

# Des approches agiles remarquables

## 1. La méthodologie Scrum

La méthode Scrum, élaborée à partir de 1986, s'est répandue et imposée depuis quelques années comme étant la référence dans la mise en œuvre de projets agiles. Il s'agit d'une approche holistique qui procède par cycles courts (des *sprints* ou itérations) pour organiser le développement logiciel de façon incrémentale et itérative. Les fonctionnalités sont réalisées progressivement et livrées au fur et à mesure.

La terminologie Scrum est quasiment passée dans le langage courant, elle est appliquée par extension à d'autres méthodes agiles.

### a. La définition du backlog

Le **backlog** représente l'ensemble des entrants fonctionnels comme techniques à réaliser. Il s'agit d'un stock continuellement mis à jour, chaque *sprint* étant l'occasion de puiser des éléments à implémenter ou d'en ajouter d'autres.

L'élément de travail de base est l'histoire utilisateur et son pendant l'histoire technique, respectivement désignées **user story** (US) et **technical story** (TS). Une US décrit la fonctionnalité telle qu'appréhendée par l'utilisateur. Il n'est pas nécessaire de faire figurer dans l'US des instructions techniques relatives à l'implémentation, bien que cela soit une possibilité. L'élément de travail comporte un titre, une référence unique déterminée par le système de gestion (Jira, VSTS...), une description fonctionnelle, mais aussi des critères d'achèvement ou *definition of done*.

De façon à cartographier le système logiciel en devenir et à structurer les US dans un ensemble d'éléments de travail parfois vaste, les systèmes de gestion introduisent des surensembles d'US tels que les **epics** et les **features**.

Le *backlog* comprend aussi des bugs remontés lors des cycles de tests ou par les utilisateurs.

Le travail de rédaction des US emprunte lui-même un processus itératif et incrémental. Il est habituel de lister les US dans le *backlog*, puis de les compléter au fil des *sprints*.

Bien que, par essence, Scrum ne fixe aucune règle quant à la structuration des *sprints* (autrement dit, l'ordre et la quantité d'US à traiter ne sont pas déterminés par la méthode), de nouvelles approches visent à organiser le découpage du projet pour atteindre plus rapidement et plus facilement la satisfaction client. La cartographie des histoires utilisateurs (ou **story mapping**) représente en deux dimensions les US, par séquence de fonctionnalités utilisateurs et par ordre d'importance métier. Cette approche aide l'équipe projet à se préoccuper en priorité de ce qui est important et impactant pour la satisfaction du client.

## b. Les sprints ou itérations

Les *sprints* structurent les projets Scrum. Ce sont des cycles d'activités généralement assez courts - deux semaines - où se succèdent des réunions appelées cérémonies et des phases de production logicielle.

Il existe sans doute autant de façons d'appliquer Scrum que d'équipes et de projets, et autant de possibilités de personnaliser les cérémonies, d'en introduire de nouvelles ou d'en écarter certaines. Cela dit, quatre d'entre elles restent des incontournables de la méthode Scrum.

En début de *sprint*, au cours de la séance d'entretien du *backlog* (*grooming*) et de conception du *sprint* (*sprint planning meeting*) sont sélectionnées les US à engager dans le cycle courant.

En début de chaque journée a lieu la réunion quotidienne (*daily standup meeting* ou DSM) au cours de laquelle chaque membre de l'équipe énonce son avancement, ses difficultés et son programme pour la journée. Le *Scrum master* est le secrétaire de séance, il guide ses équipiers dans les courts débats qui s'engagent.

La revue d'itération (*sprint review*) est une démonstration par l'équipe et au client des livrables réalisés pendant le *sprint*. C'est donc une réunion fondamentale puisque le client avec le *product owner* formulent le degré de satisfaction vis-à-vis du produit tout juste retravaillé, et donnent des orientations pour le *sprint* qui succédera.

La rétrospective (*sprint retro*), qui peut avoir lieu en même temps que la revue, met l'accent sur ce qui a fonctionné et sur ce qui pourrait être amélioré, dans un registre technique, organisationnel, communication... La rétrospective est une caractéristique qui illustre la philosophie holistique du Scrum standard, en cherchant l'amélioration continue dans le

cadre du projet lui-même.

### c. L'avancement dans Scrum

Revenons un instant sur l'organisation des flux de travail dans un projet agile ; le planning de type Gantt ne s'emploie pas, car les tâches ne sont pas estimées en j.h, mais en points de complexité.

L'équipe a déterminé en début de *sprint* combien de points de complexité pouvaient être engagés conformément à sa vélocité. Le suivi des tâches s'effectue à l'aide d'un tableau divisé en colonnes (À faire, En cours, À valider, Terminé...). Des étiquettes représentant les tâches (des US ou des bugs) sont déplacées d'une colonne à l'autre au fur et à mesure des traitements.

Bien que les systèmes de gestion tels que Jira ou VSTS proposent des tableaux de bord assez efficaces, il n'est pas rare que l'équipe installe aussi un véritable tableau sur lequel sont apposées des étiquettes collantes.

