



HAL
open science

Méthodologie de gestion agile d'un projet : Scrum - les principes de base

Alain Collignon, Joachim Schöpfel

► **To cite this version:**

Alain Collignon, Joachim Schöpfel. Méthodologie de gestion agile d'un projet : Scrum - les principes de base. I2D – Information, données & documents, 2016, 53 (2), pp.12-15. hal-01412080

HAL Id: hal-01412080

<https://hal.univ-lille.fr/hal-01412080v1>

Submitted on 13 Jun 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Méthodologie de gestion agile d'un projet. Scrum – les principes de base

Comment éviter les effets « silo » et « tunnel » des méthodes classiques de développement ? Comment rapprocher conception, programmation et déploiement, créer des liens forts entre utilisateurs et développeurs, valoriser le métier et la fonction information-documentation et limiter le risque d'une démobilité de la part des utilisateurs et des clients pendant la phase, parfois très longue, de programmation ? Les méthodes agiles, et notamment Scrum, peuvent faciliter la gestion de projet et contribuer à une culture de changement.

Tout chef de projet l'a appris un jour : le bon déroulement d'un projet implique une démarche structurante avec une planification, une gestion des ressources humaines et un suivi des enjeux financiers, aboutissant à la « recette » (test d'acceptation)¹. Dans un projet classique, avec un « *déroulé linéaire de phases (et) une séparation chronologique entre la conception et la réalisation* »², la « recette » est l'étape ultime. Or, en amont, la rédaction des spécifications demande beaucoup de temps ; en aval, l'estimation de charges parfois pharaoniques génère des délais interminables, démotivant les parties prenantes et ôtant toute dynamique à l'équipe. Trop souvent, frustration, travail inutile et explosion des délais et des ressources sont inhérents à la gestion classique de projet. Pendant ce temps, les besoins des utilisateurs et des clients continuent d'évoluer, au risque d'un produit non conforme aux attentes.

Des méthodes agiles à Scrum

En 1986, Barry W. Boehm introduit un nouveau modèle de développement reposant sur une structure commune itérative, incrémentale et adaptative, les méthodes agiles³, dont le fil conducteur consiste à découper le déroulement d'un projet en plusieurs objectifs plus petits afin d'obtenir plus sûrement et rapidement un résultat (ou *minimal vital product* – MVP). En 2001, un manifeste⁴ introduit quatre valeurs fondamentales qui définissent une nouvelle façon de développer des logiciels en valorisant :

- 1 les individus et leurs interactions plus que les processus et les outils ;
- 2 des logiciels opérationnels plus qu'une documentation exhaustive ;
- 3 la collaboration avec les clients plus que la négociation contractuelle ;
- 4 l'adaptation au changement plus que le suivi d'un plan.

Les méthodes agiles répondent aux méthodes classiques, trop prédictives et trop rigides, en exposant de nouveaux principes plus souples dont l'anticipation, l'auto-régulation, le *feedback* et la collaboration⁵. Elles renforcent aussi la capacité d'une organisation « apprenante » au changement et à la transformation.

¹ Voir notamment : Pierre-Yves DUCHEMIN, Dominique LAHARY. *L'art d'informatiser une bibliothèque : guide pratique*, 2^e édition. Électre, Éditions du Cercle de la Librairie, 2000

² David AUTISSIER, Jean-Michel MOUTOT. *Le changement agile : se transformer rapidement et durablement*. Dunod, 2015

³ Julien PLÉE. *Conduite de projets agiles : management alternatif dans une équipe de développement agile*. Éditions ENI, 2015

⁴ <http://agilemanifesto.org/iso/fr>

⁵ Fanny LENUZZA. *Les méthodes agiles, une approche complexe de la gestion de projet en équipe pluri-générationnelle*. Mémoire de master, UFR LLASIC, Université Stendhal Grenoble 3, 2015 <http://dumas.ccsd.cnrs.fr/dumas-01212068/document>

Scrum est la plus connue des méthodes agiles⁶. Créée en 1996 par Ken Schwaber, elle met en avant l'aspect soudé d'une équipe auto-organisée cherchant à atteindre un but partagé. La particularité de Scrum est de placer l'utilisateur final au cœur de l'équipe et de valoriser l'individu, l'équipe, le concret, l'application, la collaboration et l'adaptation. Scrum n'est pas un acronyme mais le mot anglais signifiant mêlée dans un match de rugby⁷. Ce n'est pas une méthode au sens strict du terme mais plutôt une approche, un cadre de processus et un ensemble de principes, presque une philosophie fondée sur le changement, la culture du résultat, la transparence et la communication, le respect des utilisateurs et des clients – et l'esprit d'équipe.

