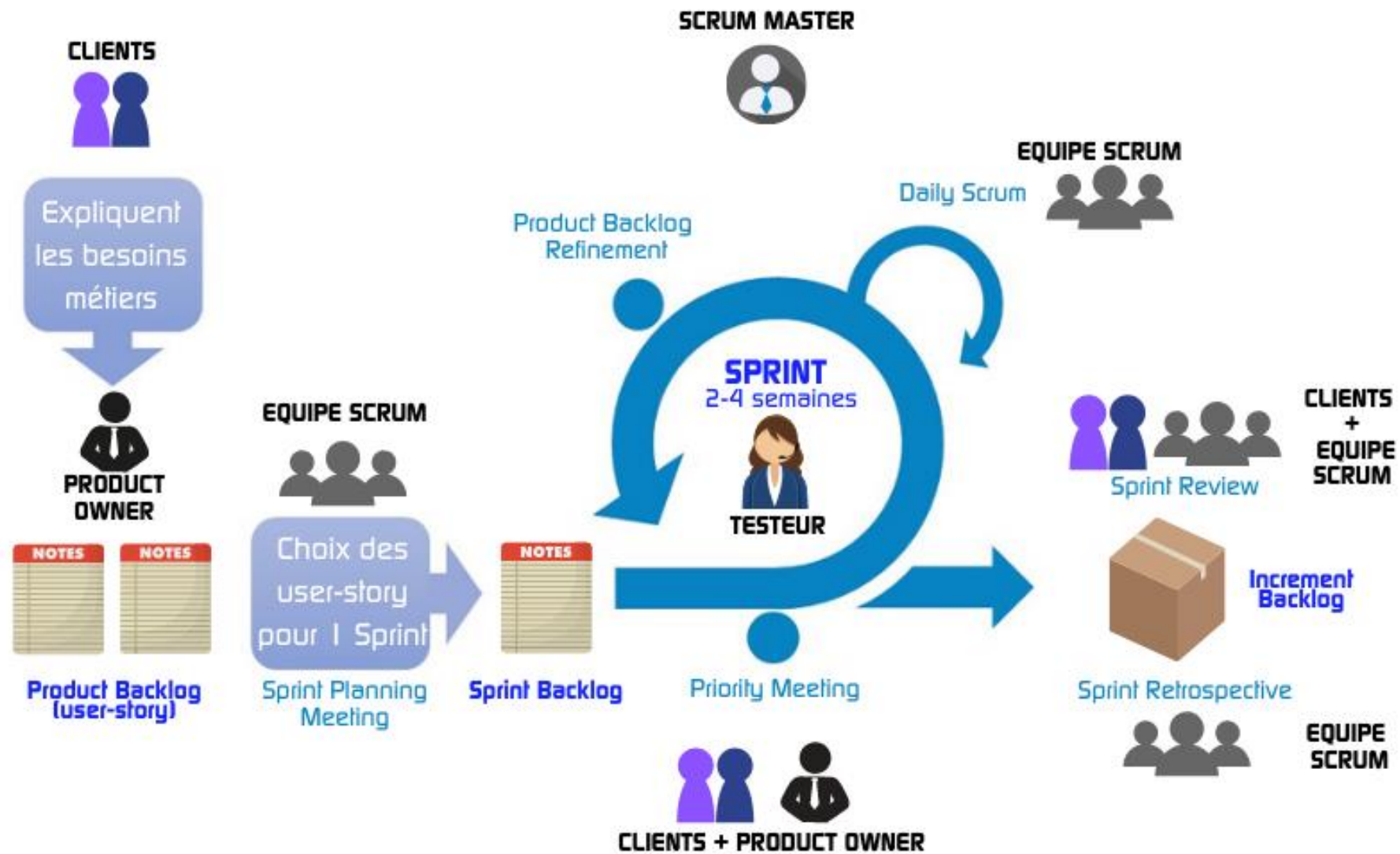
