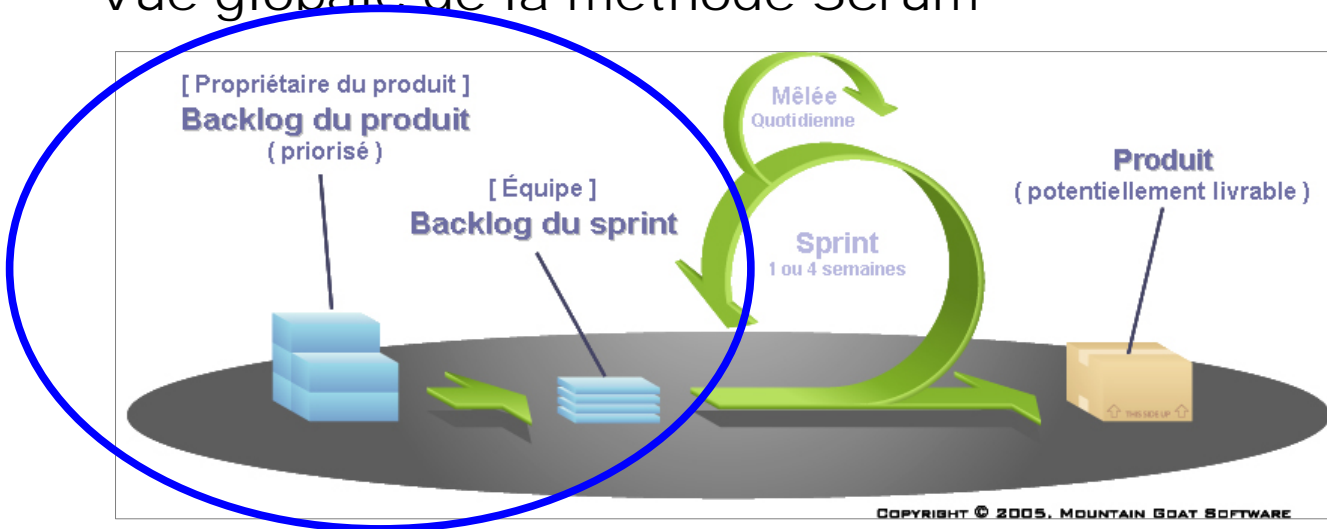


« If you fail to plan, you plan to fail. » (Adage anglais)

« Il n'y a pas de vent favorable pour celui qui ne sait pas où il va. » (Goethe)

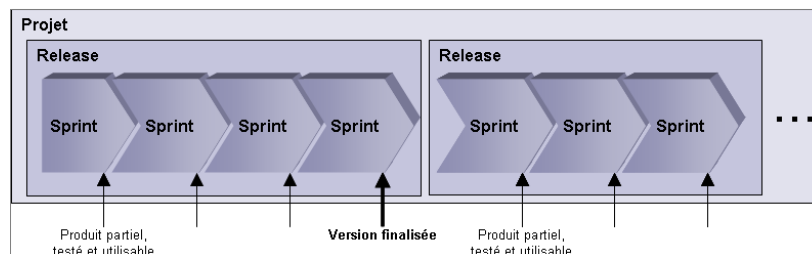
## Vue globale de la méthode Scrum



## Rappel

Scrum est un processus itératif : les itérations sont appelées des **sprints** et durent généralement entre 2 et 4 semaines. Chaque sprint possède un but et on lui associe une liste d'items du « *Backlog de produit* » à réaliser (fonctionnalités/exigences/livrables) . Un **sprint aboutit toujours** sur la **livraison d'un produit partiel, mais fonctionnel**.

Pour améliorer la lisibilité du projet, on regroupe généralement des itérations en « *Releases* ». Bien que ce concept ne fasse pas explicitement partie de Scrum, il est utilisé pour mieux identifier les versions. En effet, comme chaque sprint doit aboutir à la livraison d'un produit partiel, une « *release* » permet de marquer la livraison d'une version aboutie, susceptible d'être mise en exploitation.



La présente note de cours donne plus de précision sur 2 des artefacts de Scrum, soit le « **Backlog de produit** » et le « **Backlog de sprint** » qui concernent principalement les activités de planification d'un projet.

## Cours 3

### Planification d'un projet avec Scrum

---

#### Sprint Zéro

Le **Sprint Zéro** n'est pas un vrai sprint (agile) au sens où il ne se termine pas forcément par la livraison d'un produit qui marche, et que sa durée est variable selon l'envergure du projet (de quelques jours à 3 ou 4 semaines). Ce sprint est pourtant essentiel pour **mettre le projet sur de bons rails** et permettre à l'équipe d'apprendre à travailler ensemble et de mieux définir les paramètres du projet à réaliser.

