



# LA MANU

L'ÉCOLE DES MÉTIERS DU NUMÉRIQUE  
Manufacture de compétences



## SUPPORT APPRENANT

*Méthodes Agiles*



**CONFIDENTIEL**

*Ce document est strictement confidentiel et ne doit pas être diffusé  
sans accord préalable écrit*

## Méthodes Agiles

### Ce que l'on a vu...



Les Méthodes Agiles caractérisent une approche de gestion de projets d'abord mise en place pour le développement web et informatique, mais qui est aujourd'hui de plus en plus répandue car elles peuvent s'adapter à de nombreux types de projets, tous secteurs confondus. Les méthodes agiles sont des groupes de pratiques de pilotage et de réalisation de projets. Elles sont basées sur l'expérience, elles sont plus souples et s'adaptent aux besoins du client qui sont placés au centre des priorités du projet.

Le terme « Agile » apparaît, en 2001, avec le Manifeste Agile qui énonce les valeurs et les 12 principes qui régissent les méthodes agiles.

Le **Manifeste Agile** a été rédigé en 2001, à la suite de la rencontre de 17 experts en développement logiciel pour trouver une solution permettant de palier au taux d'échec élevé des projets dans les années 90.

Les approches traditionnelles (ou cycle V) prévoient l'élaboration d'un cahier des charges exhaustif exigeant du client une expression détaillée et la validation du projet dès le départ. Cela peut avoir pour conséquences de laisser peu de place aux changements et peut entraîner l'insatisfaction du client si sa définition du projet n'a pas été assez claire et ne correspond finalement pas à ce qu'il attendait.

L'approche Agile permet de réduire « l'effet tunnel » (élaboration d'un cahier des charges – développement du projet – livraison du projet final) de l'approche traditionnelle, en impliquant le client du début à la fin du développement du projet. Cela confère plus de visibilité à ce dernier et laisse également plus de place aux éventuelles modifications en cours de développement.

Faiblesses du Cycle V	Réponses de l'Agilité
Le cahier des charges est supposé exhaustif dès le départ	Le cahier des charges est élaboré au fur et à mesure
Les modifications sont mal accueillies	Le projet est une succession de modifications
Décourage l'amélioration continue	L'équipe est constamment challengée et participe à l'amélioration continue
La communication est peu fluide	La communication est constante

Une enquête du « Standish Group » en 1994 fait le constat suivant :

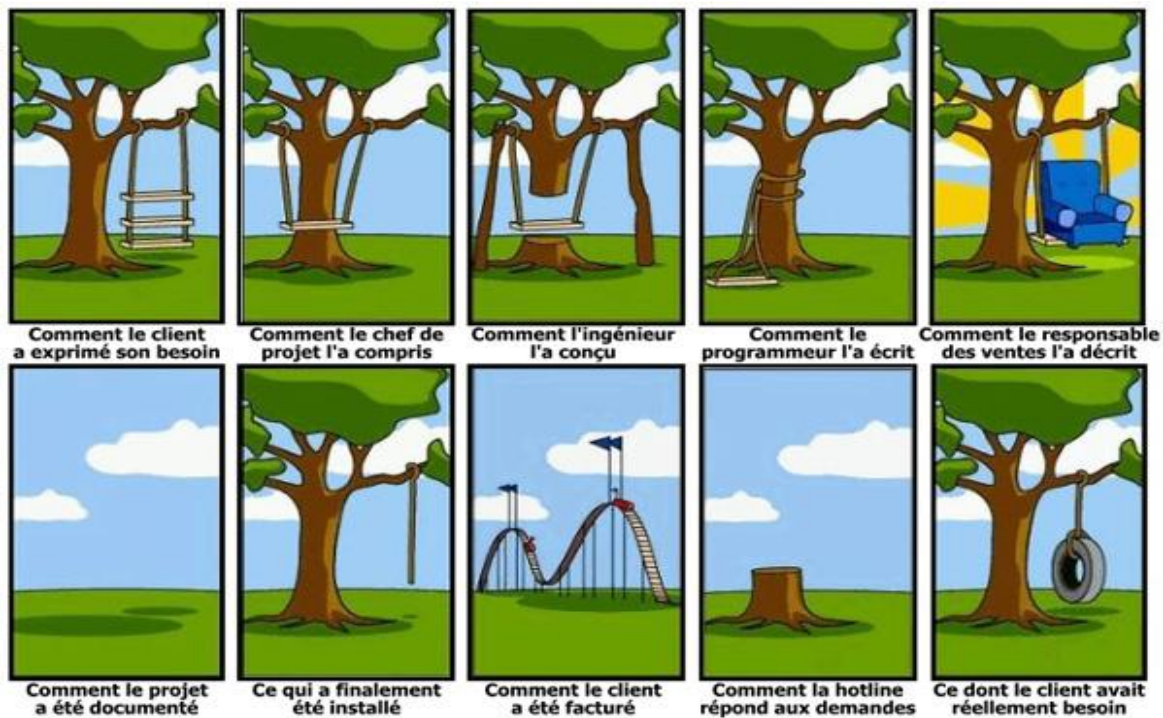
- 16% des projets sont considérés comme un succès,
- 32% des projets sont arrêtés en cours de route,
- 52% des projets n'aboutissent qu'au prix d'un important dépassement des délais et du budget tout en offrant moins de fonctionnalités.