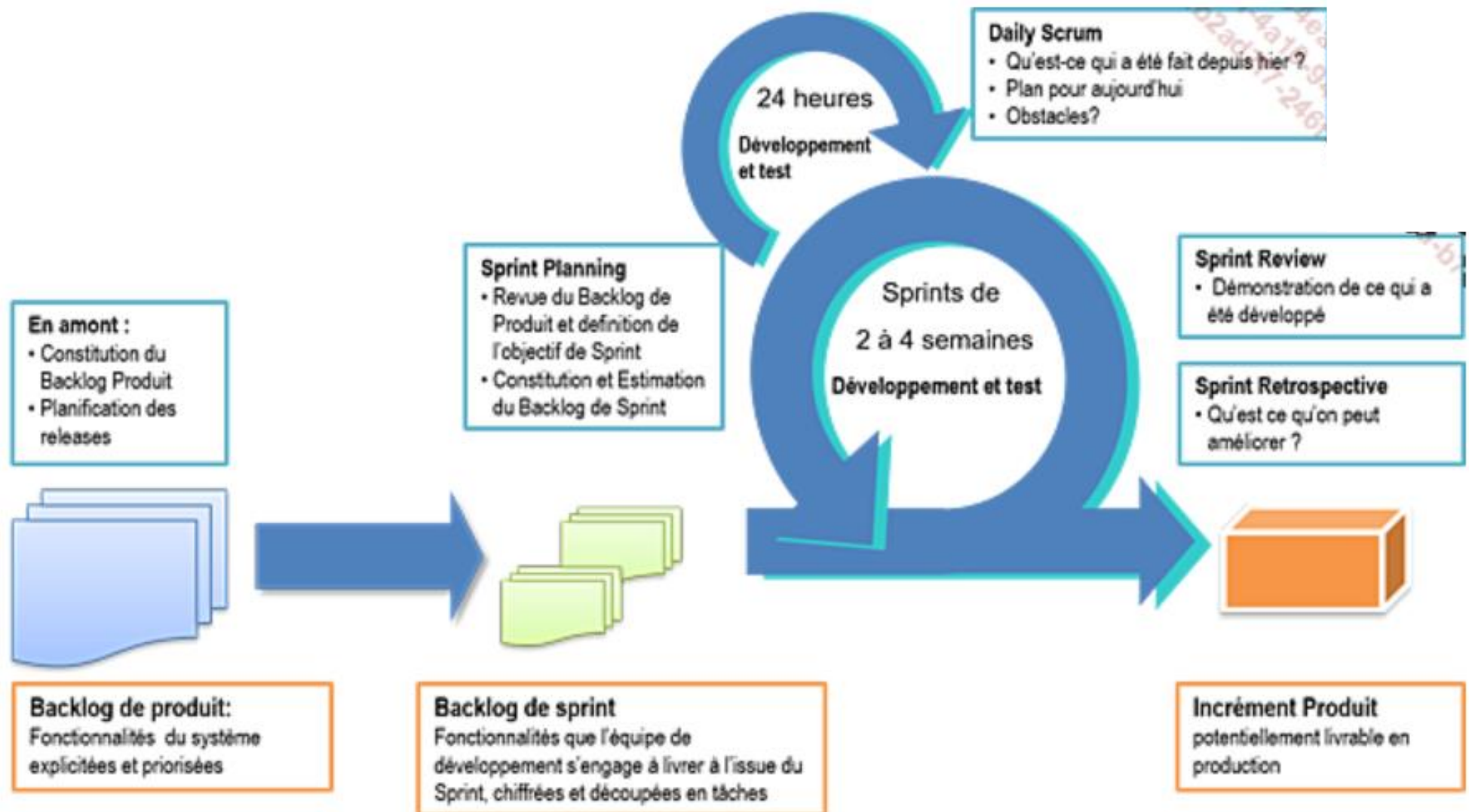


