciel qui appuient l'approche SCRUM.

CHAPITRE 4 La comparaison des approches

4.1 Les outils de gestion

En faisant un parallèle avec l'ingénierie, les outils de gestion et de planification les plus utilisés sont MS-Project de Microsoft et Primavera P6 d'Oracle. Ces outils permettent la planification et le contrôle de plusieurs aspects du projet, entre autres, le séquencement des activités, la planification des ressources, la budgétisation de coûts et les efforts nécessaires pour livrer le projet à temps et à l'intérieur du budget accordé. À la fin du processus de planification, l'équipe de gestion du projet pourrait utiliser de techniques de contrôle disponibles par l'outil, par exemple, le chemin critique, le nivèlement de ressources et l'analyse de valeur acquise, qui sont fortement utilisés par l'industrie de la construction. (Liberatore, Pollack-Johnson et Smith, 2001)

Dès que le projet débute, un suivi régulier est incontournable. La mise à jour régulière de données est fondamentale pour corriger un dépassement soit de coûts soit de délais. Par la suite, l'utilisateur doit interpréter et analyser cette information pour dresser un portrait précis de la « santé » du projet. Des rapports et des graphiques sont typiquement générés pour une meilleure communication de l'information aux gestionnaires. Cette information est consolidée dans un tableau de bord qui sera présenté au comité de pilotage du projet à une fréquence déterminée.

Du côté développement de logiciels, les outils/techniques peuvent être disponibles localement ou sur le Web. Les techniques locales, telles que le tableau et les papiers autocollants, sont les plus utilisées, tel que présenté au chapitre précédent. Cet outil important de planification Agile facilite la visualisation de qui travaille sur quoi et il s'adapte à la vitesse de changements du projet.

La Figure 4.1 démontre un exemple du tableau SCRUM en question :

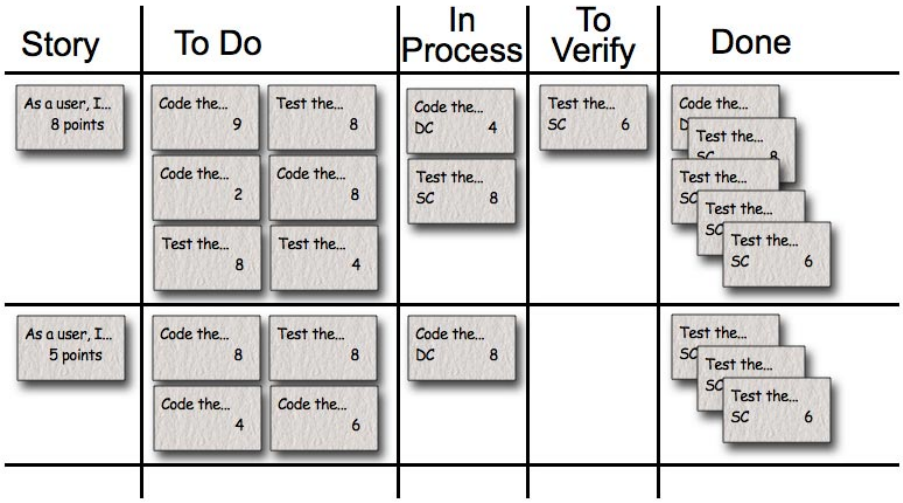

Figure 4.1 Tableau SCRUM Tirée du site (Montain Goat Software, [s.d.])

Ensuite, nous avons les outils sur le web. Il existe plusieurs outils disponibles sur le Web et les fonctionnalités typiques de base sont:

- La planification de « stories » et la correction de « bug »;
- La planification des « sprits »;
- Le suivi du projet;
- La gestion de tests;
- Les graphiques « burndown » et vitesse;
- Les rapports.

La prochaine section présente une comparaison des forces et faiblesses des outils consultés.

4.2 Comparaison

4.2.1 Outils dédies aux projets traditionnels

Tel que mentionné, ces outils, soit MS-Project soit Primavera P6, sont très puissants, car ils offrent une quantité importante de paramètres que l'utilisateur pourrait personnaliser afin de bien ajuster son projet. Deux entre eux sont, à mon avis, incontournables:

1. Gestion de baselines (références):

Cela nous permet de garder une photo du projet au moment de son départ et l'utiliser à des fin de comparaison de la performance au fur et à mesure qu'il avance. Les analyses de la valeur acquise sont tributaires de ce baseline.

La Figure 4.2 démontre un exemple d'un graphique de Gantt et le baseline associé au projet :

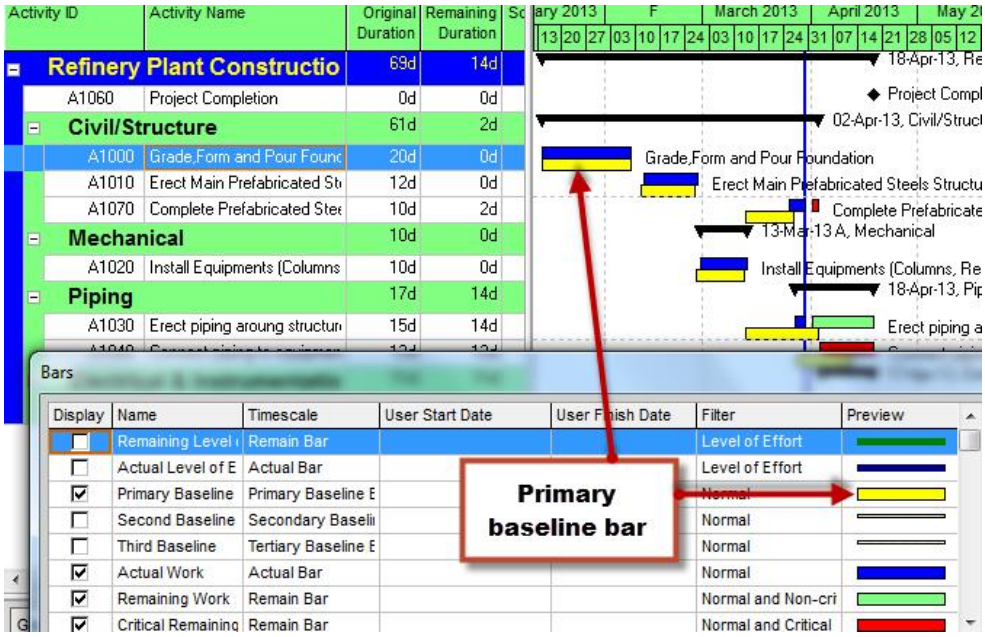

Figure 4.2 graphique de Gantt (Oo, [s.d.]-b)