Les caractéristiques de Scrum

Comme les autres méthodes agiles, Scrum repose sur une approche empirique, avec une série de cycles de développement de courte durée, appelés « sprints », et des rétroactions fréquentes. Ces sprints sont des blocs de temps fixes, le plus souvent de 2 à 4 semaines, avec un rythme et une régularité prédéterminés. Ils sont itératifs et adaptatifs en fonction des besoins ou objectifs évolutifs du client (utilisateurs). À la fin de chaque cycle, un livrable est présenté au cours d'une « revue de sprint ».

Ce processus permanent d'adaptation et d'échange entre développeurs et clients et/ou utilisateurs (l'équipe sprint mêle les deux) permet d'obtenir un produit proche des besoins client en prenant en compte l'évolution de ces besoins et de maximiser ainsi la valeur du produit livré. La notion de valeur utilisateur ou « *business value* » est le leitmotiv de Scrum. La livraison dès les premiers sprints et l'utilisation partielle du produit facilitent une remontée rapide de l'avis du *product owner* (représentant des utilisateurs).

Les spécifications sont beaucoup plus souples car si des fonctionnalités ne répondent pas aux besoins, il suffit d'ajouter une modification ou une nouvelle fonctionnalité non prévue au départ dans des sprints suivants – c'est ce qui rend cette méthode « agile ». Pour que le système fonctionne, il faut prioriser des besoins et des objectifs au sein de l'équipe Scrum. Une autre condition est l'implication des membres de l'équipe dans le projet en participant aux prises de décisions, en échangeant sur les difficultés rencontrées. La productivité de l'équipe est ainsi augmentée surtout parce qu'elle participe à l'évaluation chiffrée (sprint planning). Pas de planification lourde, celle-ci est adaptative, avec des ajustements si nécessaires au fil de l'eau en fonction des nouvelles demandes ; seule une feuille de route (*roadmap*) est rédigée, consignnant les jalons à fortes valeurs ajoutées. Comme la documentation est réduite au strict nécessaire, les interminables phases de spécifications fonctionnelles sont remplacées par des « *user stories* », plus succinctes.

Les acteurs

En mettant l'accent sur le fonctionnement en équipe, Scrum remplace les chefs de projets, maîtres d'œuvre et d'ouvrage par d'autres rôles et fonctions.

- Le *product owner*. Ce responsable du produit représente les clients et/ou utilisateurs en transcrivant leurs besoins, définit et priorise les demandes produit ; il doit posséder une expertise fonctionnelle métier.

⁶ Claude AUBRY. *Scrum : le guide pratique de la méthode agile la plus populaire*. Dunod, 2015

⁷ On traduit parfois Scrum par « fonctionnement mêlé ».

- Le *Scrum master*. Membre de l'équipe de développement, il en est aussi l'animateur. Il n'est pas chef de projet mais doit faciliter l'application de Scrum. Mettant tout en œuvre pour que l'équipe travaille dans de bonnes conditions et se concentre sur l'objectif du projet, il porte une attention au respect des différentes phases de Scrum.

- L'équipe (le *team*). Elle se gère en toute autonomie et est chargée du développement du produit. Sa taille doit rester réduite (7 à 9 personnes). Auto-organisée, dépourvue de hiérarchie, sans spécialisation des rôles, elle est « *composée des personnes qui contribuent à produire un résultat à chaque sprint [...], des créateurs de valeur* »⁸.

Applications et indications

Plusieurs grands organismes, à l'instar de la BnF ou de l'Inist du CNRS, ont formé leurs équipes aux principes Scrum dans le domaine de l'ingénierie documentaire. On lit parfois que Scrum est réservé aux projets d'envergure, compliqués et longs. Or, il n'y a pas d'indication particulière pour les méthodes agiles tant pour la taille de service, la complexité du projet ou la nature du produit. L'approche agile, bien plus qu'une méthode de développement informatique, sert déjà de cadre conceptuel au management des bibliothèques et au développement des fonctions et compétences, du travail en équipe et de la gestion des usagers et lecteurs⁹. Dans le domaine de l'information et de la documentation, on note des retours d'expérience pour la création d'un site web, le développement d'une banque d'images, la gestion d'une bibliothèque ou le développement d'un SGB¹⁰. L'essentiel est l'engagement du service et/ou de l'organisme dans une démarche de changement, avec une approche pragmatique et une orientation utilisateur clairement affichée. L'intérêt est de maintenir la motivation des équipes et des utilisateurs (clients) et d'éviter les dérapages en matière de ressources et délais, ceci dans un processus favorisant la transparence du développement (« management visuel ») et l'émergence de nouvelles idées.