Scrum





Cycle de vie de Scrum

Le cycle de vie est le suivant :

1. Le **Product Owner** rédige les **User Stories** et les place dans le **Product Backlog**.
2. Le **Product Owner** priorise ensuite ces **User Stories** et ordonne le **Product Backlog** en conséquence.
3. L'équipe Scrum se réunit lors de la réunion de planification de Sprint afin d'établir la liste **des User Stories** qui seront traitées pendant le **Sprint**. Celles-ci constituent le **Sprint Backlog** et sont ensuite découpées en tâches par l'équipe de développement.
4. Le **Sprint** peut alors commencer pour une itération de 2, 3 ou 4 semaines.
5. L'équipe se réunit quotidiennement pour réaliser le **daily scrum** (la Mêlée quotidienne).
6. À l'issue du **Sprint**, nous possédons un produit potentiellement livrable qui fait l'objet d'une démonstration lors du **sprint review** (la revue de Sprint).
7. Le cycle se termine enfin par le **sprint retrospective** (la rétrospective de Sprint).

Et ensuite, il n'y a plus qu'à répéter tout cela...

Backlog Produit

Que contient-il ?

Le **Backlog Produit**, aussi appelé **Backlog**, contient l'expression de besoins du **Product Owner** traduite sous forme de **User Stories** (nous verrons bientôt ce dont il s'agit). Il est ordonné selon des critères définis par le PO (nous verrons dans le chapitre dédié à la gestion du **Product Backlog** quelles peuvent être les approches rationnelles de priorisation). L'impact de cela est qu'on traitera les **Stories** dans l'ordre ainsi défini.

Vous l'aurez compris, le Product Backlog est un prérequis à tout projet Scrum.

Backlog de Sprint

Le **Backlog de Sprint** est la liste des **User Stories** issues du **Backlog Produit** devant être traitée durant le **Sprint**.

Chaque User Story y est découpée en tâches devant être réalisées par l'équipe de développement (suffisamment finement pour qu'on puisse facilement suivre la progression de l'équipe lors des mêlées quotidiennes).

Afin de donner un aspect visuel au Sprint Backlog, ce dernier est habituellement représenté sur un tableau (souvent appelé Scrum Board) sur lequel sont ajoutées des cartes (de type « post-it ») correspondant aux User Stories et tâches associées, avec des colonnes matérialisant les différents états des tâches (classiquement « À faire », « En cours », « Terminée »).

| Stories | Tâches | | |
|---------|--|---|---|
| | A faire | En cours | Terminé |
| Story 1 |  |  |  |
| Story 2 |  |  |  |
| Story 3 |  |  | |

Gestion de projet SCRUM

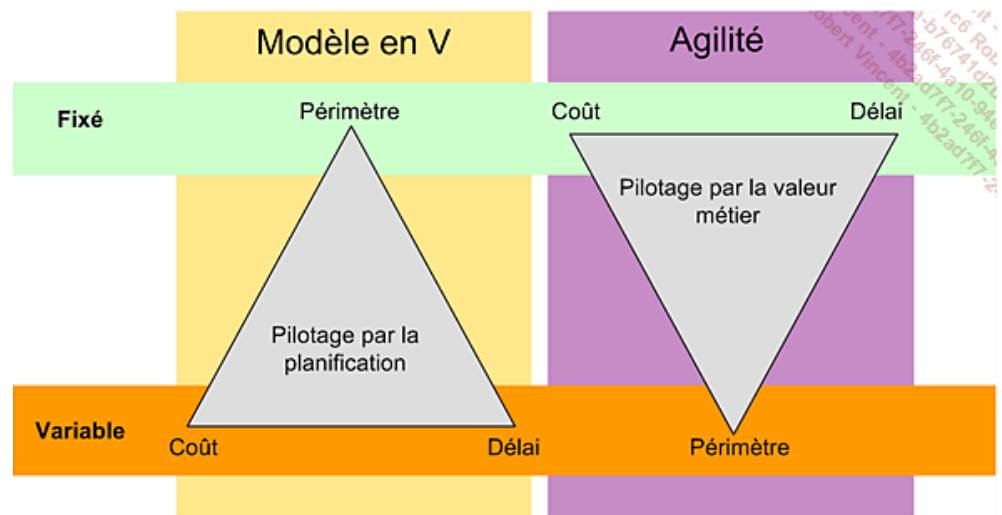
Dans une gestion de projet classique, seul le périmètre est fixé, les coûts et les délais étant quant à eux variables. En effet, si une entreprise souhaite développer une application de gestion de la relation client, le périmètre du projet est fixé mais les inconnues sont le temps de réalisation et les coûts. Ces inconnues influent sur le coût global du projet car rien ne nous indique que les coûts et délais seront respectés (souvenez-vous de l'effet tunnel).

En Scrum, c'est tout à fait l'opposé. Les coûts et délais seront respectés mais le périmètre peut varier. Il s'agit là d'une notion parfois difficile à faire comprendre au management (ainsi qu'au client) et souvent propice à la réflexion suivante :

