# Définition des concepts Agile

## Table des matières

# Rôles

## Product Owner

Source : https://agiliste.fr/exemple-dorganisation-projet-agile/

Le Product Owner est le représentant du client, il :

- Définit les fonctionnalités du produit.
- Décide des dates de livraison et de leur contenu.
- Est responsable de la rentabilité du produit (ROI).
- Priorise les fonctionnalités en fonction de la valeur métier.
- Ajuste les fonctionnalités et leur priorité avant chaque planification d'itération.
- Accepte ou rejette les fonctionnalités réalisées.
- Anime la réunion de planification de sprint.

Le rôle du Product Owner est assuré par un membre de l'équipe métier (MOA) et au besoin une assistance (AMOA).

## Equipe d'assistance au Product Owner

Elle assiste le Product Owner et à ce titre peut intervenir dans le cadre des activités associées aux responsabilités de ce dernier. Plus concrètement, elle :

- Alimente et maintient le Product Backlog, et pré priorise les éléments de ce dernier.
- Ajuste les fonctionnalités prioritaires du Product Backlog (candidates au périmètre du prochain sprint) et s'assure que les pré-requis aux développements associés seront disponibles en temps voulu (exemples : décisions métier, jeux de données du SI amont,…).
- Rédige les User Stories associées aux fonctionnalités prioritaires et dessine au besoin des maquettes d'écran.
- Répond aux questions soulevées par l'équipe de développement en cours de Sprint et complète au besoin les User Stories associées.
- Vérifie en cours de Sprint la bonne couverture du besoin des fonctionnalités terminées en collaboration avec l'équipe de développement.
- Rédige les plans de tests.
- Participe à la réunion de revue de sprint au cours de laquelle, elle aide le Product Owner à accepter ou rejeter les fonctionnalités présentées.
- Teste avant mise en production la conformité du produit dans son ensemble.

## Scrum Master

Le Scrum Master appartient à l'équipe de développement (MOE), il :

- S'assure que l'équipe est pleinement opérationnelle et productive.
- Établit une collaboration étroite entre l'ensemble des rôles et fonctions.
- Supprime les obstacles rencontrés par l'équipe de développement.
- Protège l'équipe des interférences extérieures.
- Assure le suivi du processus.