Les motifs de ces échecs sont principalement le manque d'implication des utilisateurs finaux (12.8%) et des changements de spécifications en cours de projet (11.8%).

La même enquête réalisée en 2008, indique une augmentation du taux de réussite. Les projets considérés comme des succès atteignent les 35%. On peut penser que l'application progressive des Méthodes Agiles dans les gestions de projet est une raison de cette augmentation.

Ainsi, adopter une approche Agile c'est offrir le maximum de chance au projet d'aboutir et de répondre aux exigences du client et des utilisateurs finaux.

Illustration de la perception d'un projet selon différents points de vue :



## SCRUM

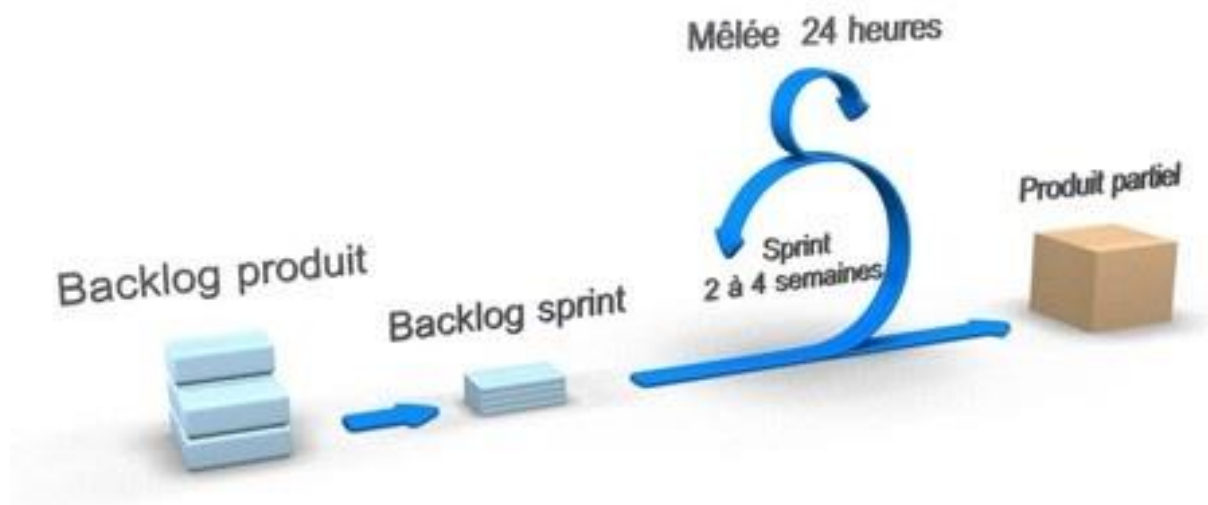
Scrum est un cadre méthodologique de gestion de projet Agile qui a pour objectif d'améliorer la productivité d'une équipe. Scrum est aujourd'hui, la méthodologie la plus utilisée dans le monde parmi les différents types de Méthodes Agiles.