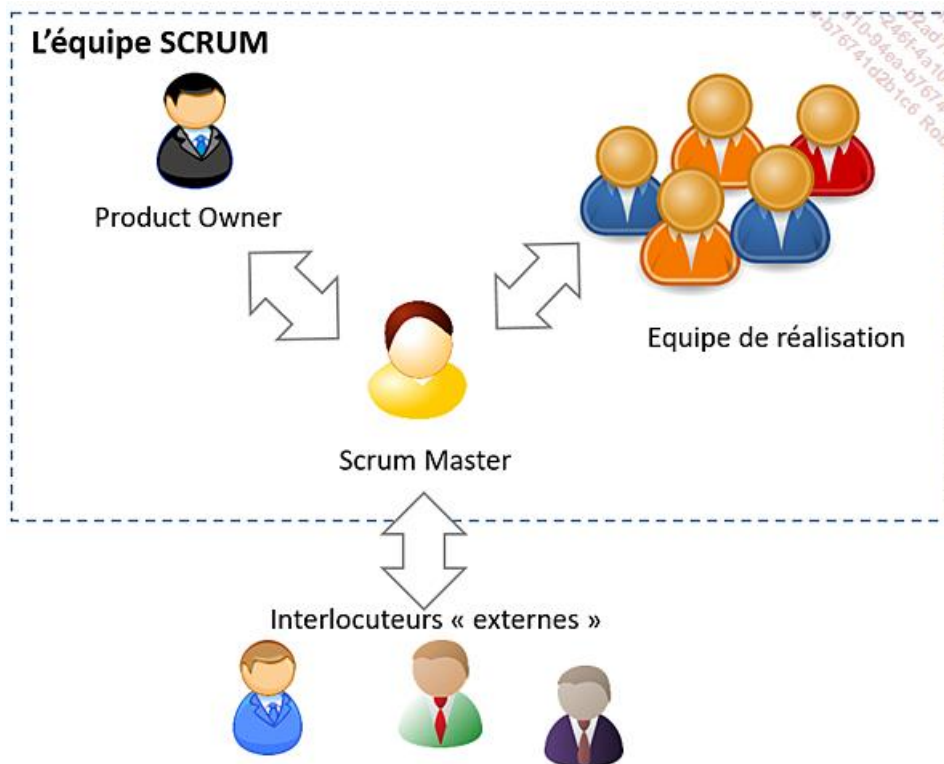
« Il se peut donc que je n'aie pas toutes les fonctionnalités attendues ? Comment cela est-il possible ? L'application ne sera pas utilisable !! »

À ces questions, il est important de répondre que les fonctionnalités à forte valeur métier seront développées en premier et que, certes, celles ayant une faible valeur seront peut-être écartées. C'est donc au Product Owner de s'assurer que le Backlog Produit soit correctement priorisé. Évidemment (mais cela va mieux en le disant), la qualité est non négociable.

Les coûts et délais seront quant à eux respectés du fait que la durée d'un Sprint est toujours constante et que le nombre de personnes travaillant durant un Sprint est connu.



L'équipe SCRUM



Scrum Master

L'une des tâches de base du Scrum Master est l'animation (ou plutôt la « facilitation », qui consiste à s'assurer du bon déroulement) des réunions faisant partie du cérémonial Scrum :

- la planification de Sprint,
- le Scrum Meeting ou Daily Meeting,
- la revue de Sprint,
- la rétrospective de Sprint.

Product owner

La première des responsabilités incombant au Product Owner est de créer la vision du produit.

a. Créer la vision du produit

Par le biais de différentes méthodes et ateliers de travail (dont nous parlerons en détail dans le chapitre suivant), le Product Owner définit la vision du produit répondant à un ou plusieurs problèmes, pilotée par des objectifs précis. Bien entendu, cette vision Produit se doit d'être en phase avec celle des utilisateurs ou clients, dont le PO doit être la voix unique vis-à-vis de l'équipe Scrum.

Insistons bien en outre sur le fait que le PO est une personne et non un comité ou une équipe : c'est au final le PO nommément désigné qui fait des choix et prend les décisions.

b. Gérer le Product Backlog

Le Product Owner est le seul responsable du Product Backlog. Par conséquent, il porte la responsabilité de l'écriture des éléments de ce Backlog et de leur priorisation.

c. Maximiser la valeur du produit et du travail de l'équipe

Définir et prioriser le Product Backlog n'est pas suffisant : il ne faut pas oublier que le PO doit pouvoir expliquer les besoins à l'équipe de réalisation et que la prise en compte des retours et questions de celle-ci est un élément essentiel dans la définition d'un Product Backlog pertinent.

d. Définir le plan de Release

Ce n'est pas au Scrum Master et à l'équipe de réalisation de décider des dates de livraison et du périmètre des Releases (une Release, c'est une livraison du projet qui regroupe le travail de quelques Sprints). C'est bien le Product Owner qui les définit, en étroite collaboration avec le Scrum Master et l'équipe de développement (en tenant compte par exemple des retours de celle-ci sur sa capacité à faire ou sur telle ou telle contrainte technique).

e. Implication dans le processus Scrum

Le Product Owner doit s'impliquer dans l'ensemble du processus Scrum en participant à l'ensemble des cérémoniaux auxquels sa présence est requise, mais également en collaborant étroitement avec l'équipe.

