## Les méthodes agiles

Face au besoin de flexibilité dans le processus de création d’un logiciel, les méthodes en cascades sont progressivement délaissées face aux méthodes itératives. Pour rappel, les méthodes en cascade correspondent à un processus monolithique où toute la conception et le développement sont définis au départ et est difficilement modifiable. A l’inverse, les méthodes itératives correspondent à un processus évolutif qui laisse une grande liberté dans la mise en œuvre du projet avec des navettes à chacune de ses étapes. L’avantage des méthodes agiles réside dans une meilleure réponse au besoin client.

## Les 4 valeurs de la méthode agile

* Privilégier les personnes et les interactions entre elles
* Privilégier les fonctionnalités du logiciel plutôt que la documentation
* Favoriser la collaboration avec le client plutôt que la négociation de contrat
* Être réactif au changement plutôt que suivre un plan établi à l'avance

## Les 12 principes des méthodes agiles

1. Livraison rapide et continue de logiciel.
2. Possibilité de changements tardifs dans le développement.
3. Livrer du logiciel fréquemment, de 2 semaines à 2 mois.
4. Les acteurs métier et les développeurs doivent travailler ensemble quotidiennement pendant toute la durée du projet.
5. Construire les projets autour de personnes motivées. Leur donner l’environnement nécessaire, répondre à leurs besoins et faites leur confiance.
6. Le moyen le plus efficace de véhiculer l’information vers et à l’intérieur d’une équipe de développement est la conversation en face à face.
7. La métrique principale pour juger de la progression d’un projet est le logiciel fonctionnel.
8. Les processus agiles encouragent le développement durable. Les financeurs, les développeurs, et les utilisateurs doivent maintenir un rythme constant indéfiniment.
9. Une attention continue à l’excellence technique et au bon design améliore l’agilité.
10. La simplicité –L’art de maximiser le montant de travail non fait—est essentiel.
11. Les meilleures architectures, besoins, et conceptions proviennent des équipes auto organisées.
12. A intervalles réguliers, l’équipe réfléchit aux manières de devenir plus efficace, puis ajuste ses comportements de façon à s’y conformer.

## Qu’est-ce que SCRUM ?

Un projet utilisant Scrum est composé d’une suite d’itérations courtes de l’ordre de 3 à 6 semaines appelées sprints. Le projet peut être réorienté par le client à la fin de chaque sprint et uniquement à ces moments-là.

## Les principes clés de Scrum

* Auto-organisation et pouvoir de décision donnée à l’équipe
* Sprints en isolement avec le client et le product owner
* Délais fixes
* Réunion quotidienne de 15 minutes, le daily scrum, avec une liste de questions prédéfinies
* Démonstration du résultat du sprint
* Planning adaptatif en fin de sprint

## Les rôles dans Scrum

Le product owner (directeur produit) gère l’interface entre le client et l’équipe de développement. Il a pour mission de :

* Représente toutes les personnes ayant un intérêt dans le logiciel produit.
* Finance le projet.
* Est responsable de l’établissement des besoins initiaux du projet qui sont listés dans le backlog produit (catalogue des tâches à effectuer).
* A des objectifs de retour sur investissement.
* Décide des releases (versions) qui seront mises à disposition des utilisateurs.
* Est responsable de la mise à jour du backlog produit.

Le scrummaster (maître de mêlée) est la référence de la méthode scrum et s’attache à ce que l’équipe applique les principes de cette méthode. Son rôle est de :

* Isoler l’équipe des perturbations extérieures en cours de sprint.
* Apprendre la méthodologie Scrum à l’équipe
* S’assurer que l’équipe respecte bien la méthode.
* Répondre aux besoins de l’équipe.
* Faire cadrer Scrum dans la culture d’entreprise.
* Travailler avec le product owner pour sélectionner avec lui les besoins apportant le plus de valeur ajoutée au logiciel. Il est responsable de la réussite du projet, au même titre que l'équipe.

L’équipe est constituée des développeurs ayant une liberté d’action dans leurs tâches et dans laquelle il n’y a plus de chef de projet.

## Gestion du projet en méthode scrum

1. Le client souhaite un produit en collaboration avec le product owner
2. Le product owner définit le backlog produit qui correspond à toutes les fonctionnalités à développer. Les fonctionnalités sont affectées d’un poids correspondant à leur importance.
3. Le product owner présente à l’équipe le travail à faire et s’accorde avec l’équipe pour définir un backlog de sprint, qui est un sous-ensemble du backlog produit.
4. Lors de la planification du sprint, l’équipe divise le backlog de sprint en petites tâches à réaliser d’une durée comprise entre ½ et 2 jours.
5. Chaque jour, un daily scrum est organisé.
6. Chaque sprint est ponctué par une revue de sprint. Elle réunit l’équipe, product owner et les personnes ayant un intérêt dans le projet et a pour but de présenter les résultats du sprint.
7. Enfin, une réunion post-mortem du sprint est tenue entre les membres de l’équipe de développement dans le but d’identifier les éléments pouvant être améliorer pour le sprint suivant.

## Daily Scrum

Le daily scrum est une réunion quotidienne permettant d’échanger sur le sprint en cours. Chaque participant répond à 3 questions :

1. Qu'as-tu fait hier ?

2. Que fais-tu aujourd'hui ?

3. Quels sont les problèmes qui t'empêchent de progresser ?

## Les outils du sprint

* Le tableau blanc divisé en 3 parties : les besoins et tâches qui ne sont pas encore pris en charges, ceux qui sont en cours de réalisation et ceux qui sont achevés. Les besoins issus du backlog produit sont écrits sur des post-its de couleur blanche et la décomposition des besoins en tâches sur des post-its de couleur jaune. Le burndown chart est un graphique tracé en supplément et permet de visualiser l’avancée de l’équipe dans son sprint. Par ailleurs, si l’équipe découvre que des tâches non-envisagées initialement lors de la planification du sprint doivent être réalisées, il est ajouté des post-its jaunes dans la section *unplanned items*. Enfin, une section *next* est prévu pour motiver l’équipe à réaliser des besoins supplémentaires si le sprint se déroule plus rapidement que prévu.