Dans ce cadre de gestion de projet Agile, on retrouve notamment des rôles précis pour les différents acteurs d'un projet :

- Le **Scrum Master** : il est le garant de l'application de la méthodologie Scrum, il s'assure que les principes et les valeurs des Méthodes Agiles sont bien respectées, il facilite la communication au sein de l'équipe et avec le Product Owner, il cherche à améliorer la productivité et le savoir-faire de son équipe.
- La **Team** : elle peut être composée d'architectes, de développeurs, de graphistes/ergonomes, d'ingénieurs système, de testeurs... chaque membre de l'équipe apporte son savoir-faire. C'est elle qui réalise le produit.
- Le **Product Owner** : il représente le client et porte la vision du produit à réaliser, il s'agit généralement d'un expert du domaine métier du projet. Il définit les spécifications fonctionnelles, établit la priorité des fonctionnalités à développer ou corriger, et valide les fonctionnalités réalisées.

La méthodologie Scrum se veut souple. Ses acteurs peuvent toujours ajuster leur propre méthode au fil des itérations, en fonction de leur expérience et selon le contexte. De plus, Scrum peut tout à fait être complété par d'autres méthodes.

Le déroulé type de la méthodologie Scrum repose sur différentes étapes.



### Atelier (= Backlog product)

**Participants :** Product Owner, Scrum Master

L'**Atelier** se déroule avant la vente et permet de se mettre d'accord sur ce qui peut être fait. On fait le point sur les attentes du Product Owner et sur le temps que l'on estime pour réaliser le projet (attention, on ne définit pas de calendrier précis mais estimatif). Pour faire cela le Scrum Master et le Product Owner définissent les fonctionnalités du projet et rédigent les User Stories.

Une **User Story** est la description d'un besoin / d'une fonctionnalité en répondant aux questions :

- QUI ?
- QUOI ?
- COMMENT ?
- POURQUOI ?

Exemple :

Expression classique d'un besoin	User Story
Création d'un formulaire pour que l'administrateur enregistre un nouveau client	<u>En tant qu'administrateur, je peux inscrire un nouveau client en remplissant le formulaire de la page d'ajout pour enregistrer ses données dans la base de données</u>

Plus une User Story est détaillée plus elle sera claire pour tous les acteurs du projet. La difficulté de réalisation de chaque User Story est évaluée par le Scrum Master et le Product Owner. Ensemble ils définissent une estimation du temps de réalisation de chaque User Story. Cela peut être fait avec le **Planning Poker** par exemple. Les User Story doivent autant que possible ne pas dépasser une journée de développement. Cependant, si tel est le cas il est possible de diviser la User Story en plusieurs afin de faire des étapes intermédiaires.

### Challenge technique (optionnel)

Lorsque la réalisation du projet nécessite des compétences ou du matériel dont l'équipe de production ne dispose pas, il est alors possible d'organiser un **Challenge technique**.

Le Challenge technique est ainsi une phase où l'équipe se forme à une technologie ou acquiert le matériel nécessaire pour le projet. Cette phase peut survenir dès la fin de l'atelier et peut se produire plusieurs fois en fonction des besoins du projet. Il faut prévenir le Product Owner dès qu'une difficulté nécessitant de faire un Challenge technique est soulevée, car cela peut décaler la période de livraison

du projet. Ainsi informé, il décidera d'accorder le Challenge technique à l'équipe ou bien ils opteront pour une autre solution possible ne nécessitant pas de compétences supplémentaires.