Il fait partie intégrante de l'équipe, ne l'oublions pas, et partage les mêmes objectifs que tous ses autres membres : on doit éviter de tomber, dans sa relation avec l'équipe de réalisation, dans un schéma du type client/fournisseur.

f. Accepter ou non le résultat d'un Sprint

Lors de la revue de Sprint, le Product Owner a la possibilité d'accepter ou non la livraison effectuée par l'équipe de développement. Ce refus ou cette acceptation doit se baser sur des indicateurs qui lui sont propres mais partagés avec tous les membres de l'équipe Scrum. L'un de ces critères est évidemment le résultat des tests d'acceptation sur les User Stories du Sprint.

g. Ses pouvoirs et limites

Le Product Owner a pour pouvoir :

- de construire le contenu du Backlog Produit et ainsi de définir le contenu des Sprints et Releases,
- de définir les priorités et fonctionnalités candidates à chaque Sprint.

Une règle importante régit le dernier point : en Scrum, lorsqu'un Sprint est commencé, la modification des fonctionnalités du Sprint (par exemple, remplacer une Story par une autre non prévue) n'est plus possible. C'est la condition indispensable au respect de l'engagement pris par l'équipe sur le contenu du Sprint.

Il est assez commun que les membres de l'équipe de réalisation soient sollicités par des interlocuteurs externes à l'équipe projet pour réaliser des tâches non prévues dans le Sprint : il faut absolument résister à celles-ci, même si la demande vient du chef de service ou du DSI et s'appuyer pour cela sur le Scrum Master, qui expliquera à l'interlocuteur comment interagir avec l'équipe dans le cadre de Scrum et le mettre en relation avec le PO.

La User Story

Pour guider la rédaction des User Stories, un principe très simple à retenir a été proposé par Ron Jeffries. Il s'agit de la règle des 3C :

- La Story est écrite sur une carte de taille assez réduite. Ces fiches peuvent être annotées (estimation, etc.).
- Les détails de la Story seront exprimés lors de conversations avec le Product Owner.
- Des tests de validation sont décrits avec la Story (ils serviront à valider qu'elle a été réalisée correctement).



Lorsqu'on entre dans le processus de rédaction, on peut rapidement être perdu... Dans ce cas, un autre principe très utile vous guidera : il s'agit du **principe INVEST**.

Il indique qu'une bonne User Story doit être :

- Indépendante des autres histoires d'utilisateur (dans la mesure du possible).
- Négociable : elle doit pouvoir être discutée avec l'équipe chargée de la réalisation du produit, notamment lors de l'estimation.
- Source de Valeur : elle doit être porteuse d'une valeur pour le client ou l'utilisateur.
 - Une User Story ne décrit pas des finalités techniques !
- Estimable : elle peut être estimée par l'équipe de réalisation avec un risque d'erreur faible.
 - Elle doit, à cette fin, être rédigée de manière claire et compréhensible.
- D'une taille Suffisamment petite afin de faciliter son estimation et afin d'assurer qu'elle puisse être conçue, développée et testée au sein d'un Sprint.
 - Si la Story est trop grosse, c'est sans doute plutôt une Epic, qui devra être découpée plus finement préalablement à la réalisation.
- Testable : une User Story doit être accompagnée des critères de validation permettant sa validation.

Identifier les fonctionnalités clés avec la Product Box

Cet atelier très ludique, qui n'est pas spécifique à Scrum, permet de faire éclore et partager la vision produit avec toutes les parties prenantes d'un projet. Son résultat n'est pas de produire un Product Backlog détaillé, mais permet d'identifier les grandes orientations/fonctionnalités du projet ou produit et de construire une vision commune avec l'ensemble de l'équipe concernée (PO, développeurs, testeurs, UX/UI designers, responsable métier, marketing...).

Il consiste à matérialiser les différents aspects de la vision sous forme d'une boîte qui représente le produit, comme on le trouverait dans les rayons d'un supermarché pour un produit physique (d'où le nom de l'atelier...). Ceci dit, dans une version plus « techno », on peut imaginer cela comme les éléments qu'on mettrait sur une page web de présentation du produit à ses clients potentiels.