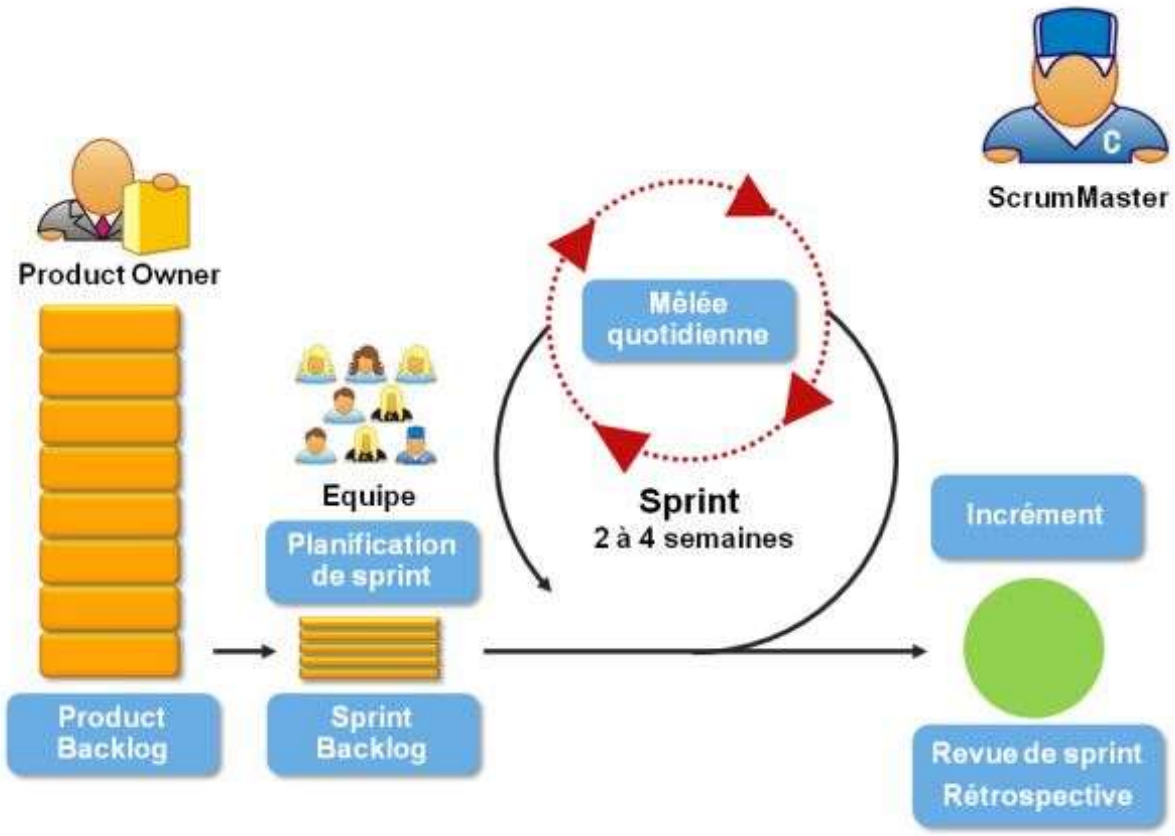
## Equipe de développement

L'équipe de développement :

- Réalise les fonctionnalités du produit.
- Présente au Product Owner les résultats de son travail sous forme de démonstrations.

- Maintient à jour les spécifications détaillées du produit.
- Package et livre le produit.

# Processus




*Zoom sur le processus Scrum (sources des icônes des personnages : Mike Cohn)*

Le processus itératif et incrémental du projet structure le développement du produit en cycles de travail appelés « Sprints ». La durée de ces sprints est fixée à 2 semaines dans le cadre de ce projet. L'objectif générique associé à chaque sprint consiste à transformer les exigences constituant le périmètre de ce dernier en fonctionnalités utilisables.

Juste avant de démarrer un sprint, l'équipe de développement sélectionne les éléments de la liste ordonnancée des exigences (Product Backlog) qu'elle pense pouvoir réaliser dans le délai associé au sprint. Au cours de ce dernier, le Product Owner et l'équipe de développement collaborent étroitement pour atteindre les objectifs fixés. Chaque jour, l'équipe de développement se coordonne et mesure son avancement. A la fin du Sprint, elle présente les résultats de son travail au Product Owner et aux utilisateurs finaux sous la forme d'une démonstration des nouvelles fonctionnalités réalisées. Les feedbacks sont recueillis.

Juste après le Sprint, l'équipe de développement se réunit afin de tirer les leçons du sprint écoulé et étudie les moyens de s'améliorer. S'enchaine ensuite la planification du sprint suivant.

## Préparation

Avant le démarrage d'un Sprint, le Product Owner doit s'assurer que le Product Backlog (Liste des exigences attendues) est correctement ordonnancé en fonction de la valeur métier et du coût de chaque exigence. Pour les exigences dépourvues de coût, il peut solliciter l'équipe de développement afin de procéder à des séances d'estimation. Il participe à ces séances – assisté par son équipe – en répondant aux questions de l'équipe de développement. Il doit également s'assurer que le besoin associé aux exigences prioritaires du Product Backlog (candidates au périmètre du prochain Sprint) est suffisamment mûr (décisions métiers structurantes prises, besoin formalisé sous forme de User Stories,…).

## Planification de Sprint

Le Product Owner et l'équipe de développement se réunissent afin de partager ou repartager ensemble la vision du projet ainsi que les objectifs associés aux éléments du Product Backlog. L'équipe de développement sélectionne ensuite les éléments du Product Backlog en commençant par le haut (exigences prioritaires) en fonction de sa capacité à faire. Enfin l'équipe de développement, découpe chaque exigence en tâches et estime le temps nécessaire à la réalisation de chacune d'entre elles. Ces tâches sont enregistrées dans le Sprint Backlog.

## Sprint

Le sprint englobe les activités d'analyse, de conception, de test et de développement. Au cours de ce dernier l'équipe de développement soulève des questions métiers. L'AMOA répond aux questions (se retourne vers d'autres acteurs tels que la MOA si nécessaire) et complète les User Stories associées. L'AMOA vérifie en cours de sprint la bonne conformité de la fonctionnalité développée, émet au besoin des feedbacks qui permettront d'adapter la fonctionnalité.