Afin de quantifier l'effort restant pour compléter les US du *sprint*, on utilise un *burndown chart*. Ce graphique représente en abscisse le temps (en jours, en semaines...) et en ordonnée l'effort restant. La courbe est souvent comparée à une droite qui part du point (0, Total engagé) et qui descend jusqu'au point (Fin de *sprint*, 0).

### d. Le rôle du Scrum master

La méthode Scrum ne prévoit pas de hiérarchie entre les rôles. Elle constitue un tout et son application rigoureuse est la condition sine qua non de réussite du projet.

Le **Scrum master** a la charge de l'application à bon escient de la méthode. Il a pour mission d'aider l'équipe à dérouler le projet, à s'entraider, à s'améliorer et à dépasser les difficultés d'organisation. Le *Scrum master* n'est donc pas un chef de projet ni un leader dirigiste, mais un facilitateur qui guide et coach son équipe. Cette définition du rôle du *Scrum master*, déterminé par la méthode elle-même, est une nouvelle illustration de son concept tout-en-un, et d'un accent davantage mis sur la performance de l'équipe que sur la prédictibilité du projet.

Ce lemme produit deux corollaires. D'abord, la fonction de *Scrum master* n'est pas réservée à une personne, elle peut être attribuée successivement à n'importe quel membre de

l'équipe. Ensuite, les *Scrum masters* d'une organisation peuvent se réunir dans des réunions de Scrum à Scrum (**Scrum to Scrum**) appelées méta-Scrum, destinées à solutionner les problèmes de coordination entre équipes Scrum.

## 2. Kanban

Les organisateurs de la productivité de Toyota (dont les célèbres Taiichi Ohno et Shigeo Shingo) ont élaboré patiemment le *lean manufacturing* et la méthode Kanban dans les années 1950. Jusque-là très proches du modèle tayloriste des usines américaines, leurs méthodes de production se sont démarquées par leurs performances supérieures et se sont ensuite généralisées à l'ensemble de l'industrie, bien au-delà du secteur de la construction automobile.

Une usine développe des processus de production complexes où les centres de coût et de défaut sont multiples. Le stockage des matières premières comme des produits en cours de fabrication est une première difficulté, le transport entre les chaînes d'assemblage en est une autre, sans compter les temps d'attente à provisionner une étape de la chaîne de valeur en éléments exempts de défauts.

Or, la production doit répondre à une demande de livraison qui ne peut être complètement contrôlée. Les ingénieurs du TPS (*Toyota Productivity System*) ont alors modifié l'organisation à base de flux poussé en flux tiré.

Dans l'approche historique, le stock de matières premières est assemblé en éléments déplacés à chaque étape de la construction jusqu'à devenir un autre stock de produits finis en attente d'être vendus. Ce stock coûte cher (entreposage, transport...), alors que sa valorisation peut être dépréciée selon les conditions économiques. Cette organisation pose également un autre problème. La surabondance du stock à chaque étage de la chaîne de valeur perturbe le traitement des défauts : pas le temps de rectifier un article, tout le stock intermédiaire doit être traité puis expédié à l'étape suivante.

Le parallèle avec un projet informatique est assez évident ; les goulets d'étranglement se créent de façon systémique. Même avec une méthode comme Scrum, il n'est pas prévu d'aligner la capacité des testeurs sur celle des développeurs et il faut plusieurs *sprints* de rodage pour déterminer la vitesse moyenne de l'équipe. Ce qui n'empêche pas une contre-performance lors d'un *sprint* si de nombreux bugs apparaissent au cours du codage,

révélés lors des tests, corrigés ou non dans le cadre du *sprint* en cours.

