

INTRODUCTION AUX MÉTHODES AGILES ET SCRUM



FLORENT LOTHON
EXPERT en GESTION DE PROJET et management
D'ÉQUIPE



Ce document est sous licence Creative Commons - Usage non commercial.

En clair, cela signifie que vous êtes libre d'utiliser et diffuser son contenu aux seules conditions suivantes :

- Citer l'auteur avec un lien vers son site.
- Ne pas modifier le contenu sans autorisation préalable de sa part.
- Ne pas commercialiser le contenu.

SOMMAIRE

[Introduction](#)

[Approche Agile plutôt que méthode Agile](#)

[Une autre approche de gestion de projet](#)

[Fonctionnement des méthodes agiles](#)

[Témoignage client](#)

[Historique des méthodes agiles](#)

[Le Manifeste Agile, un changement culturel](#)

[Valeurs agiles et idées reçues](#)

[Principes sous-jacents au Manifeste Agile](#)

[Le cadre méthodologique Scrum](#)

[Le "package" Scrum](#)

[L'organisation générale](#)

[Travaux préparatoires](#)

[Enchaînement des sprints](#)

[Mesure de l'avancement](#)

[La contractualisation agile](#)

[Envie d'aller plus loin ?](#)

[A propos de l'auteur](#)

Introduction

Vous avez sûrement entendu parlé des méthodes agiles ou de la méthode agile. Certains la perçoivent comme une énième méthodologie à la mode, difficilement compatible avec leur contexte. Surtout dans le cadre d'un contrat au forfait. Qu'est ce que l'approche agile au juste ? D'où vient-elle ? Comment s'applique-t-elle concrètement ? Voilà ce dont traite cette introduction aux méthodes agiles et à Scrum en particulier.

Approche Agile plutôt que méthode Agile

Si l'approche agile est nouvelle pour vous, il me semble important de partir sur de bonnes bases. Laissez moi vous dire au passage, que je suis honoré d'être votre guide vers ce nouvel horizon.

Le terme "méthode" est trop réducteur pour parler de cette façon d'aborder la gestion de projet. Il s'agit de bien plus qu'une méthode. On parle plutôt de paradigme agile, d'état d'esprit agile, de philosophie agile, de culture Agile ou encore d'approche agile, de mouvement agile, de courant agile, etc. En poursuivant votre lecture et en concentrant notamment votre attention sur le paragraphe "Le Manifeste Agile, un changement culturel", vous comprendrez mieux cette dimension cruciale.

On parle cependant de "méthodes agiles" pour définir les méthodes qui relèvent de ce courant.

Une autre approche de gestion de projet

Le terme "agile" définit une approche de gestion de projet qui prend le contre-pied des approches traditionnelles prédictives et séquentielles de type cycle en V ou waterfall (en cascade). La notion même de "gestion de projet" est remise en question au profit de "gestion de produit". De façon à raisonner davantage "produit" que "projet". Après tout, l'objectif d'un projet consiste bien à donner naissance à un produit.

Une approche dite "traditionnelle" attend généralement du client une expression détaillée et validée du besoin en entrée de réalisation, laissant peu de place au changement. La réalisation dure le temps qu'il faut et le rendez vous est repris avec le client pour la recette. Cet effet tunnel peut être très néfaste et conflictuel, on constate souvent un déphasage entre le besoin initial et l'application réalisée. On se rapporte alors aux spécifications validées et au contrat. Certains projets se terminent dans la douleur (surtout dans le cadre d'un contrat au forfait classique) au risque de compromettre la relation client. De plus, il n'est pas rare que certaines fonctionnalités demandées se révèlent finalement inutiles à l'usage alors que d'autres, découvertes en cours de route, auraient pu donner plus de valeur au produit.

Une enquête de 1994 du « Standish Group » (certes controversée, comme toutes les enquêtes qui traitent d'un sujet sensible) fait le constat suivant : « 31 % des projets informatiques sont arrêtés en cours de route, 52 % n'aboutissent qu'au prix d'un important dépassement des délais et du budget tout en offrant moins de fonctionnalités qu'il n'en était demandé ; seuls 16 % des projets peuvent être considérés comme des succès. ».