L'équipe de développement se réunit chaque jour lors de la mêlée quotidienne afin d'évoquer le travail réalisé la veille, celui de la journée et soulever les obstacles rencontrés. Chaque membre de l'équipe actualise son « Reste A Faire » sur ses tâches en cours de manière à actualiser le graphique d'avancement du sprint. Ce point est limité à 15 minutes, d'autres acteurs du projet peuvent être présents mais seuls les membres de l'équipe de développement ont la parole.

## Revue de Sprint

A la fin du sprint, l'équipe de développement présente au Product Owner accompagné par l'AMOA (et autres acteurs intéressés) les nouvelles fonctionnalités produites sous la forme d'une démonstration. Le Product Owner – avec l'aide de l'AMOA – accepte ou rejette les fonctionnalités présentées. Les feedbacks sont notés. En fonction des résultats présentés, le Product Owner peut demander une livraison du produit pour réaliser des tests de l'ensemble du produit en prévision d'une mise en production. Auquel cas, quelques jours de consolidation du produit en fonction des résultats de ces tests peuvent être nécessaires. Si le Product Owner ne demande pas de livraison, un nouveau sprint est planifié (cf. paragraphe « Planification de Sprint »).

## Rétrospective de Sprint

Après la revue du Sprint, l'équipe de développement ainsi que le Product Owner se réunissent afin d'identifier les adaptations susceptibles d'augmenter sa productivité. Les choses qui fonctionnent, celles qui ne fonctionnent pas ainsi que les améliorations à apporter, sont identifiées dans cette réunion. Elle s'inscrit dans une démarche d'amélioration continue. Les idées de chacun sont mises à profit.

# Réunions

NB : les comités de pilotage ne sont pas décrits dans cette organisation. Ils diffèrent généralement assez peu d'une approche traditionnelle (exemple : comité de suivi hebdo et comité de pilotage mensuel). Il peut cependant être intéressant de se caler sur le rythme des sprints, voire de mutualiser les revues de sprint avec un comité de pilotage par exemple.

## Planification de sprint

- Objectif(s) : définir le périmètre et les objectifs du sprint puis découpage en tâches de développement.
- Responsable : Product Owner.
- Participants : Product Owner, ScrumMaster, équipe de développement.
- Fréquence : Avant le sprint.
- Durée maximale : 4 heures.
- Document(s) en entrée : Product Backlog priorisé.
- Document(s) en sortie : Sprint Backlog.

## Mêlée quotidienne

- Objectif(s) : coordination de l'équipe de développement, identification des obstacles qu'elle rencontre, mesure de l'avancement du Sprint.
- Responsable : ScrumMaster.
- Participants : équipe de développement, ScrumMaster, Product Owner/AMOA (optionnel).
- Fréquence : quotidienne.
- Durée maximale : 15 minutes.
- Document(s) en entrée : Sprint Backlog.
- Document(s) en sortie : Sprint Backlog et graphique d'avancement mis à jour.

## Revue de Sprint

- Objectif(s) : présentation des fonctionnalités produites au cours du Sprint au Product Owner et utilisateurs finaux, réception des feedbacks.
- Responsable : Equipe de développement.
- Participants : équipe de développement, Product Owner, ScrumMaster, utilisateurs finaux, invités.
- Fréquence : A la fin du Sprint.
- Durée maximale : 2 heures.
- Document(s) en sortie : Liste des feedbacks.

## Rétrospective de Sprint

- Objectif(s) : améliorer la productivité (vélocité) de l'équipe de développement.
- Responsable : ScrumMaster.
- Participants : équipe de développement, ScrumMaster, Product Owner/AMOA (optionnel).
- Fréquence : Après la revue de Sprint.
- Durée maximale : 1h30.
- Document(s) en sortie : compte rendu de réunion (bilan de Sprint + plan d'actions).