2. Gestion de ressources

Les ressources sont sans doute une des grandes contraintes des projets. Savoir qui fait quoi et quand est important pour l'avancement du projet. Plus encore, cela nous permet de bien gérer la mobilisation et la démobilisation du personnel donc, il y a un impact non négligeable sur le coût de la main-d'œuvre.

La Figure 4.3 démontre un exemple de plan de ressources distribué par mois/semaine :

Resource Name	Budgeted Units	ugust 2009				S				October 2009				N				D			
		08	15	22	29	0	12	19	26	03	1	17	24	31	07	14	2	28	05	12	19	2
Total		32	72	81	90	56	06	01	41	08	25	15	99	62	95	29	00	53	23	75	29	20
Coded Welder		48	56	50	47	80	62	48	23	67	48	43	01	19	60	06	07	62	16	26	06	78
Labourer - Hot Work Sentry		24	15	00	52	74	01	66	45	47	21	20	63	94	00	07	52	15	58	94	35	35
Labourer - Trade Assistant		87	64	77	51	06	11	04	94	29	43	99	94	05	21	65	45	68	04	33	00	15
Pipe Fitter / Boilermaker		96	08	59	15	55	93	07	55	78	59	04	08	14	38	13	15	86	31	15	58	30
Rigger		76	30	94	25	42	40	76	24	86	53	49	33	31	76	39	81	23	15	8	31	62

Figure 4.3 Plan de ressources (Oo, [s.d.]-a)

Cet ensemble de données nous permet de créer un historique qui nous aidera à mieux planifier un nouveau projet.

4.2.2 Les outils dédiés aux projets logiciels

Les projets de développement de logiciels, contrairement aux projets d'ingénierie, sont souvent décentralisés, c'est-à-dire, l'équipe de projet est distribuée, parfois dans plusieurs pays, ce qui représente une difficulté additionnelle pour la gestion du projet soit, à cause du décalage horaire soit, à cause de la difficulté de mettre l'équipe ensemble dans la même salle. Ces outils visent, entre autres, à réduire cette barrière.

Selon le sondage mené par l'entreprise Ericsson en 2011, les outils sont utilisés par des équipes de projets tel que présentés à la Figure 4.4.

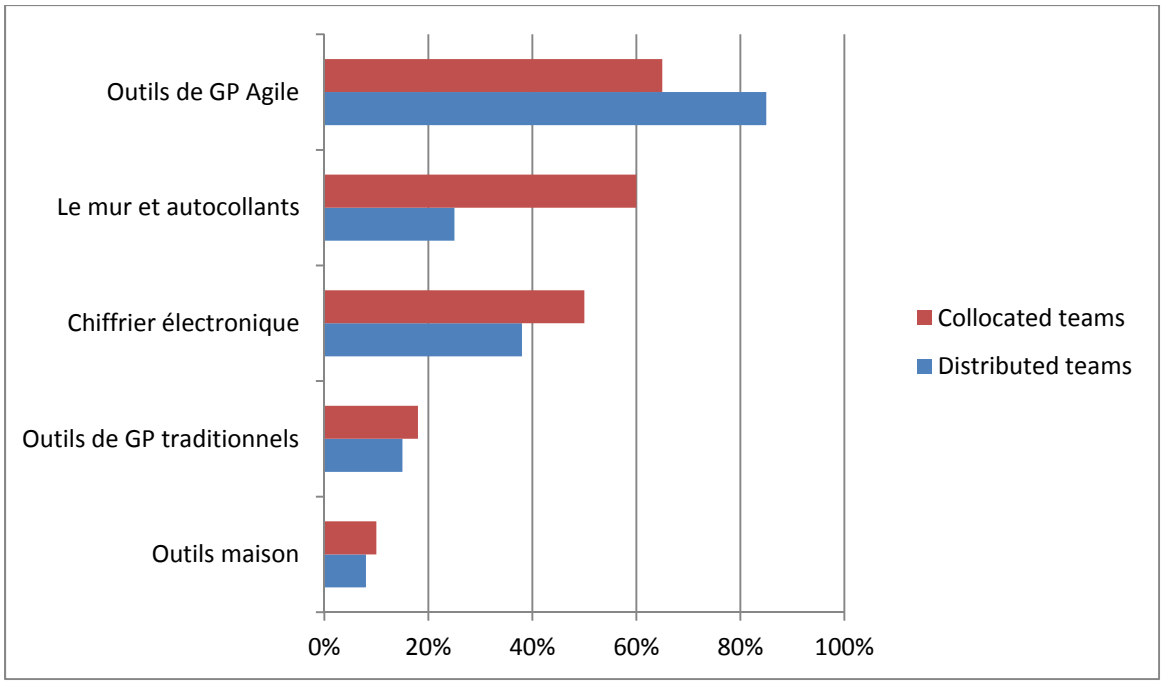

Figure 4.4 L'utilisation des outils selon le type de localisation de l'équipe (Azizyan, Magarian et Kajko-Matsson, 2011)

De plus, 31 % des répondants utilisent les deux types d'outils: Outils de GP Agile et le mur et autocollants, même si ses équipes sont centralisées à un endroit. Cela démontre qu'un outil simple et facile à utiliser comme le mur et l'autocollant sera difficile à remplacer. (Azizyan, Magarian et Kajko-Matsson, 2011)

Étant donné que le SCRUM est la méthodologie Agile la plus utilisée, les outils étudiés sont basés sur cette méthodologie. Les figures suivantes démontrent comment ces outils visent à remplacer les outils physiques.