Limites et obstacles

Se lancer dans un projet agile n'est pas sans risque. Scrum n'est pas synonyme de réussite. L'application des principes agiles ne garantit pas le succès d'un projet de développement. Contrairement aux projets classiques, le périmètre du projet n'est pas consigné dans un cahier des charges, en amont du développement. Cependant, l'analyse des besoins reste indispensable.

L'avantage d'un *feedback* rapide par la livraison de produits partiels fonctionnels est conditionné par une très bonne collaboration et par l'acceptation du changement¹¹. Mais quand les équipes ne fonctionnent pas, si le *Scrum master* ne connaît pas suffisamment les autres acteurs ou si le *product owner* ne maîtrise pas le métier, des problèmes surgiront. De même, si les rôles traditionnels – chef de projet, maître d'ouvrage (MOA), maître d'œuvre (MOE), etc. – persistent et si chaque membre d'équipe est évalué individuellement et non pas la performance collective, Scrum ne donnera pas de résultats satisfaisants.

⁸ AUBRY, *loc.cit.*, p.28-29

⁹ C. WILLIAMS. « Flexible and agile university library and information services: skills and management methodologies ». In : David BAKER, Wendy EVANS (dir.). *Trends, Discovery, and People in the Digital Age*. Chandos Publishing, 2013, p. 241-251

¹⁰ Voir aussi www.citeulike.org/user/Schopfel/tag/agile_management

¹¹ S. LEGRAS. « L'agilité, nouvelle transformation pour l'entreprise », *Documentaliste-Sciences de l'information*, 2014, n°4, p. 5-6

La peur du changement est l'un des facteurs de blocage de l'adoption de Scrum. Il en est de même d'une méthode mal expliquée, les différentes réunions de *reporting* pouvant être mal interprétées ou vécues, tel le *sprint daily* transformé en réunion de surveillance. Ici, la fonction MOA ne se limite plus à la rédaction d'un cahier des charges ; elle est au cœur de chaque cycle et en interaction permanente avec les développeurs (MOE). Chaque partie doit trouver sa place dans un nouvel équilibre à établir.

Un autre point de vigilance est la prise de conscience des notions de coût, de délai et de périmètre. Dans les méthodes classiques, le périmètre du produit à développer est une constante ; pour réussir le produit, les coûts et les délais sont réajustés pour atteindre les objectifs fixés. Dans la méthode Scrum, les délais et les coûts sont définis comme des invariables et le périmètre devient la variable du projet. Par conséquent, le rôle du *product owner* est déterminant afin que les fonctionnalités à hautes valeurs ajoutées puissent être correctement priorisées au cours des *sprints backlog*. Aussi, un échange trop technique peut empêcher une vision plus globale (« politique ») du produit et un contrat trop rigide sans flexibilité s'avérer contre-productif¹².

En matière de durée, ce sont *a priori* des cycles courts de 2, 3 ou 4 semaines qui s'enchaînent d'une façon régulière et sans interruption pour maintenir le rythme de lancement et la motivation et l'énergie de la « mêlée ». Ceci implique que l'ensemble du processus (le projet) ne devrait pas durer trop longtemps, au risque d'épuiser l'équipe. En règle générale, la durée totale jusqu'à la livraison du produit est de 3 ou 4 mois.

Perspectives

Est-ce la fin du cahier des charges ? Oui, s'il s'agit d'un planning précis et du caractère définitif des spécifications fonctionnelles puisque « *spécifier pendant plusieurs mois un périmètre complet qui mettra encore plus de temps à être livré n'a pas de sens* »¹³. Non, s'il s'agit de l'analyse des besoins et de l'étude de l'existant pour fixer l'objectif et la description des principales fonctionnalités attendues. Le cahier des charges jouera toujours un rôle pour le pilotage et la gestion d'un projet. Cependant, le « *point focal de pilotage [...] se déplace : le périmètre qui était jusqu'à présent le point central du projet s'écarte peu à peu pour être remplacé par la valeur* »¹⁴.

Ce glissement vers la valeur et les besoins et attentes des utilisateurs rapproche le processus Scrum d'autres approches dans le domaine de l'ingénierie documentaire qui placent l'expérience de l'utilisateur (*user experience* ou UX) au cœur du développement de nouveaux outils et services.