#### Quelles sont les activités typiques menées en Sprint 0 ?

- Partager une vision claire du produit entre les membres de l'équipe, le « Product Owner » et le « ScrumMaster »;
- Analyser les risques;
- Déterminer le time-box des sprints (durée);
- Déterminer un plan de « *release* » (version);
- Produire un « **Backlog de produit** » estimé et priorisé;
- Préparer l'environnement de développement (y compris quelques docs tels: normes de codage ou de documentation, etc.);
- Définir la posture ergonomique de l'interface et identifier les cibles (Personas);
- Établir les normes graphiques;
- Structurer l'arborescence;
- Classifier les contenus;
- Développer le modèle des données;
- Selon les contextes, travailler l'architecture (exemple MVC);
- Etc.

#### « Backlog du produit »

Nous avons vu la semaine dernière que le « **Backlog du produit** », est la liste des exigences/ fonctionnalités/livrables à réaliser (« *wish list* ») généralement exprimées par des « *User Stories* » (voir note de cours de la semaine 2). Le « Backlog du produit » est le **référentiel** qui regroupe **tout ce qu'il y a à faire, tous les besoins du client ou des utilisateurs**. Il peut être traduit par cahier, liste ou carnet de commandes, qui ne collent pas bien avec l'esprit du terme anglais qui évoque aussi une réserve, une file d'attente, un restant à faire.



Les items du « Backlog du produit » peuvent être identifiés différemment selon la nature et les caractéristiques du projet à réaliser: lecture approfondie (*et entre les lignes...*) du cahier des charges du client (*se rappeler que le client peut de pas identifier clairement des besoins*

## Cours 3

### Planification d'un projet avec Scrum

*implicites...*), un remue-méninges de l'équipe de projet, l'analyse des forces et faiblesses d'un site Web existant (*dans le cas d'une refonte par exemple...*), une analyse de produits concurrentiels (« *l'offre suscite le besoin...* »), entrevues ou « focus group » avec des utilisateurs potentiels, etc.

Le « Backlog du produit » sert à formaliser les besoins et à les hiérarchiser. À cet effet nous avons également vu la semaine dernière qu'à chaque **item** du « Backlog du produit » sont associées **deux attributs** : une **estimation** en **points arbitraires**, et une **importance**:

- **L'estimation**<sup>1</sup> des items du « Backlog du produit » se fait selon des points arbitraires. Par exemple, l'équipe prend un item représentatif, et lui affecte un nombre de points arbitraire. Cela devient un **référentiel** pour estimer les autres items. Un item qui vaut 2 points représente deux fois plus de travail qu'un item qui en vaut 1. Par exemple, si une fonctionnalité très simple à implanter, comme l'accès courriel, est été évalué à 1 ou 0.5, les autres estimations doivent être définies en conséquence. Toutefois, les fonctionnalités obtenant des valeurs trop grandes selon le « time-box » des sprints doivent être segmentées en sous-fonctionnalités et estimées de nouveau. Il faut garder à l'esprit qu'un item du « Backlog du produit » doit pouvoir être réalisé complètement à l'intérieur d'un sprint (conception, modélisation, traitement des médias, programmation, testage, débogage, optimisation, etc.);
- **L'importance** est une priorité donnée aux items du « Backlog du produit » par le « *Product Owner* » (ou le client) en fonction de leur valeur ajoutée en regard du degré de satisfaction résultant de leur implémentation. Elle permet de définir dans quel **ordre** ils devront être réalisés. Par exemple 10 est une importance faible et 150 une importance plus élevée. Le « *Product Owner* » (ou le client) peut changer cet ordre en cours de projet, et même ajouter, modifier, ou supprimer des items dans le « Backlog du produit ». La hiérarchisation des items est, la plupart du temps, identifiée avec le **modèle de Kano**. Toutefois, il importe de préciser que les membres de l'équipe de projet doivent également donner leur avis sur les priorités en fonction du **risque technique** (complexité de réalisation, problèmes d'intégrité ou de sécurité des données, etc.), de la **complémentarité** ou de la **relation** des **items** du « Backlog du produit ».

**Faut-il implémenter les fonctionnalités les plus risquées, au détriment de la valeur ajoutée pour le client? Faut-il au contraire, focaliser sur la satisfaction du client en reportant la confrontation au risque? Il est important de mettre en balance les deux aspects.**

Risque élevé	À éviter	Prioritaire
Risque faible	Reporté	Secondaire
	Faible valeur ajoutée	Forte valeur ajoutée

<sup>1</sup> L'estimation est la plupart du temps réalisé avec la technique du **Planning Poker** que nous verrons à la fin du cours.