La Figure 4.5 présente un exemple de tableau SCRUM créé par VersioOne :

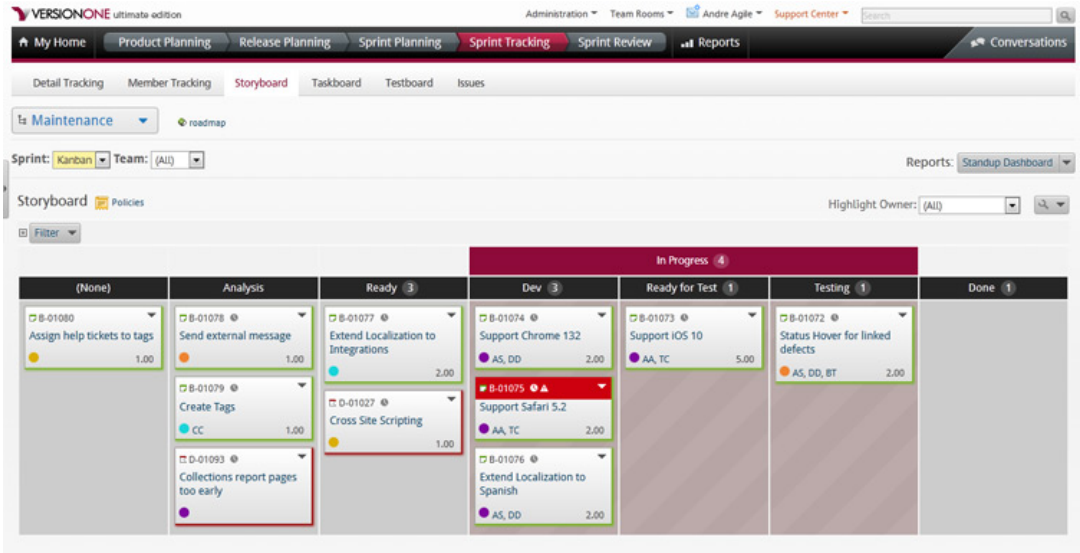

Figure 4.5 Exemple de tableau SCRUM (VersionOne Inc, 2015b)

Chaque membre de l'équipe peut actualiser l'avancement de ses tâches en temps réel, ce qui augmente la performance du groupe. La Figure 4.6 présente un exemple de graphique « burndown » créé par VersioOne.

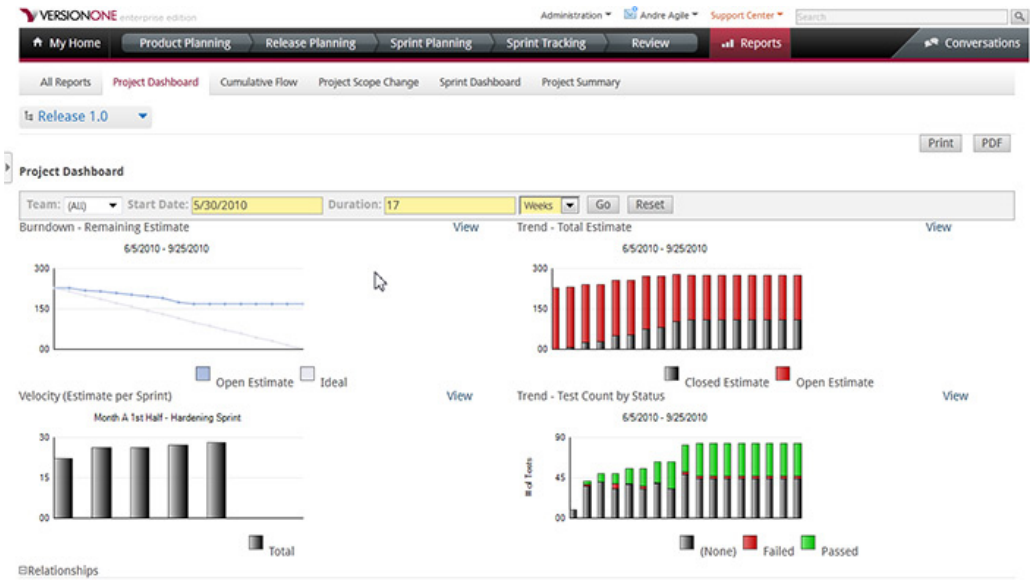

Figure 4.6 Exemple de graphique « Burndown » (VersionOne Inc, 2015a)

Le graphique burndown peut présenter trois situations par rapport à l'évolution du projet (voir Figure 4.7).