Cette même enquête renouvelée en 2008 indique un taux de réussite de 35%, ce qui est plutôt positif mais demeure très faible. Le problème reste entier. Parmi les motifs d'échecs, arrivent en tête :

- △ Manque d'implication des utilisateurs finaux : 12,8 %.
- △ Changements de spécifications en cours de projet : 11,8 %.

L'approche Agile propose au contraire de réduire considérablement voire complètement cet effet tunnel en donnant davantage de visibilité, en impliquant le client du début à la fin du projet et en adoptant un processus itératif et incrémental. Elle considère que le besoin ne peut être figé et propose au contraire de s'adapter aux changements de ce dernier. Mais pas sans un minimum de règles.

Anthony Bleton de Novius; dans [sa vidéo intitulée "Oubliez le cahier des charges, soyez agiles !"](#); explique très bien en quoi l'approche agile se distingue de l'approche traditionnelle. Le tout en moins de 4 minutes, belle performance !

Fonctionnement des méthodes agiles

Les méthodes agiles partent du principe que spécifier et planifier dans les détails l'intégralité d'un produit avant de le développer (approche prédictive) est contre productif. Cela revient à planifier dans les détails un trajet "Paris - Narbonne" en voiture par les petites routes. Spécifiant chaque villes et villages traversés, l'heure de passage associée, chaque rue empruntée dans les agglomérations, litres d'essence consommés, kilomètres parcourus, etc. Les imprévus ne manqueront pas d'arriver : embouteillages, déviations, travaux, sens de circulation inversés, voire la panne, etc. Rendant votre planification et vos spécifications très vite obsolètes. Combien de temps aurez vous passé à planifier cet itinéraire, comment réagirez vous face à vos frustrations de ne pas pouvoir appliquer votre plan à la lettre ?

L'idée consiste à se fixer un premier objectif à courts termes (une grande ville par exemple) et se lancer sur la route sans tarder. Une fois ce premier objectif atteint, on marque une courte pause et on adapte son itinéraire en fonction de la situation du moment. Et ainsi de suite jusqu'à atteindre la destination finale. On parle donc d'une approche empirique. Dans le cadre d'un projet de développement logiciel, le client élabore sa vision du produit à réaliser et liste les fonctionnalités ou exigences de ce dernier. Il soumet cette liste à l'équipe de développement, communique directement avec elle (plutôt que par papier) qui estime le coût de chaque élément de la liste. On peut ainsi se faire une idée approximative du budget global.

L'équipe sélectionne ensuite une portion des exigences à réaliser dans une portion de temps courte appelée itération. Chaque itération inclut des travaux de conception, de spécification fonctionnelle et technique quand c'est nécessaire, de développement et de test. A la fin de chacune de ces itérations, le produit partiel mais utilisable est montré au client. Ce dernier peut alors se rendre compte par lui même très tôt du travail réalisé, de l'alignement sur le besoin. L'utilisateur final quant à lui peut se projeter dans l'usage du produit et émettre des feedbacks précieux pour les futures itérations. La visibilité ainsi offerte est clef. Cette transparence peut également apporter davantage de confiance et de collaboration dans la relation client/fournisseur. Les risques quant à eux sont levés très tôt.

Si le client a priorisé avec soin son besoin, il peut saisir l'opportunité d'accélérer le "time to market" si il estime que le produit en l'état (partiel) peut aller en production. Économisant ainsi son budget et récoltant un premier retour sur investissement. Il a aussi la possibilité de changer en cours de route la priorité des fonctionnalités qui n'ont pas encore été développées (prévues pour les itérations futures). Afin de retarder une fonctionnalité dont

le besoin n'est pas mûr, ajouter une nouvelle fonctionnalité cruciale en échange du retrait d'une autre (respectant ainsi budget et délais), etc.