Le Guide Scrum¹⁵ qualifie son cadre de léger, simple à comprendre mais difficile à maîtriser. Cet article ne vise qu'à présenter la méthode, de manière générale. Pour appliquer Scrum et le maîtriser, on s'appuiera sur les documents cités. Rappelons toutefois aussi que, dans un projet, une fois la méthode de gestion, la technologie, les *frameworks* ou structures logicielles, etc. choisies, ce sont des humains qui réaliseront le projet. ■

¹² AUBRY, *loc. cit.*, p. 288-290

¹³ S. LEGRAS, *loc. cit.*

¹⁴ *Id.*

¹⁵ K. SCHWABER, J. SUTHERLAND. *Le guide Scrum*. Juillet 2013
www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-FR.pdf

Hors-texte

Terminologie Scrum¹⁶

Daily Scrum (ou mêlée quotidienne) : moment de partage, courte « cérémonie » ou réunion (environ 15 minutes) menée chaque jour avec les membres de l'équipe et le *product owner* pour informer sur l'activité de tous, déterminer les tâches de la journée et identifier les obstacles empêchant l'équipe d'atteindre l'objectif du sprint.

Product Backlog (ou produit backlog) : liste priorisée des fonctionnalités du produit, soit une liste partagée des choses à faire.

Rétrospective : « cérémonie » à l'issue de chaque sprint dont l'objectif est de discuter, en présence du *Scrum master* et du *product owner*, des axes d'amélioration de l'équipe. Moment capital qui incarne l'un des principes fondamentaux énoncés par le Manifeste agile : à intervalles réguliers, l'équipe réfléchit sur la manière de préparer le futur et d'accroître son efficacité, puis adapte et ajuste son comportement en conséquence.

Sprint Backlog : liste des fonctionnalités à réaliser lors d'un sprint. La liste des tâches correspondantes est établie lors d'un sprint planning. La charge des tâches est déterminée par les développeurs.

Sprint Daily (voir Daily Scrum).

Sprint Planning (ou planification) : il réunit l'équipe et le *product owner* pour déterminer l'objectif et le contenu du sprint à venir. Les *user stories* sont découpées en tâches faisant chacune l'objet d'une estimation des charges, exprimée en heures ou en jours, par l'équipe. Le principal résultat est le sprint backlog.

Sprint Review (ou revue de sprint) : elle consiste pour l'essentiel, au terme de chaque cycle, à faire une démonstration publique du résultat du sprint, à collecter le *feedback* des parties prenantes et à prendre des décisions.

User Story : élément du backlog. Une exigence du système à développer, une brève description d'une fonctionnalité telle que vue par l'utilisateur (client).

Vision produit (ou vision) : décrivant les principaux objectifs, jalons, utilisateurs visés et rédigée par le Product Owner, elle contribue à guider et à fédérer les acteurs du projet.

Hors texte

Trois questions à Patrick Kremer, Scrum Master à l'Inist – CNRS

> Quelle est la particularité du *Scrum master* ?

En méthode agile, le *product owner*, en relation avec les utilisateurs, possède la vision du produit et se partage les fonctions de chef de projet utilisateur classique avec le *Scrum master* qui assure l'animation, la cohésion et l'autonomie de l'équipe. Le *Scrum master* joue essentiellement un rôle de facilitateur et

¹⁶ C. AUBRY, *loc. cit.*

de médiateur pour améliorer le fonctionnement de l'équipe et les relations entre celle-ci et l'extérieur. Il lui appartient d'anticiper et gérer les éventuels conflits ; il n'a aucune fonction hiérarchique.

> Dans votre travail, quels sont les bénéfices de la méthode Scrum ?

Le premier intérêt est de travailler sur des cycles courts : les sprints d'environ trois semaines obligent à présenter un livrable (fonctionnalités du produit final), ce qui est très valorisant et motivant pour l'équipe et le projet. La méthode Scrum permet également à chacun de se concentrer sur ses tâches tout en étant en contact permanent avec le reste de l'équipe, ce qui amène à gérer rapidement les problèmes, améliorer les processus et avoir un niveau de connaissance global du projet.

> Trois écueils à éviter ?

Le *Scrum master* anime l'équipe mais ne la dirige pas, il n'attribue pas de tâches et ne gère pas les ressources humaines.

Le cérémonial Scrum prévoit un nombre de réunions lors desquelles peuvent être abordés tous les aspects du projet mais sans les amalgamer, chacun ayant une spécificité particulière. Chaque réunion a un but et une durée : lors des réunions journalières (*daily*), par exemple, on expose des problèmes dont les solutions seront discutées à d'autres moments.

Le *Scrum master* est un facilitateur à qui il n'incombe pas de résoudre tous les problèmes ; c'est l'équipe qui le fera.■

Alain Collignon
Ingénieur de recherche Inist - Cnrs
Chargé d'enseignement, Université de Lorraine
Alain.COLLIGNON@inist.fr

Joachim Schöpfel
Maître de conférences, Université Lille 3
joachim.schopfel@univ-lille3.fr