### **Cérémonie (= Backlog sprint)**

**Participants :** Product Owner, Scrum Master, Team

**Durée :** 2h par semaine de sprint (ex : 4h pour un sprint de 2 semaines)

Au cours de la **Cérémonie**, on priorise les différentes User Stories, cela peut être fait avec la **méthode MoSCoW** par exemple. Puis le Product Owner définit les priorités pour attribuer des User Stories au sprint à venir et les répartir au sein de l'équipe. L'estimation du temps de réalisation des User Stories retenues est revue et confirmée (ou ajustée) avec l'équipe.

La somme du temps estimatif des User Stories allouées à un membre de l'équipe doit être inférieure ou égale au temps du sprint, c'est-à-dire que pour un sprint de 2 semaines les tâches affectées à un membre ne doivent pas dépasser les 2 semaines. Avoir des objectifs réalistes et atteignables évite les découragements de certains membres qui pourraient se sentir dépassés et permet ainsi de motiver l'équipe.

### **Sprint (= Développement)**

**Participants :** Scrum Master, Team

**Durée :** 2 à 4 semaines

Le **Sprint** est la phase de développement au cours de laquelle les User Stories attribuées à ce Sprint sont réalisées.

### **Stand-up Meeting (= Mêlée)**

**Participants :** Scrum Master, Team

**Durée :** 15 minutes

Au cours d'un Sprint, on procède à une **mêlée** ou **stand-up meeting** quotidiennement. Cette réunion concise (maximum 15-30 minutes) permet aux membres de l'équipe de remonter leurs blocages, difficultés, questions, réussites, et permet également de suivre l'avancée du Sprint. En effet, chaque membre répond à 3 questions : « Qu'ai-je fait hier ? Que vais-je faire aujourd'hui ? Quels obstacles pourraient m'empêcher d'avancer ? ». La mêlée quotidienne contribue à l'esprit d'équipe. Elle est animée par le Scrum Master qui doit s'efforcer de lever les éventuels obstacles afin de garder l'équipe soudée et concentrée sur l'objectif.

### **Sprint review**

**Participants :** Product Owner, Scrum Master, Team

**Durée :** 1h par semaine de sprint (ex : 2h pour un sprint de 2 semaines)

Le **Sprint review** se situe à chaque fin de sprint. L'équipe présente les fonctionnalités fonctionnelles au Product Owner ce qui lui permet de valider ou refuser les User Stories développées. C'est aussi au cours du Sprint review que l'équipe va comparer les estimations de temps de développement des User Stories avec le temps réellement passé pour leur développement, grâce à cela les futures estimations pourront être améliorées ou ajustées.

Si le projet contient encore des User Stories à développer, une rétrospective et une nouvelle Cérémonie auront lieu pour organiser le prochain Sprint.



Si toutes les User Stories ont été développées et qu'elles sont validées par le Product Owner lors du Sprint review, l'ensemble du projet pourra ensuite être livré au client.

### Rétrospective

**Participants :** Scrum Master, Team

**Durée :** 45 minutes par semaine de sprint (ex : 1h30 pour un sprint de 2 semaines)

La rétrospective permet de faire un retour sur le Sprint précédent et de suivre ainsi l'évolution du projet. Elle est animée par le Scrum Master et elle a pour but l'amélioration continue du processus de développement en faisant notamment ressortir les idées de chacun pour y parvenir. Le sujet de la rétrospective n'est forcément technique, en effet, tous les domaines peuvent être abordés (humain, organisation, pratiques, outillage, qualité de vie au travail, interactions avec le Product Owner...). Il existe différentes techniques d'animation de rétrospectives qui sont aussi variées que diverses.