La boîte doit permettre à celui qui la découvre de comprendre très rapidement :

- Quel est le produit ?
- Pour notre projet : un logo et surtout un slogan/une phrase très synthétique qui permet de le comprendre en un coup d'œil.
- Quelles sont les caractéristiques principales (et avantages) de ce produit ?
- Le cas échéant, quelle est la composition du produit et quelles sont ses conditions d'utilisation ? Dans le cas d'un projet, cela serait typiquement la composition de l'équipe (au moins les compétences nécessaires), les prérequis nécessaires au démarrage du projet.

Déroulement

Une durée totale de 1h00 à 1h30 convient en général. C'est bien d'avoir un « time keeper » pour faire attention à la durée de chaque phase. Après présentation du principe et questions-réponses (10 minutes), on se divise (selon le nombre de participants) en petits groupes de 4 ou 5 personnes maximum.

Étape 1

Rassembler les idées pour fournir :

- Pour la face avant de la boîte : un nom, un logo, un slogan.
- Pour la face arrière de la boîte : les fonctionnalités clés du produit (3 ou 4 au moins, mais pas trop non plus : il ne s'agit pas d'initialiser un Backlog détaillé...).

On peut y ajouter pour les côtés de la boîte tous les éléments complémentaires qui pourraient être utiles au « futur consommateur » :

- conditions/précautions d'utilisation,
- prérequis techniques ou organisationnels,
- composition d'équipe,
- performances attendues,
- ...

Pour cela, prévoir 30 à 40 minutes de discussion/brainstorming. Il est bon de faire des points d'étape avec les groupes (par exemple, toutes les 15 minutes) pour s'assurer que les choses avancent.

Étape 2

Ensuite, chaque groupe réalise sa boîte (prévoir environ 15 minutes). Le résultat final devrait ressembler à ceci (face avant et face arrière) :



Étape 3

10 min pour préparer une présentation rapide (en mode « Elevator pitch ») qui sera faite à l'ensemble des participants.

Chaque groupe doit se choisir un speaker qui dispose de 2 minutes pour convaincre : il doit vendre sa boîte !

On peut ensuite élire la meilleure boîte.

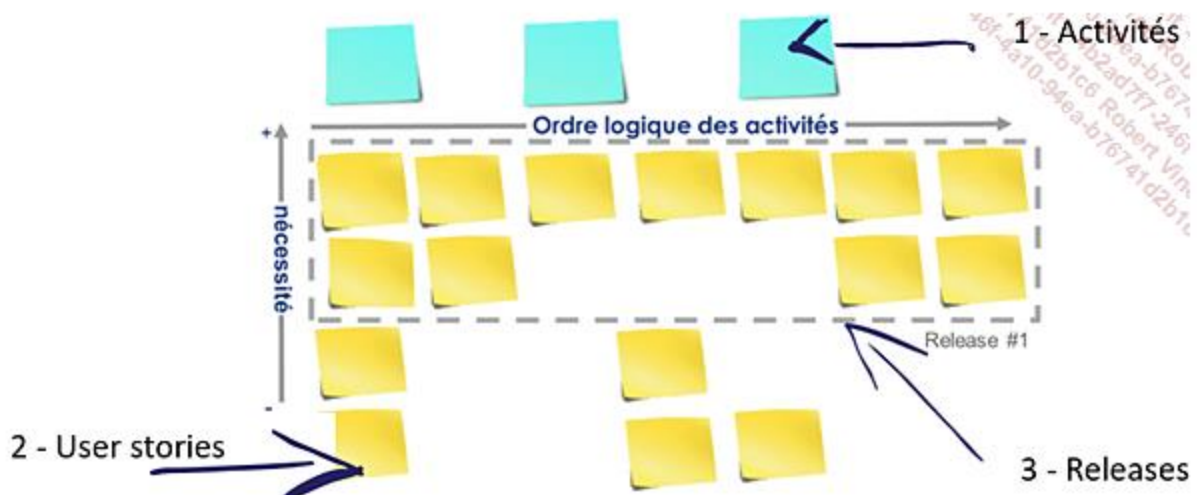
Le story mapping

Le « User Story Mapping », est une technique qui permet de découvrir les fonctionnalités de son système de manière très intuitive. Elle ne fait pas formellement partie de Scrum, mais le complète utilement. Elle consiste à construire une représentation visuelle ordonnée, en s'appuyant sur l'expérience utilisateur : ses constituants de base sont donc les activités et tâches que l'utilisateur accomplit avec le système quand on raconte l'histoire de l'utilisation de celui-ci.