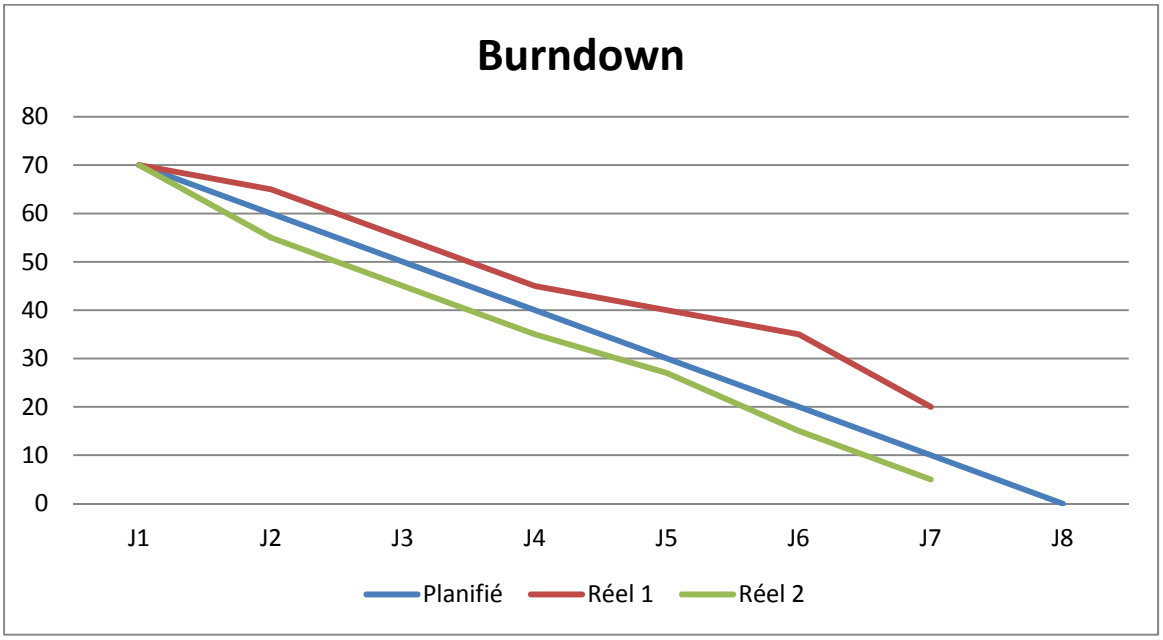

Figure 4.7 Analyse du graphique Burndown « adapté de » (Kniberg, 2007)

1. Les « stories points » réels suivent les « stories points » planifiés (courbe bleue)
2. Les « stories points » réels sont plus élevés que le planifié (courbe rouge)
3. Les « stories points » réels sont moins élevés que le planifié (courbe verte)

Dans la situation 1, le SCRUM master n'a aucune action à prendre. Par contre, pour les autres situations, il doit enlever de « stories points » pour la courbe rouge et ajouter de « stories points » pour la courbe verte. (Kniberg, 2007)

L'outil Jira Agile, présente son tableau SCRUM selon la Figure 4.8.

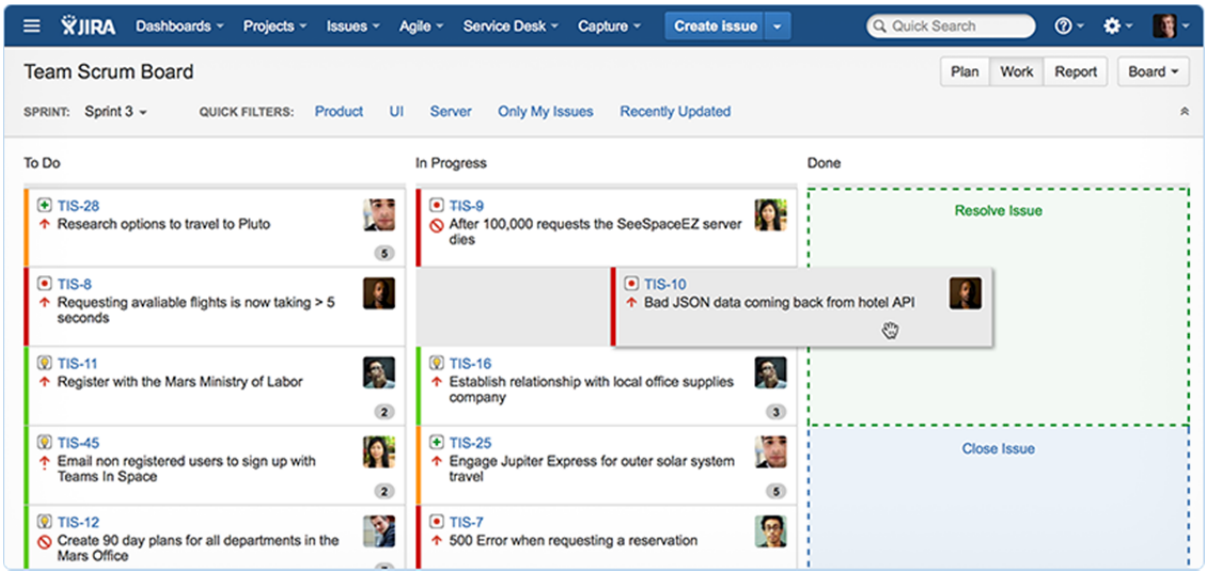

Figure 4.8 Exemple de tableau SCRUM Jira Agile (Jira Agile, 2015a)

Le graphique « Burndown » créé par l'outil Jira Agile est présenté dans la Figure 4.9.

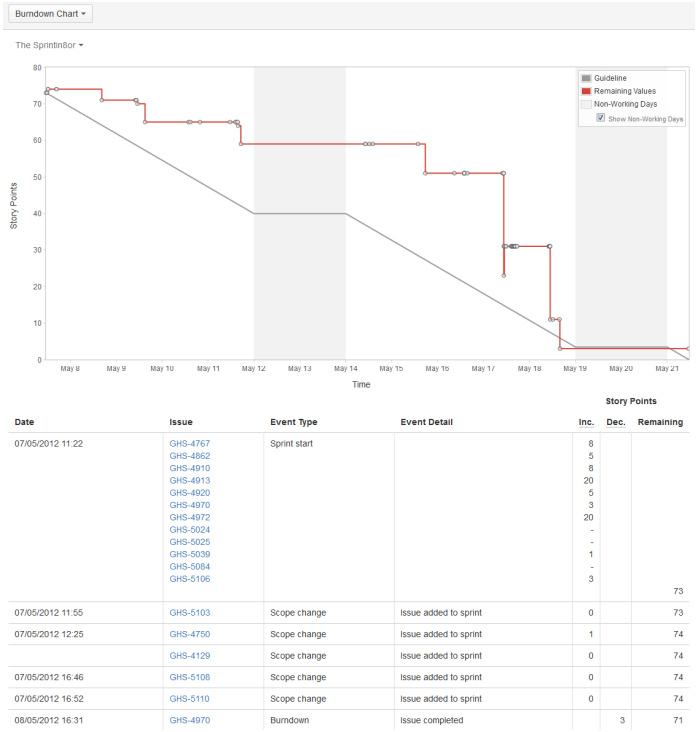

Figure 4.9 Exemple de graphique « Burndown » de Jira Agile (Jira Agile, 2015b)