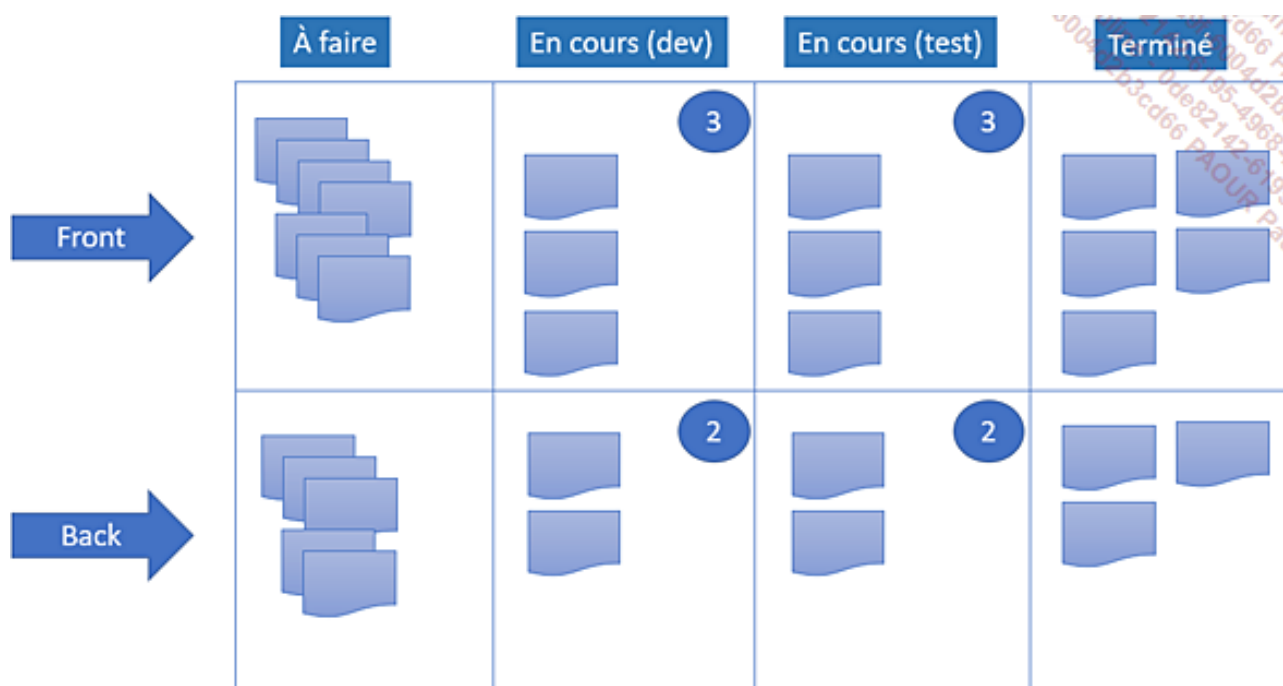
Dans l'organisation à flux tiré, les équipes déterminent et ajustent la capacité de chaque étage de la chaîne de valeur, le processus est conçu en partant de la fin. Ce changement est radical, les goulets d'étranglement systémiques sont évités, le stock intermédiaire ramené à son strict minimum, le temps d'exécution prévoit ce qu'il faut pour retraiter des éléments défectueux.

Pour satisfaire la demande fluctuante des clients sans malmener les équipes de production, les concepteurs de la méthode ont intégré l'application permanente de petits changements. Cette approche a fait la preuve d'une meilleure efficacité que de grandes révolutions ou le recours à des méthodes rigides.

### a. Une cartographie visuelle suivie par tous

Le terme « Kanban » signifie « carte visuelle » ; le procédé repose sur la connaissance par toute l'équipe de l'état du projet et des capacités de traitement. Les tableaux de bord de Scrum et de Kanban sont assez similaires, présentant en colonnes les différentes étapes de construction (À faire, En cours, À valider, Terminé).

Sur le tableau de bord de Kanban, des chiffres indiquent les capacités de chaque processus :



## b. Flux continu

L'élimination des stocks et la production juste à temps sont l'essence même du *lean management*. Appliquées aux projets informatiques, les fonctionnalités sont publiées dès qu'elles sont réalisées, les *releases* étant cadencées selon les critères de l'équipe et en fonction des besoins client.

Plus encore qu'avec Scrum, le recours à des systèmes d'intégration continue, de tests automatiques et de publication rapide est primordial dans la mise en œuvre de projets agiles au format Kanban.

## c. Pas de rôles

En tant que système d'organisation et de planification, Kanban ne définit pas de rôles. L'équipe régule le flux de traitement des tâches selon ses propres capacités et peut faire appel à un coach agile pour la guider.

Cependant, les principaux rôles d'un projet agile trouvent naturellement leur place dans le dispositif : *product management*, *delivery management*, *design management*, et bien entendu développement ou *testing*.

## d. Lead time plutôt que vélocité

Chaque format de projet possède une métrique qui le caractérise : le respect QCD pour le modèle en cascade, la vélocité pour le Scrum, et pour Kanban, qui se trouve à l'opposé des méthodes classiques, le **lead time**.

Le *lead time* exprime le temps entre la commande et la livraison, délai que l'on souhaite le plus court possible pour éviter le risque d'un changement, voire d'une annulation. Cette métrique est à rapprocher du **takt time** qui exprime la pression des attentes client (rythme de livraison attendu).

Dans le cas de Scrum, la vélocité est une mesure de la performance de l'équipe. Elle est donc plutôt similaire au **temps de cycle**, qui traduit le délai entre la première et la dernière opération pour réaliser un produit fini. Le temps de cycle comme la vélocité dépendent de la capacité de l'équipe à livrer dans un contexte donné, alors que le *lead time* est résolument destiné à maximiser la satisfaction du client.

### e. Des changements à tout moment

Dans un projet réalisé avec un niveau de charge assez élevé, les équipes ont un réflexe répandu qui consiste à avancer au maximum la production intermédiaire, en menant toutes les activités en parallèle. Ce qui est fait n'est plus à faire, entend-on souvent.

En fait, loin d'être efficace, cette approche paralyse le système de production, qui n'admet plus aucun changement de priorité et qui ne trouve jamais la ressource nécessaire pour se réformer.

À commencer par les exigences client dont le flux ne peut être totalement contrôlé, les changements de priorité surviennent constamment. Là encore, Kanban se distingue de Scrum par sa dynamique de production tournée vers le client (le flux tiré) et sa production juste à temps (*just in time* ou JIT). L'écart entre la capacité de production nominale et la demande client constitue un gisement de temps inépuisable pour s'améliorer ou ajuster la production à la demande instantanée.