Product Backlog

Son résultat aboutit à un présenté de la manière suivante :

- Activités : ce sont les actions de « haut niveau », ordonnées de gauche à droite selon l'ordre logique (celui qui apparait quand on raconte l'histoire de l'utilisation du système par ses utilisateurs).
- User Stories : les tâches accomplies dans le cadre de chaque activité sont placées en dessous de celle-ci et ordonnées de haut en bas par « nécessité » (les plus indispensables en haut) et de gauche à droite selon l'ordre logique (comme les activités). En général, on a une ligne supérieure composée d' « Epic Stories » qui sont les macrotâches principales.
- Releases : une fois que les Stories ont été ordonnées, on découpe la carte en tranches qui constituent des Releases en partant des Stories de plus grande valeur.



Story Mapping illustré par un exemple

Prenons le cas concret d'un portail de gestion de comptes bancaires en ligne. Nous allons effectuer dans l'ordre :

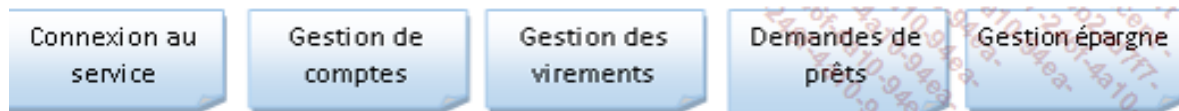
- l'identification des activités,
- l'identification des tâches principales,
- la définition des User Stories,
- la classification par ordre de nécessité.

Ces tâches seront menées à bien via des ateliers menés avec le Product Owner.

Les activités

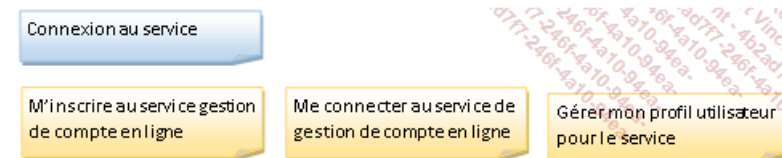
On peut imaginer le scénario typique suivant (vu du client de la banque) : je me connecte, je consulte la situation de mes comptes courants, je gère mes virements, je consulte l'état d'avancement de ma demande de prêt, je m'informe sur les produits financiers d'épargne.

On pourrait dans ce cas matérialiser les activités de la manière suivante :



Les tâches principales

Dans le cadre de ces activités, en tant que client, on accomplit un certain nombre de tâches. Prenons l'exemple de l'activité « Connexion au service ». Dans cette thématique, je vais typiquement : demander mon inscription au service de gestion de compte en ligne, me connecter/déconnecter, gérer mon profil, etc.



Deux remarques importantes :

- Ces tâches principales sont clairement des Epic Stories. Elles décrivent bien une attente de l'utilisateur vis-à-vis du système mais en réfléchissant un peu on s'aperçoit qu'elles couvrent un périmètre qui n'est pas très détaillé et qu'elles peuvent donc être découpées en fonctionnalités plus fines.
- Ici, on a volontairement simplifié la rédaction...

Les Stories

Comme indiqué ci-dessus, les tâches principales de l'utilisateur telles que décrites ici ne sont pas encore assez détaillées pour bien décrire le système.

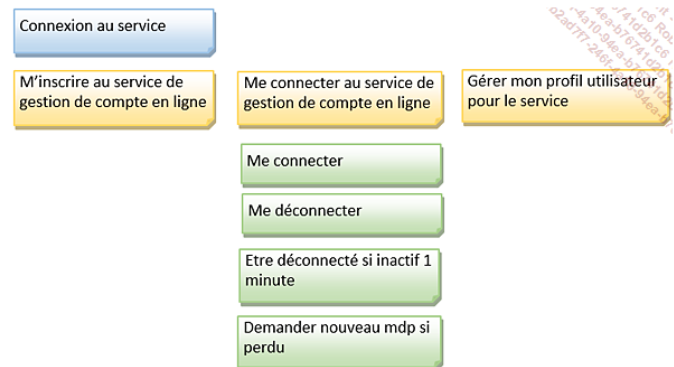
Prenons l'exemple de la macrotâche « Me connecter au service de gestion de compte en ligne ». Si on la découpe finement, on aboutit par exemple aux « User Stories » suivantes :

« En tant que client de la banque inscrit au service de gestion de comptes en ligne, je veux pouvoir me connecter au service en saisissant mon identifiant d'utilisateur et mot de passe afin de gérer mes comptes en ligne »

« En tant que client de la banque inscrit au service de gestion de comptes en ligne, je veux pouvoir être déconnecté automatiquement du service au bout de 1 minute d'inactivité afin de ne pas laisser ma session ouverte »

etc.