# Documents

## Product Backlog

Le Product Backlog centralise la liste des exigences attendues (fonctionnalités, exigences non fonctionnelles, défauts à corriger). Le Product Owner s'approprie ce Product Backlog et priorise ses éléments en fonction de la valeur métier et du coût estimé de chacun. L'unité de coût est le « point ». Le choix d'une telle unité permet d'éviter la confusion entre délais et coût, souligne le

caractère « estimatif » (imprécis) et facilite la planification de release et de sprint. Ces estimations sont réalisées dans un premier temps à partir d'un besoin peu détaillé, il peut être révisé au cours de la planification d'itération en cas d'écart par rapport aux hypothèses établies lors de la première estimation.

## Sprint Backlog

Le Sprint Backlog comporte la liste des tâches du Sprint (son périmètre donc) ainsi que la charge de travail associée à ces dernières. Chaque jour, le Reste A Faire de chaque tâche est actualisé par l'équipe de développement afin de tracer le graphique d'avancement de Sprint.

## Tableau des tâches

Ce tableau comporte la liste des tâches du Sprint Backlog, il permet à l'équipe de développement de se coordonner. Il comporte généralement 4 colonnes : « A faire », « En cours », « A vérifier », « Terminé ». Il est mural et simple à interpréter, permettant à n'importe quel acteur du projet de connaître en un coup d'œil l'avancement « temps réel » du Sprint.

## Graphique d'avancement de Sprint

Tout au long du Sprint, n'importe quel acteur du projet peut consulter la progression de l'équipe de développement grâce au graphique d'avancement actualisé quotidiennement. Ce graphique indique l'évolution du Reste A Faire (généralement exprimé en heures) en fonction du temps.

# Méthodes de travail

## Introduction

Partant des constats suivants :

- Les besoins ne sont pas complètement connus tant qu'un projet n'a pas débuté
- Les utilisateurs savent ce qu'ils veulent uniquement après avoir vu une première version du logiciel (Principe d'incertitude du besoin de Humphrey).
- Les besoins changent souvent durant le processus de développement du logiciel (Principe d'incertitude de Hadar Ziv – 1996)
- Spécifier intégralement un système interactif est impossible (Lemme de Peter Wegner – 1995)

- L'approche du projet, concernant les spécifications, consiste à lisser le travail de rédaction des spécifications tout au long du projet et à leur accorder une valeur contractuelle faible. Privilégiant plutôt la réalisation au plus tôt d'un logiciel qui fonctionne à la fin de chaque sprint grâce à une collaboration client/fournisseur (métier/technique, MOA/MOE) étroite.

## Liste des besoins

Le Product Owner établit et maintient la liste des exigences attendues (fonctionnalités, exigences non fonctionnelles ou défauts à corriger). A ce stade, le niveau de précision du besoin est faible. Chaque exigence peut se résumer à 1 à 3 phrases en moyenne. La liste des exigences vient alimenter le Product Backlog.

## Besoins prioritaires

Avant la planification d'un nouveau Sprint, le Product Owner approfondit le besoin associé aux fonctionnalités prioritaires (en tête de liste du Product Backlog). Il faut rédiger pour chaque fonctionnalité attendue une User Story (le verso décrit le besoin et le recto énumère les conditions ou test d'acceptation permettant de vérifier la couverture du besoin). A l'issue de la réunion de planification de Sprint, le périmètre de ce dernier est fixé. Il correspond à la liste d'exigences sélectionnées par l'équipe de développement matérialisées par un ensemble de User Stories.

## Conception/Développement

Une fois le Sprint démarré, le développeur s'approprie la User Story qui servira de support de communication entre lui et le rédacteur de la User Story (Product Owner, AMOA ou utilisateur). Au fil des questions soulevées et des réponses apportées, la User Story se complètera jusqu'à la fin des développements associés.