Cette souplesse ainsi offerte est donc un véritable atout pour le client.

Témoignage client

Thierry Roche, alors DSI de l'APEC

Quels ont été les avantages de ces méthodes pour votre projet ?

“Déjà, on voit concrètement l'évolution du projet car après chaque itération, les utilisateurs peuvent visualiser un « bout » de projet qui fonctionne, ça brise l'effet tunnel des méthodes classiques. Cette évolution nous permet de prioriser nos réels besoins, l'application s'enrichit selon nos demandes. Le surplus n'existe pas, on ne développe pas des fonctionnalités qui ne nous serviront jamais comme c'est régulièrement le cas en adoptant des méthodes classiques. On atteint un bénéficiaire utilisateur plus rapidement mais aussi un gain financier non négligeable. Nous n'imaginons même pas un retour avec des méthodes classiques.”

Source : Le Monde Informatique | [Dossier méthodes agiles : Le renouveau des relations client/fournisseurs](#).

Historique des méthodes agiles

Sans rentrer dans les détails, la première chose à savoir est que l'approche Agile n'est pas un effet de mode né de la dernière pluie. En effet la première approche de gestion de projet de développement itératif date de 1986. La première mise en œuvre de la méthode Scrum (la méthode Agile la plus utilisée, documentée et éprouvée aujourd'hui) date de 1993.

La seconde concerne un événement majeur rassemblant, en 2001, dix sept figures éminentes du développement logiciel pour débattre du thème unificateur de leurs méthodes respectives. De cet événement est né le [Manifeste Agile](#) rassemblant à la lueur des expériences de chacun les critères pour définir une nouvelle façon de développer des logiciels. Le terme "Agile" pour qualifier ce type de méthode est également né à cette occasion.

Aujourd'hui ces méthodes ont fait leurs preuves. Tout le monde (dans le monde de l'informatique) ou presque a au moins entendu parler d'une méthode Agile (Scrum, eXtreme Programming, RAD, Crystal Clear,...). L'outillage associé est maintenant disponible sur le marché y compris dans le secteur Open Source. Les formations, certifications, conférences, livres, blogs se sont multipliés. Nous pouvons également noter la prise de position en faveur de l'approche Agile de la part des organismes faisant "autorité" en matière de gestion de projet informatique :

- Le SEI à l'origine de CMMI en 2008. Voir le rapport ["CMMI or Agile: Why Not Embrace Both!"](#).
- Le PMI quant à lui créé en 2011 ses propres certifications Agile.
- En 2012, le Gartner invite à [abandonner le Cycle en V](#).

Le Manifeste Agile, un changement culturel

Le manifeste Agile contient l'essence et la philosophie de l'approche en question. Il illustre à lui seul le changement culturel profond qui est en jeu.

Valeurs agiles et idées reçues

EXTRAIT DU MANIFESTE AGILE > LES 4 VALEURS

Nous découvrons comment mieux développer des logiciels par la pratique et en aidant les autres à le faire. Ces expériences nous ont amenés à valoriser :

- ★ 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

Nous reconnaissons la valeur des seconds éléments, mais privilégions les premiers.

ATTENTION AUX IDÉES REÇUES

La dernière phrase a son importance car si on néglige sa lecture, on peut tomber dans les idées reçues très répandues du style : "Sur un projet agile, il n'y a pas de spécifications, de plan, de processus, d'outil et même pas de contrat".

Au passage, passons en revue d'autres idées reçues "Un projet Agile ne fonctionne que sur des projets de développement web, qu'avec une dizaine de personnes maximum, qu'avec des super développeurs" ou encore "sur un projet agile, le client change d'avis tout le temps et on tourne en rond à faire et défaire des choses".