4.2.3 Par rapport à l'application

Tous les outils précités offrent de versions de base jusqu'à des versions complètes. Le choix d'un outil dépend du type de projet, de la localisation de l'équipe et de la valeur que l'entreprise veut investir dans ce produit.

Si d'un côté, le fait que ces applications puissent être utilisées sur le nuage représente un gain pour les entreprises, en réduisant la complexité d'installation et le coût de l'infrastructure, d'un autre côté, il y existe un risque associé à la protection de l'information.

La Figure 4.10 présente un exemple de « Cloud computing ».

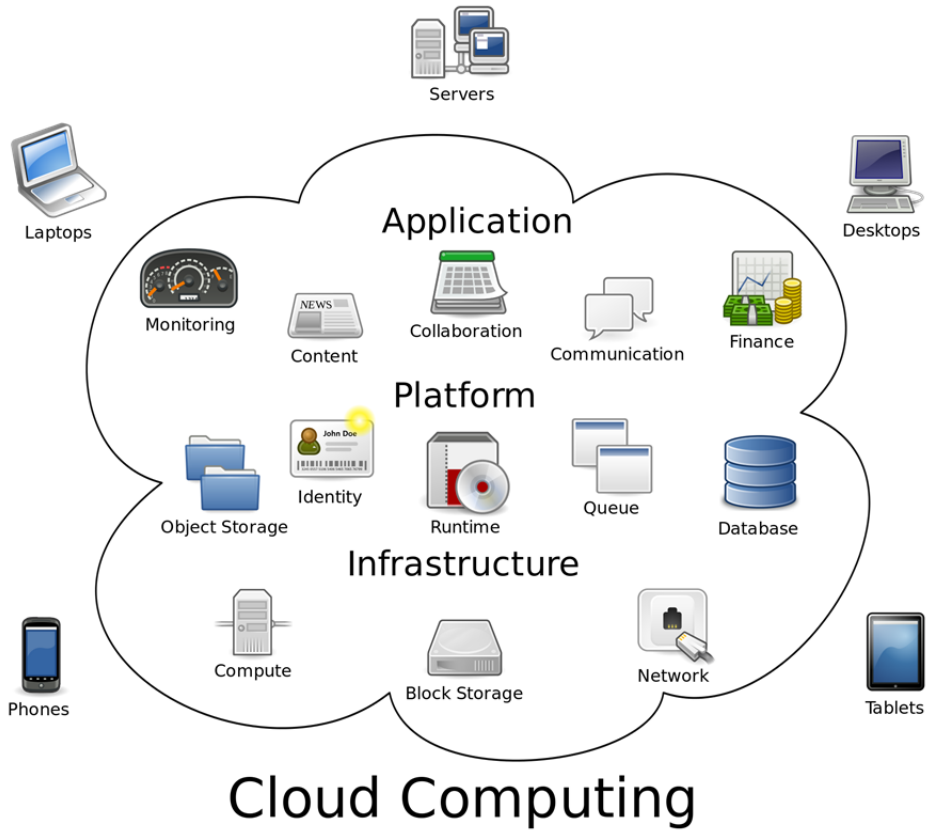

Figure 4.10 Exemple de « Cloud computing » (Aswin et Kavitha, 2012)

En 2011, deux groupes de « hackers » ont réussi à pénétrer les sites de Sony et Fox, ils ont volé des informations sensibles de plusieurs utilisateurs, comme le numéro de la carte de crédit. (Aswin et Kavitha, 2012) Cela veut dire que les entreprises doivent trouver des moyens pour sécuriser leurs informations en choisissant la bonne infrastructure pour héberger leurs données.

4.2.4 Comparaison des outils

Les outils étudiés, dans ce rapport, comblent les besoins de la majorité des équipes de projets, dans le sens où toutes les fonctionnalités de base sont comprises et l'on trouve même plus.

D'un autre côté, les utilisateurs ont exprimé ce qui leur plait de plus ou de moins dans un outil. Pour 91 % de répondants la « facilité d'utilisation » est l'aspect le plus important, tandis que, « l'absence d'intégration avec d'autres outils » est l'aspect plus décevant. (Azizyan, Magarian et Kajko-Matsson, 2011)

Pour les outils en question et surtout pour les outils de gestion de projets traditionnels, la facilité d'utilisation n'est pas vraiment une force, selon mon expérience. Ces outils demandent du temps pour bien être maitrisés. Par contre, l'outil MS-Project présente beaucoup d'intégration avec la suite « Office » par exemple, les courriels électroniques ou le navigateur web.

Les outils pour la gestion de projets logiciels, qui sont moins connus pour moi, ne semblent pas non plus être faciles à utiliser, car les équipes continuent à utiliser les moyens physiques pour gérer leurs projets, tel que mentionné à l'item 4.2.2 de ce rapport.

Le tableau suivant présente une comparaison, non exhaustive, entre les outils de gestion de projets.

Tableau 4.1 Comparaison entre les outils de gestion de projets traditionnels versus logiciels.