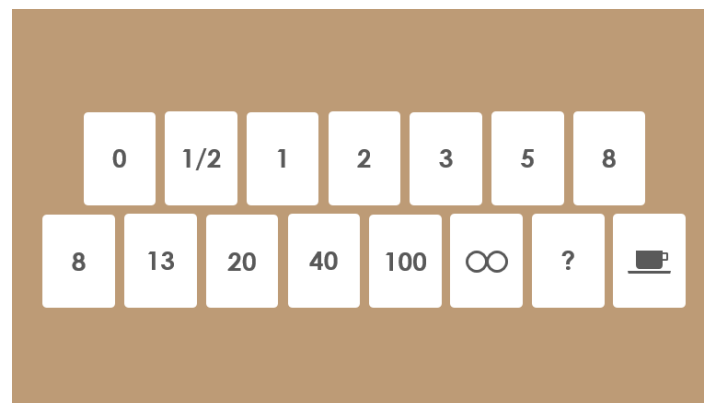
Après la rétrospective, l'équipe peut repartir du bon pied pour une nouvelle itération (Cérémonie + Sprint suivant).

### Pour compléter...

#### Le planning poker

Le planning poker est un jeu de carte permettant de mieux appréhender la complexité d'une tâche et donc le temps qu'il faudra pour la réaliser.

Un jeu comporte 4 séries de cartes comprenant chacune des cartes allant de 0 à 100 (story points), un « ? » et un symbole de l'infini.



Le planning poker se déroule sur 4 étapes :

- 1- Le **Product Owner** présente l'User Story qui doit être estimée.  
*Exemple : En tant qu'administrateur je peux ajouter un produit en remplissant le formulaire dédié pour l'ajouter à la base de données.*
- 2- Les **estimateurs** (le **Scrum master** et le **Product Owner**) discutent de l'User Story pour ajouter plus de précisions si nécessaire.  
*Exemple : combien d'informations faut-il renseigner, y a-t-il des champs spécifiques selon les catégories de produits ?*
- 3- Les **estimateurs** estiment la difficulté / complexité de la tâche et choisissent une carte pour la représenter.

- 🌿 o : facile / déjà en place
- 🌿 100 : très difficile mais toujours faisable
- 🌿 ∞ : trop difficile, besoin d'une acquisition de connaissances
- 🌿 ? : ne sait pas, incertain, trop peu d'informations données

D'autres cartes existent, telle que la pause-café par exemple, qui signifie que l'équipe a besoin d'une pause.

C'est une étape de discussion, on débat sur les différentes cartes choisies par l'interlocuteur ce qui permet de soulever les besoins éventuels de formation ou de soutien technique.

A la fin de cette étape, on se met d'accord pour une estimation collective.

4 – On convertit les story points en temps, il s'agit de la vitesse.

*Exemple : si on estime qu'une User Story vaut 21 story points et que la vitesse est de 1, alors le temps de développement est de 21 heures.*

### La méthode MoSCoW

C'est une méthode de priorisation des tâches.

M (Must have) – Ce qui doit être fait (nécessaire).

S (Should have) – Ce qui devrait être fait (essentiel).

C (Could have) – Ce qui pourrait être fait (confort).

W (Won't have this time) – Ce qui sera fait plus tard (accessoire).

Les o servent uniquement à la prononciation de l'acronyme.

### Liens utiles

<https://manifesteagile.fr/>

<https://agilemanifesto.org/iso/fr/principles.html>

<https://www.cs.nmt.edu/~cs328/reading/Standish.pdf>

<https://www.piloter.org/projet/methode/methode-agile.htm>

<https://www.planzone.fr/blog/quest-ce-que-la-methodologie-agile>

<https://agiliste.fr/introduction-methodes-agiles/>

<https://agiliste.fr/guide-de-demarrage-scrum/>

<https://blog.trello.com/fr/methode-agile-scrum-gestion-projet>

<https://www.thierry-pigot.fr/scrum-en-moins-de-10-minutes/>

### Bonnes pratiques

Rappel des principales bonnes pratiques	
<input type="checkbox"/>	Communication, remonter les points positifs et négatifs pour capitaliser.
<input type="checkbox"/>	Adaptabilité, ne pas hésiter à modifier le cahier des charges si nécessaire.
<input type="checkbox"/>	Répondre aux questions Qui ? Quoi ? Comment ? Pourquoi ? pour rédiger les User Stories.
<input type="checkbox"/>	Faire une estimation de réalisation de chaque User Story réaliste et au plus juste avec le temps de développement nécessaire.