Principes sous-jacents au Manifeste Agile

EXTRAIT DU MANIFESTE AGILE > LES 12 PRINCIPES

Nous suivons ces principes :

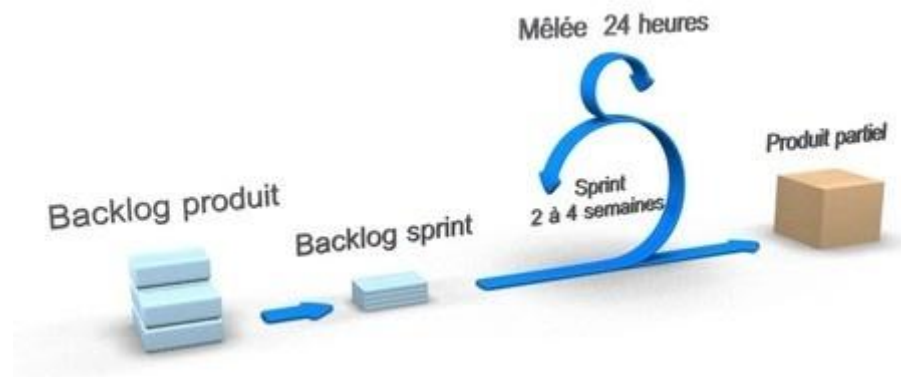
- ★ Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.
- ★ Accueillir positivement les changements de besoins, même tard dans le projet. Les processus Agiles exploitent le changement pour donner un avantage compétitif au client.
- ★ Livrer fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.

- ★ Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.
- ★ Réaliser les projets avec des personnes motivées. Fournissez-leur l'environnement et le soutien dont ils ont besoin et faites-leur confiance pour atteindre les objectifs fixés.
- ★ La méthode la plus simple et la plus efficace pour transmettre de l'information à l'équipe de développement et à l'intérieur de celle-ci est le dialogue en face à face.
- ★ Un logiciel opérationnel est la principale mesure d'avancement.
- ★ Les processus agiles encouragent un rythme de développement soutenable. Ensemble, les commanditaires, les développeurs et les utilisateurs devraient être capables de maintenir indéfiniment un rythme constant.
- ★ Une attention continue à l'excellence technique et à une bonne conception renforce l'agilité.
- ★ La simplicité – c'est-à-dire l'art de minimiser la quantité de travail inutile – est essentielle.
- ★ Les meilleures architectures, spécifications et conceptions émergent d'équipes auto-organisées.
- ★ À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence.

Le cadre méthodologique Scrum

Je vous propose maintenant de zoomer sur l'une des méthodes Agile existantes afin de vous montrer plus concrètement le fonctionnement. Pourquoi traiter de Scrum en particulier ? Tout simplement parce que Scrum est de très loin la méthodologie la plus utilisée parmi les méthodes agiles existantes. Elle est donc la plus éprouvée, documentée et supportée. Livres, blogs, formations, vidéos, associations, conférences traitant de Scrum ne manquent pas et bon nombre de ces ressources sont accessibles gratuitement. On pourrait pratiquement parler d'un standard Agile. Un autre atout important : Scrum est simple à comprendre. Sa maîtrise est en revanche difficile.

Le "package" Scrum



Processus Scrum

Scrum est considéré comme un cadre ou « framework » de gestion de projet. Ce cadre est constitué d'une définition des rôles, de réunions et d'artefacts.

Scrum définit 3 rôles :

- Le « Product Owner » qui porte la vision du produit à réaliser (représentant généralement le client).
- Le « Scrum Master » garant de l'application de la méthodologie Scrum.
- L'équipe de développement qui réalise le produit.

La vie d'un projet Scrum est rythmée par un ensemble de réunions clairement définies et strictement limitées dans le temps (timeboxing):