Outil	MS-Project	Primavera P6	VersionOne	Jira-Agile
Points positifs	Interface connue; Outil très puissant; Intégration avec les outils Microsoft; Gestion de ressources;	Outil très puissant; Plusieurs façons d'afficher le projet; Présentation de graphiques; Gamme complète de rapports;	Basé sur le web; Coût raisonnable; Collaboration entre les équipes; Planification et suivi puissants; Rapports d'avancement;	Planification suivie de projets puissants; Intégration avec d'autres outils (Confluence); Coût raisonnable; Facile à personnaliser; Graphique Burndown;
Points négatifs	Beaucoup d'options et de configurations requièrent du temps pour bien connaitre l'outil; Coût très élevé pour plusieurs utilisateurs; Environ $1,500.00 US dollars pour la licence;	Demande beaucoup de connaissance des utilisateurs pour donner les résultats espérés; Coût très élevé pour licence; Infrastructure (serveur) dédiée; Environ $2,500.00 US dollar pour la licence.	Interface complexe; Demande du temps pour bien maitriser l'application; La version de base pour une équipe de projet est gratuite et le prix augmente jusqu'à $39,00 par mois par utilisateur pour la version complète, plus une mensualité;	Le module de rapport est faible; Interface complexe pour les débutants. L'entreprise offre une large fourchette de prix dépendamment des besoins du client, de la quantité d'utilisateurs et du lieu d'hébergement du système; Pour 50 utilisateurs et basé sur le nuage le prix est de $300,00 US dollar par mois.

4.2.5 Le processus de mise à jour de projets (ingénierie versus logiciel)

Cette comparaison permet de voir les différences entre chaque approche. Commençons par le pourcentage d'avancement qui pour les projets traditionnels a tout un poids, car il a un impact sur le calcul de la valeur acquise et du coté de projets logiciels il est aussi important, mais, pour informer que la tâche est terminée et principalement cette partie du logiciel fonc-

tionne, ce que fait un lien avec les douze principes de la méthode Agile soit « livrer un logiciel qui fonctionne est la première mesure d'avancement ». (Agile Alliance, 2015)

L'autre différence importante entre les méthodes apparait au niveau du pouvoir des membres de chaque équipe. Dans les projets traditionnels ce sont les ingénieurs en chef ou les ingénieurs de projets qui estiment la durée et les efforts pour terminer une tâche tandis que dans les projets logiciels la ressource qui fait la tâche estime le temps qui reste pour terminer, cela donne plus de valeurs à l'estimation dans le sens que la ressource assume un compromis de terminer dans le délai annoncé.

Concernant le cycle de mise à jour, il existe deux explications pour l'écart entre les approches, la première est que les projets traditionnels évoluent beaucoup plus lentement que ceux de logiciels. La deuxième est que les projets traditionnels dépendent de l'entrée de données financières qui ne sont disponibles qu'à fin de la période fiscale.

Enfin, le processus de mise à jour des projets traditionnels est fait par une ressource spécifique et qualifiée pour travailler avec l'outil de gestion de projet tandis que les projets logiciels sont à mis à jour par les membres de l'équipe via le web.

Ces écarts entre les approches sont reflétés sur les outils de gestion, tel que décrit au tableau 3, les outils tels que, MS-Project et Primavera P6, demandent plus de temps à cause de leur complexité et la lourdeur de la gestion.

Tableau 4.2 Comparaison entre les données qui doivent être mises à jour dans un projet d'ingénierie versus un projet logiciel.

Données à mettre à jour	Projet d'ingénierie	Projet logiciel
Pourcentage d'avancement	Oui, il est fondamental pour les calculs de la valeur acquise; Valeurs de zéro à 100%	Oui, mais, il a deux valeurs seulement zéro ou 100%
Date de début et de fin	Oui, définie par les contraintes	Oui, définie par la durée du « sprint »
Les heures dépensées	Oui, via les feuilles de temps	Oui, définies par les « histories points » réalisées
Le reste à faire	Oui, défini par l'équipe de gestion, en heures	Oui, défini par les membres de l'équipe responsable de la tâche basée sur les « histories points » à faire
Cycle de mise à jour	Mensuel	Journalier
Source de l'information	Ingénieur du projet	Membre de l'équipe responsable de la tâche
Responsable pour la mise à jour	Planificateur du projet	Membre de l'équipe responsable de la tâche

4.3 L'impact des outils sur le succès des projets

4.3.1 Définition de succès des projets

Selon le PMI le succès d'un projet se définit comme suit :

> Comme les projets sont de nature temporaire, le succès du projet doit être mesuré en termes de réalisation du projet dans les contraintes de la portée, du temps, du coût, de la qualité, des ressources, et du risque approuvé par les chefs de projet et les cadres supérieurs. Pour assurer la réalisation des avantages pour le projet entrepris, une période de test (comme un prélancement dans les services) peut être une partie du temps total du projet avant de le remettre aux opérations permanentes. Le succès du projet devrait être mesuré par rapport aux dernières références approuvées par les intervenants autorisés. (Project Management Institute, 2013)

4.3.2 Le lien entre les outils de gestion et le succès de projets

Tel que mentionné dans le rapport du Standish Group, « Les outils et l'infrastructure peuvent contribuer au succès de projets, au même titre qu'ils peuvent le nuire. » (The Standish Group, 2013)

En effet, il n'existe pas une quantité suffisante de données probantes qui puissent vérifier l'impact direct des outils sur le succès de projets. Par contre, il n'est pas possible d'envisager, de nos jours, un projet qui démarre sans un outil de gestion de projet donc, le besoin est réel. De plus, un lien indirect pourrait être fait si l'on prend en considération que les outils présentés ici seront alimentés avec les bonnes données, pour professionnels qualifiés, de telle sorte que, les gestionnaires de projets puissent prendre de bonnes décisions au fur et à mesure que les projets avancent.