## Mise à jour des spécifications

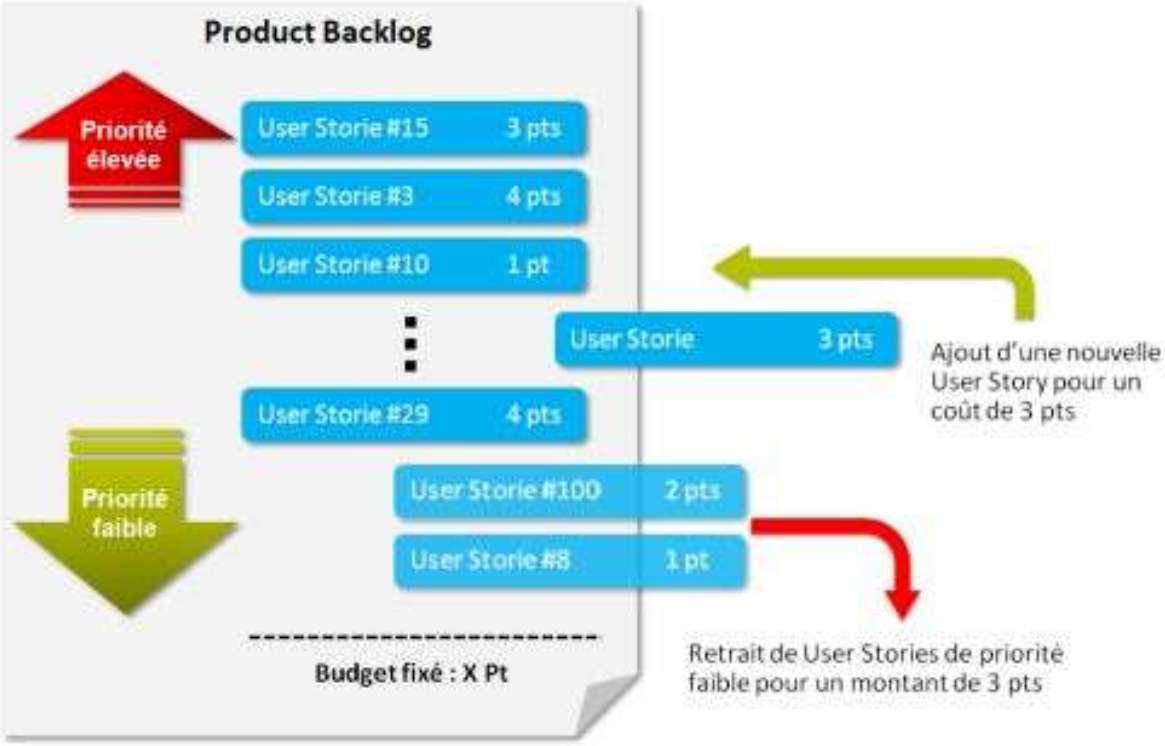
A l'issue du Sprint, l'équipe de développement met à jour la documentation du produit.

## Gestion des changements

Le besoin peut changer tout au long du projet, y compris tardivement, notamment dans les cas de figure suivants :

- Découverte d'un nouveau besoin correspondant une nouvelle fonctionnalité à réaliser.
- Modification d'une fonctionnalité déjà réalisée.
- Retrait d'un besoin finalement inutile ou infaisable.

## Principe de troc




Pour rappel, le Product Backlog centralise l'ensemble des exigences (ou besoins) attendues du projet. Chaque élément de ce dernier comporte une estimation de coût en points. La somme des estimations permet donc de déterminer le coût du projet (estimation).

Si les contraintes de budget voire de délais du projet sont fortes ; en cas de découverte d'un nouveau besoin ou de modification d'une fonctionnalité déjà réalisée ; un nouvel élément est ajouté au Product Backlog en échange du retrait d'un autre de même coût et de valeur métier plus faire (cf. bas de la liste). Inversement dans le cas d'une disparition d'un besoin finalement inutile, l'élément associé est retiré du Product Backlog libérant ainsi une provision pour de nouveaux besoins ou futures modifications.

## Changement pendant un Sprint

Le périmètre d'un sprint fixé à l'issue de la réunion de planification ne doit pas changer. En cas de force majeure, le Product Owner peut annuler le Sprint en cours, ce qui peut engendrer des conséquences coûteuses. Les développements en cours sont stoppés, l'équipe de développement est coupée dans son élan. Un nouveau sprint doit alors être planifié.