- Planification du Sprint (Sprint = itération) : au cours de cette réunion, l'équipe de développement sélectionne les éléments prioritaires du « Product Backlog » (liste ordonnancée des exigences fonctionnelles et non fonctionnelles du projet) qu'elle pense pouvoir réaliser au cours du sprint (en accord avec le « Product Owner »).
- Revue de Sprint : au cours de cette réunion qui a lieu à la fin du sprint, l'équipe de développement présente les fonctionnalités terminées au cours du sprint et recueille les feedbacks du Product Owner et des utilisateurs finaux. C'est également le moment d'anticiper le périmètre des prochains sprints et d'ajuster au besoin la planification de release (nombre de sprints restants).
- Rétrospective de Sprint : la rétrospective qui a généralement lieu après la revue de sprint est l'occasion de s'améliorer (productivité, qualité, efficacité, conditions de travail, etc) à la lueur du "vécu" sur le sprint écoulé (principe d'amélioration continue).
- Mêlée quotidienne : il s'agit d'une réunion de synchronisation de l'équipe de développement qui se fait debout (elle est aussi appelée "stand up meeting") en 15 minutes maximum au cours de laquelle chacun répond principalement à 3 questions : « Qu'est ce que j'ai terminé depuis la dernière mêlée ? Qu'est ce que j'aurai terminé d'ici la prochaine mêlée ? Quels obstacles me retardent ? »

L'organisation générale

Travaux préparatoires

L'approche Scrum propose de commencer par lister les exigences du client afin de produire le « Product Backlog ». Voir l'exemple ci dessous pour la réalisation d'un site d'e-commerce :

ÉLÉMENT DU BACKLOG	ESTIMATION
Un internaute peut rechercher un article selon différents critères	5
Un gestionnaire du catalogue de produits peut ajouter des articles	2
L'internaute peut acheter en ligne un ou plusieurs articles	3
...	...

L'unité de coût (ou complexité) de la colonne "Estimation" est arbitraire, on procède généralement par relativité en définissant un étalon de base. Par exemple, "voir le détail d'un article" étant une exigence simple, elle servira d'étalon et son estimation convenue sera par exemple de "1 point", "modifier les caractéristiques d'un article" étant 2 fois plus compliquée, son estimation sera de "2 points", etc. Le recours à une telle unité (plutôt que des jh ou €) permet de faciliter l'ordonnancement du Product Backlog, la planification des sprints et des releases. D'autre part il souligne le fait qu'il ne s'agit que d'une estimation (par définition fausse) et non pas un chiffrage en tant que tel.

Le Product Owner ordonnance ensuite la liste en fonction de la valeur ajoutée métier, du coût estimé de chaque exigence et des risques identifiés. Les exigences seront réalisées dans l'ordre ainsi défini selon les

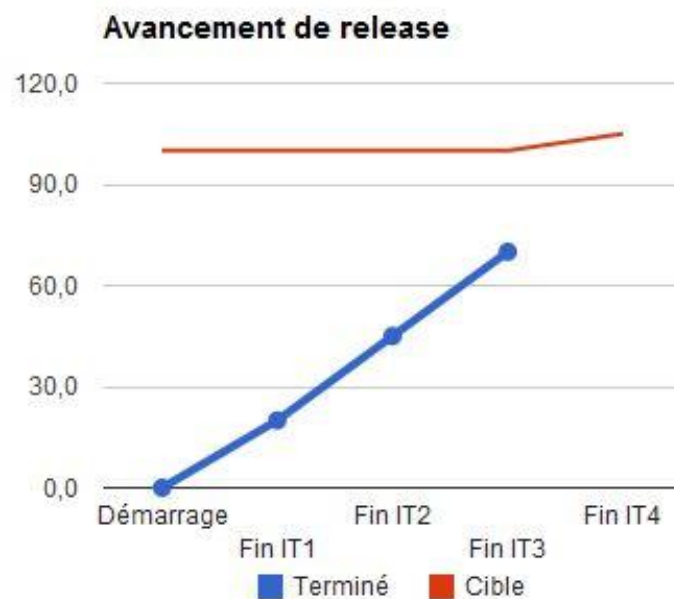
contraintes de l'équipe de développement et les éventuelles dépendances (exigence D à faire avant l'exigence X). On fixe ensuite la durée des sprints durant laquelle un certain nombre d'éléments du « Product Backlog » seront réalisés. L'objectif de Scrum consiste à produire le plus tôt possible la plus grande valeur possible, afin de créer des opportunités d'accélération du "Time to market".