4.3.3 De bonnes pratiques à utiliser

Il est connu que plus un projet avance plus il en coute cher de faire des changements. (Project Management Institute, 2013) Cela dit, il faut éviter de déduire ce que le client veut. Dans les projets d'ingénierie, de plus en plus, les clients soit ceux qui payent pour le produit, soit ceux qui vont utiliser le produit, sont convoqués pour participer à des analyses fonctionnelles et des revues de conception. De cette manière, ils ont l'opportunité d'exprimer leurs besoins et de s'assurer du résultat final.

Des études récentes démontrent que l'application de la méthode Agile dans les projets de construction est possible, et ce surtout lors de la phase de conception du projet. Par contre, l'adoption de la méthodologie dans la phase de construction s'avère plus difficile, car la main-d'œuvre n'a pas les mêmes qualifications et un changement culturel serait nécessaire. (Fernandez et Fernandez, 2009)

CONCLUSION

Ce projet de recherche a débuté en questionnant les méthodologies et les outils de gestion de projets pour deux types de projets : les projets du type traditionnel, plus stable, avec les phases bien définis comme les projets d'ingénierie et les projets de développement de logiciels, qui sont plus susceptibles à de changements en cours d'exécution.

Les chapitres 1 et 2 ont présentés un historique de l'évolution de chaque méthodologie de gestion de projets jusqu'à aujourd'hui. Des concepts importants, comme le cycle de vie y a été discuté. Le PMBOK, la référence principale de la gestion de projets traditionnelle, propose une série de procédures et de processus que l'équipe de gestion doit suivre, ce qui illustre l'importance portée à la documentation, à la gestion de coût et aux délais pour cette méthodologie.

Le chapitre 3 a présenté la méthodologie Agile et la méthode SCRUM pour gérer les projets de développement de logiciels. Le document fondateur, le Manifesto Agile, et ses principes, a été présenté. Il insiste sur l'importance de la participation des clients et aux produits finaux qu'aux procédures et à la documentation, notamment, la divergence entre les deux approches.

Le chapitre 4 a présenté et comparé les outils de gestion de projets les plus utilisés pour chaque domaine en mettant en évidence ses qualités principales. Une liste, non exhaustive, démontre les points forts et faibles de chaque outil et l'investissement que l'entreprise doit faire pour les utiliser.

Ce chapitre a démontré aussi les différences entre les approches traditionnelles et Agile concernant la mise à jour de projets, ce qui donne un impact important sur les outils de gestion de projets pour chaque type de méthode. La lourdeur exprimée par l'approche traditionnelle par rapport à la flexibilité de l'approche Agile est claire. De plus, pour faire face à cette flexibilité demandée par l'agilité, l'outil de gestion doit être simple à utiliser et surtout dispo-

nible via le web pour que l'équipe puisse mettre à jour ses tâches en temps réel, ce qui ne fonctionne pas avec l'approche traditionnelle.

Par la suite, nous avons vu les détails plus importants de chaque outil soit la gestion de références et de ressources, soit la gestion du « product backlog » et de tableau SCRUM.

Ces applications doivent être hébergées. Celles de projets traditionnels sont normalement hébergées à l'interne sur un serveur d'entreprise. Les applications pour la gestion de projets logiciels peuvent aussi être hébergées sur le nuage, ce qui pourrait représenter un risque de perte de données. De plus, ces derniers sont accessibles via le web à cause de la nature décentralisée de certains projets de logiciels, c'est-à-dire, peu importe où l'équipe de projet est localisée, elle doit avoir accès à l'outil de gestion pour faire avancer ses tâches.

Un lien entre ses outils et le succès de projets a été fait, tout en respectant la fonction des outils qui est d'aider à la prise de décision.

Pour conclure, un survol de bonnes pratiques développées par les projets de logiciels qui pourront s'intégrer aux projets traditionnels a été fait. Une de bonnes pratiques mentionnées est la présence plus accentuée de clients pendant la phase de conception d'un projet.

De plus, le grand investissement qui représente ses outils au niveau d'achat de licences, d'infrastructure, de formation et d'administration n'est pas négligeable donc, il est impératif de bien les choisir.

Comme recommandation, une étude approfondie du lien entre les outils de gestion et les facteurs de succès de projets apporteraient un gain important pour les équipes de projets et aussi un facteur décisionnel plus éclairé. De plus, si l'application de méthodes Agile aux projets traditionnels devient une réalité, il va falloir ajuster les outils en conséquence.

LISTE DE RÉFÉRENCE BIBLIOGRAPHIQUE

Agile Alliance. 2015. « The Twelve Principles of Agile Software ». In *Le site d'Agile Alliance*. En ligne. < http://www.agilealliance.org/the-alliance/the-agile-manifesto/the-twelve-principles-of-agile-software/ >. Consulté le 3 mars 2015.

Aswin, M. R., et M. Kavitha. 2012. « Cloud intelligent track - Risk analysis and privacy data management in the cloud computing ». In *Recent Trends In Information Technology (ICRTIT), 2012 International Conference on*. (19-21 April 2012), p. 222-227.

Azizyan, G., M. K. Magarian et M. Kajko-Matsson. 2011. « Survey of Agile Tool Usage and Needs ». In *Agile Conference (AGILE), 2011*. (7-13 Aug. 2011), p. 29-38.

Cadle, James. 2014. *Developing Information Systems - Practical Guidance for IT Professionals*. Swindon, UK: BCS Learning & Development Ltd, 320 p.

Carayannis, Elias G., Young Hoon Kwak et Frank T. Anbari. 2005. *The Story of Managing Projects: An Interdisciplinary Approach, 1e éd.* Westport, Connecticut: Praeger Publishers, 388 p.