# Gestion des défauts

## Processus de gestion des défauts

Un défaut non corrigé pendant le Sprint est enregistré dans le « Bug Tracker » par le testeur. L'AMOA et l'équipe de développement qualifient ensemble les défauts constatés afin de savoir si une correction est nécessaire et possible en fonction des informations du testeur.
Les défauts confirmés sont ensuite priorisés par le Product Owner assisté par l'AMOA. Le Product Owner identifie la liste des défauts à corriger dans le prochain sprint. Il communiquera cette liste à l'équipe de développement lors de la réunion de planification de sprint au même titre que les exigences prioritaires à réaliser. Les défauts critiques (bloquant le travail de test AMOA ou présentant un risque majeur dans l'utilisation du produit prochainement mis en production) peuvent être ajoutés au besoin au périmètre du sprint en cours.

# Concepts

## Planning

Le projet est piloté par la valeur et les délais. Chaque année compte 4 releases de 3 mois afin de créer un rythme régulier, favoriser la naissance des automatismes projet, réduire l'effort de planification du projet et de coordination des acteurs internes et partenaires du projet. Ce principe de release de même durée permet également de construire des indicateurs plus facilement comparables afin d'améliorer régulièrement le processus de développement et l'organisation.
Les releases sont donc des trains partant et arrivant à heure fixe sans décalage possible. Le périmètre est donc la seule variable d'ajustement, afin de sécuriser la qualité, le budget et les délais.

## Planning Poker

Le Planning Poker est une technique d'estimation de coût d'exigences. Cette technique se pratique en équipe et permet de procéder à des estimations rapides et aussi précises que possible selon le niveau de précision du besoin disponible.
Au cours des séances d'estimation, le Product Owner (assisté de l'AMOA au besoin) soumet une à une à l'équipe de développement les exigences dont il souhaite connaître l'estimation de coût. Il répond aux questions de l'équipe de développement qui en cas d'absence de réponse (besoin encore flou) établit des hypothèses. Les estimations de coût (dont l'unité est le point) ainsi obtenues viennent alimenter le Product Backlog et aideront le Product Owner à prioriser ses exigences. La technique de Planning Poker peut également être utilisée pour estimer en équipe

(Product Owner, AMOA, utilisateurs, marketing,…) la valeur métier en points des exigences du Product backlog.

## User Story

La User Story permettant de synthétiser le besoin lié à une fonctionnalité en quelques phrases courtes et simples utilisant exclusivement le vocabulaire métier (absence de connotation technique ou de description d'IHM). D'autre part, une User Story énumère les tests fonctionnels connus à date permettant de vérifier la couverture du besoin de la fonctionnalité associée.

Les avantages sont nombreux :

- Une User Story est compréhensible rapidement aussi bien par un développeur que par un utilisateur,
- Elle favorise les interactions entre développeur et rédacteur, favorisant ainsi une bonne compréhension du besoin (communication face à face),
- Les tests énumérés renforcent la qualité des développements.

## Intégration continue

L'intégration continue a pour rôle de détecter au plus tôt les défauts d'intégrité du code (plus un défaut est corrigé tôt moins il coûte cher). Elle est matérialisée par un outillage logiciel récupérant le dernier code disponible sur le gestionnaire de version, compilant ce dernier et exécutant les tests unitaires existants. Il réalise cette tâche automatiquement à chaque modification du code soumise au gestionnaire de version ou plusieurs fois par jour. En cas de défaut détecté, une alerte est déclenchée (email, signal sonore ou visuel …). Il permet donc de renforcer la qualité des développements et de favoriser le travail indispensable de refactoring régulier du code.

## Développements pilotés par les tests

Le développement piloté par les tests consiste à sensibiliser le développeur sur les tests en amont de ses développements. Concrètement, la User Story associée à une fonctionnalité à développer liste les tests qui permettrons de confirmer que cette fonctionnalité couvre effectivement le besoin. Cette approche limite les erreurs d'interprétation du besoin du développeur.

## Programmation en binôme

La programmation en binôme consiste à rassembler deux développeurs sur un même poste de travail pour accomplir une même tâche de développement. Cette approche apporte plusieurs avantages :

- Permet de réaliser efficacement des développements pointus,

- Meilleure qualité du code (peu de défauts, uniformité, maintenabilité),
- Rapidité de développement,
- Rapidité de montée en compétences (cf. nouveaux développeurs, en particulier junior).