Enchaînement des sprints

Une fois que le Product Backlog est prêt et que la durée du sprint est fixée en accord avec le client, il n'y a plus qu'à remplir le sprint avec des éléments du Product Backlog (planification de sprint). C'est également à ce moment que le Product Owner exprime plus précisément son besoin (qu'il aura affiné au préalable) pour permettre à l'équipe de développement d'estimer plus précisément la charge de travail du sprint. Inutile pour autant de réaliser la conception détaillée en séance, des ateliers dédiés pourront avoir lieu en cours de sprint. Le Product Owner peut à tout moment revoir la priorité des exigences qui n'ont pas encore été planifiées dans le sprint en cours. En revanche, les exigences engagées dans le sprint en cours sont "sanctuarisées", seule l'équipe de développement a le pouvoir de modifier le périmètre du sprint en cas d'avance ou de retard.

Chaque sprint se termine par la revue de sprint suivie de la rétrospective. Le sprint suivant s'enchaîne à la suite selon le même cycle et ainsi de suite jusqu'au dernier sprint de la release.

Mesure de l'avancement



Exemple de graphique d'avancement de release.

Grâce aux estimations individuelles des exigences du « Product Backlog » ainsi qu'à la segmentation en sprints, on peut aisément produire un graphique de suivi d'avancement représentant l'évolution du travail accompli en fonction du temps (voir illustration ci contre : total de "points" d'estimation des exigences terminées en bleu et charge totale de "points" de la release en rouge).

La contractualisation agile

La contractualisation d'un projet agile n'est pas la partie la plus facile étant donné que le périmètre est par définition variable. La régie ferait bien l'affaire mais difficile de rassurer le client avec un tel contrat. En France et ailleurs, le contrat au forfait domine; surtout sur les gros projets. Malheureusement pour le fournisseur - dans le cadre d'un forfait classique - tous les risques sont pour lui (aussi bien sur un projet agile que sur un projet traditionnel). On peut limiter ces risques en prenant quelques précautions, mais on limite également la souplesse offerte par une approche Agile.

Cela ne veut pas dire qu'il n'existe pas de solutions. La forfaitisation de chaque itération avec la possibilité pour le client d'arrêter le contrat à la fin de chaque itération est assez intéressante. Ainsi que le principe de troc d'exigence : réalisation d'une exigence imprévue en échange du retrait d'une autre moins importante, de priorité faible et de même coût.

Voici quelques liens qui traitent du sujet :

- [Contrat Agile proposé par Xebia](#)
- [Différents types de contrats proposés pour des projets agiles](#)

Envie d'aller plus loin ?

Visitez la rubrique "[Par où commencer ?](#)". Vous y trouverez par exemple, le [guide de démarrage Scrum](#) ainsi que des vidéos d'introduction.

Découvrez les formations et services proposés par Florent Lothon : <http://www.agiliste.fr/formations/>.

A propos de l'auteur

Florent Lothon

FLORENT LOTHON est expert en transformation agile, qu'il met en pratique depuis 2007. Il partage son expérience à travers son site <http://agiliste.fr> et ses formations certifiantes.

« Les méthodes agiles ont définitivement changé ma façon de mener et vivre un projet. J'ai créé le site L'Agiliste dans le but de partager cette expérience avec un maximum de monde. » - Florent Lothon



Du même auteur

Le livre "Devenir une Entreprise Agile".

Vous avez maintenant une bonne idée du potentiel que peut offrir l'agilité dans un contexte projet. Imaginez maintenant ce que l'agilité peut apporter à l'entreprise toute entière. Imaginez les atouts dont disposent les entreprises agiles dans le monde d'aujourd'hui. Un monde devenu complexe, incertain, avec de nouvelles générations de travailleurs aux attentes affirmées.

Découvrez la culture et les pratiques des entreprises agiles, au-delà de la gestion de projet agile. Intégrez l'agilité dans vos pratiques professionnelles pour transformer votre quotidien et votre entreprise.

[Lien pour acheter ce livre](#)