Fernandez, Daniel J., et John D. Fernandez. 2009. « Agile Project Management - Agilism versus traditional approaches ». *The journal of Computer Information Systems*, n° Winter 2008-2009, p. 10-17.

Flyvbjerg, Bent, et Alexandre Budzier. 2011. « Why your IT projects may be riskier than you think ». In *Harvard Business Review*. En ligne. < https://hbr.org/2011/09/why-your-it-project-may-be-riskier-than-you-think/ar/1 >. Consulté le 16 avril 2015.

International Organization for Standardization. 2008. *Systems and software engineering -- Software Life Cycle Processes,* 2e éd. Internatinal standard, ISO/IEC 12207: 2008. Geneva, Switzerland: International Organization for Standardization, 123 p.

International Organization for Standardization. 2011. *Software engineering - Lifecycle profiles for Very Small Entities (VSEs) - Part 4-1: Profile specification: Generic profile group, 1er éd.* International standard, ISO/IEC 29110-4-1:2011. Geneva, Switzerland: International Organization for Standardization, 52 p.

International Scrum Institute. 2015. « THE SCRUM PRODUCT BACKLOG ». In *Le site de SCRUM-Institute*. En ligne. < http://www.scrum-institute.org/The_Scrum_Product_Backlog.php >. Consulté le 18 juin 2015.

Jira Agile. 2015a. « Jira Agile Go Agile with easy ». In *Le site de l'entreprise Atlassian*. En ligne. < https://www.atlassian.com/software/jira/agile >. Consulté le 3 avril 2015.

Jira Agile. 2015b. « Viewing the Burndown Chart ». In *Le site de l'entreprise Atlassian Confluence*. En ligne. < https://confluence.atlassian.com/display/AGILE/Viewing+the+Burndown+Chart >. Consulté le 3 avril 2015.

Kent Beck et al. 2001. « Manifesto for Agile Software Development ». In *Le site d'Agile Manifesto*. En ligne. < http://www.agilemanifesto.org/ >. Consulté le 03 mars 2015.

Khan, PM, et MMS Sufyan Beg. 2013. « Extended decision support matrix for selection of sdlc-models on traditional and agile software development projects ». In *Advanced Computing and Communication Technologies (ACCT), 2013 Third International Conference on*. p. 8-15. IEEE.

Kniberg, Henrik (142). 2007. *SCRUM and XP from the trenches,* 2e éd. USA: C4Media, 142 p.

Larman, C., et V. R. Basili. 2003. « Iterative and incremental developments. a brief history ». *Computer,* vol. 36, no 6, p. 47-56.

Liberatore, Matthew J., Bruce Pollack-Johnson et Colleen A. Smith. 2001. « Project Management in Construction: Software Use and Research Directions ». *Journal of Construction Engineering and Management,* vol. 127, no 2, p. 101-107.

Microsoft Corporation. 2015. « Project and Portfolio Management Software ». In *Le site de Microsoft Corporation.* En ligne. < https://products.office.com/en-us/project/project-and-portfolio-management-software >. Consulté le 19 juillet 2015.

Montain Goat Software. [s.d.]. « Moutain Goat Software ». En ligne. < http://www.mountaingoatsoftware.com/agile/scrum/task-boards >. Consulté le 3 avril 2015.

Mulcahy, Rita. 2005. *PMP Exam Prep,* 8e éd. Minnesota, USA: RMC Publications, Inc, 624 p.

Oo, Min. [s.d.]-a. « Creating discipline wise manpower histogram using Primavera P6 export data ». En ligne. < http://www.toolboxforplanning.com/2011/01/tutorial-generate-s-curvehistogram-from.html#.VZAetvl_Oko >. Consulté le 10 avril 2015.

Oo, Min. [s.d.]-b. « How to Show Baseline vs Current Project on P6 Gantt Chart ». En ligne. < http://www.toolboxforplanning.com/2013/07/how-to-show-baseline-vs-current-project.html#.VZAa4fl_Oko >. Consulté le 10 avril 2015.

Project Management Institute. 2013. *Guide to the Project Management Body of Knowledge, 5e éd.* American National Standard, ANSI/PMI 99-001-2013. Pennsylvania, USA: Project Management Institute Inc, 567 p.

Schwaber, Ken, et Jeff Sutherland. 2013. « The Scrum Guide ». En ligne. 16 p. < http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us.pdf >. Consulté le 20 avril 2015.

The Standish Group. 2013. « Chaos Manifesto 2013 ». En ligne 48 p. < http://athena.ecs.csus.edu/~buckley/CSc231_files/Standish_2013_Report.pdf >. Consulté le 20 avril 2015.

VersionOne Inc. 2014. « 8th Annual State of Agile Survey ». En Ligne. 14 p. < http://www.versionone.com/pdf/2013-state-of-agile-survey.pdf >. Consulté le 12 novembre 2014.

VersionOne Inc. 2015a. « Agile Reports & Metrics ». In *Le site de VersionOne*. En ligne. < http://www.versionone.com/product/agile-reporting/ >. Consulté le 3 avril 2015.

VersionOne Inc. 2015b. « Kanban Board Software ». In *Le site de VersionOne*. En ligne. < http://www.versionone.com/product/kanban-board/#top >. Consulté le 3 avril 2015.

Wikipédia l'encyclopédie libre. 2015. « Scrum (méthode) ». In *Le site de Wikipédia.* En ligne. < http://fr.wikipedia.org/w/index.php?title=Scrum_(m%C3%A9thode)&oldid=116063910 >. Consulté le 19 juin 2015.

Wysocki, Robert K. 2014. « Effective Project Management - Traditional, Agile, Extreme ». In *Le site Library books 24x7*. En ligne. < Le site http://library.books24x7.com/toc.aspx?bookid=31996 >. Consulté le 5 mars 2015.