Sur notre « carte » des Stories, cela donnerait quelque chose comme ceci :



Priorisation

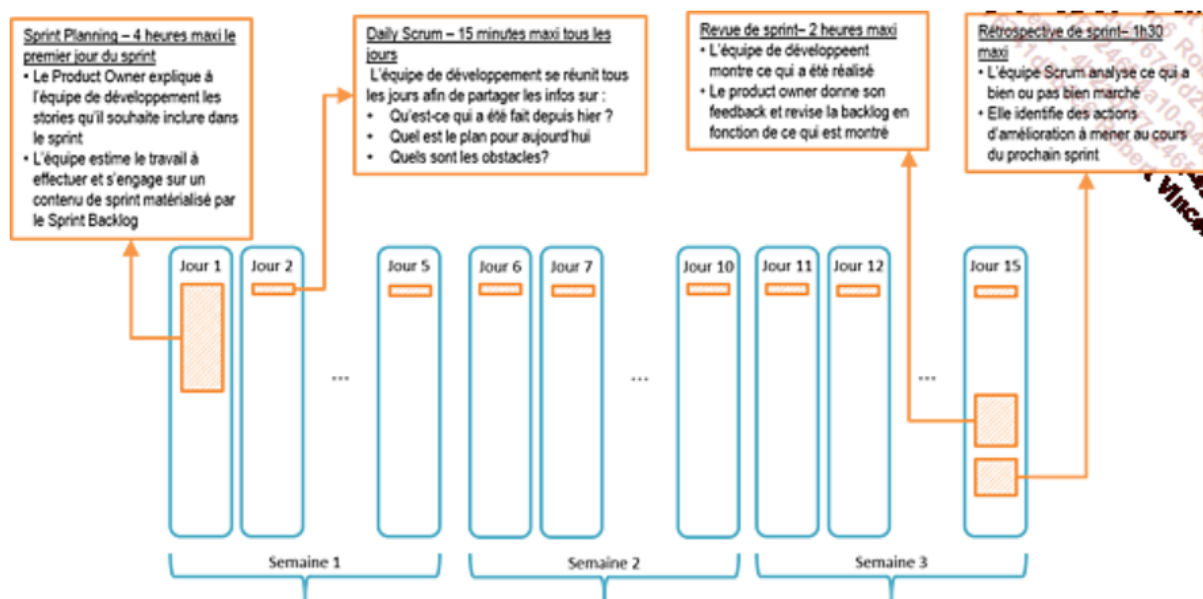
S'il est important d'identifier l'ensemble des fonctionnalités attendues, il l'est tout autant de les ordonner puisque l'approche Agile nous impose de découper le développement en cycles courts à la fin desquels on livre une application potentiellement utilisable par son client final.

Dans le cadre du Story Mapping vous allez, sur tout le périmètre du système, déterminer quel est l'ordre de nécessité des Stories (les plus nécessaires en haut sur la « carte » du système). On n'essaiera pas de déterminer cela avec une finesse trop grande qui serait inutile : l'essentiel est d'aboutir à un certain nombre de « tranches horizontales » qui regroupent un ensemble de fonctionnalités. Chaque tranche représente typiquement une « Release » du système (par exemple, une version qu'on va mettre en production).

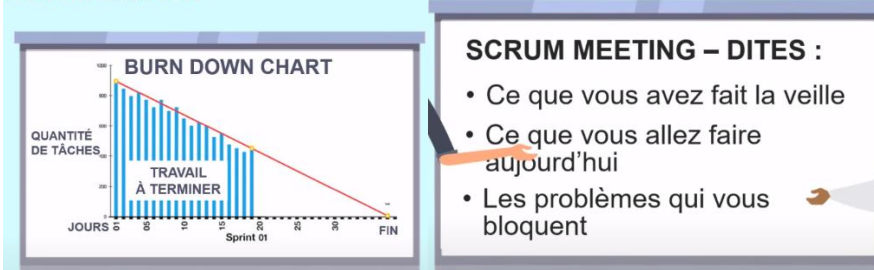
Donc, au final dans notre exemple de service de gestion de comptes en ligne, on pourrait aboutir à quelque chose comme cela (c'est forcément une vue partielle, car à ce stade, on a en principe plusieurs dizaines de Stories) :



Les sprints



SCRUM MASTER



Lors du **Sprint Review**, on présente la version au client et on tient compte des modifications qu'il propose.

L'ensemble des livraisons des Sprints cumulés se nomme le "**Sprint Backlog**"