Ainsi, les fonctionnalités à forte valeur ajoutée ( par exemple, celles identifiées comme proportionnelles ou attractives avec le modèle de Kano), seront prioritaires; au sein de ce groupe, celles qui sont les plus risquées seront développées avant celles qui sont le moins risquées. Celles qui ont peu de valeur et sont très risquées seront reportées, voire éliminées.

**L'échelle d'importance doit donc prendre en compte les risques et la satisfaction du client et être établie, généralement, en consensus avec ce dernier.**

- Par ailleurs, rappelons qu'un item du « Backlog du produit » défini comment cet item (ou « *User Story* ») sera démontré à la **démo** de fin d'itération. C'est essentiellement la spécification d'un test simple : « *Fais ci, ensuite fais ça, puis ceci devrait arriver* ».

BACKLOG DE PRODUIT (exemple)					
ID	Nom	Imp	Est	Démo.	Notes
1	Dépôt	30	5	Authentification, ouvrir la page de dépôt, déposer 10€, aller sur la page du solde et vérifier que ça a bien augmenté de 10€.	Nécessite un diagramme de séquences UML. Ne pas se soucier du cryptage pour l'instant.
2	Voir l'historique de ses transactions	10	8	Authentification, cliquez sur « transactions ». Faire un dépôt. Revenir aux transactions, vérifier que le nouveau dépôt apparaît.	Utiliser la pagination pour éviter des requêtes volumineuses . Conception semblable à la page des utilisateurs.

Exemple partiel d'un « backlog de produit ».

Enfin, dans la section « **Notes** » on précise, s'il y a lieu les informations que les membres de l'équipe de projet peuvent avoir à prendre en compte pour réaliser ou estimer l'item du « Backlog du produit ». Cette rubrique peut également préciser des exigences supplémentaires comme des exigences non fonctionnelles, des critères de performance ou des contraintes de conception, par exemple: « *la recherche doit être la plus rapide possible: 95% des requêtes doivent aboutir en 3 sec* », « *Pour garantir la sécurisation et l'intégralité des données, il est impératif que l'envoi des données se fasse de manière cryptée* », etc.

## Cours 3

### Planification d'un projet avec Scrum

---

Chaque item du « **backlog du produit** » est généralement consigné sur une fiche ou carte.

Elément Backlog #55

# Dépôt

Notes

Besoin d'un diagramme de séquence UML.  
Nul besoin de se soucier du chiffrement pour l'instant.

Comment démontrer

S'authentifier, ouvrir la page des dépôts, déposer 10 € et vérifier sur la page du solde qu'il a augmenté de 10 €.

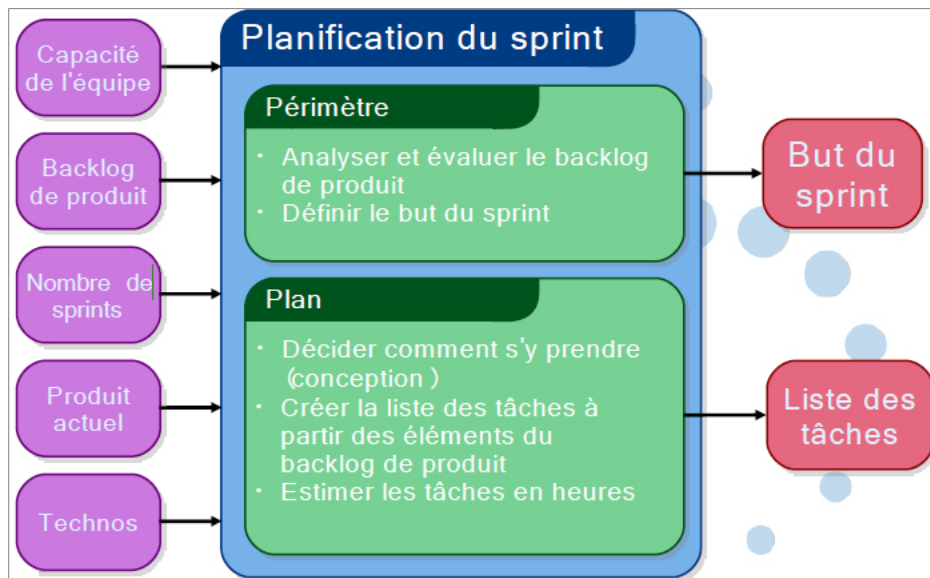
Importance

30

Estimation

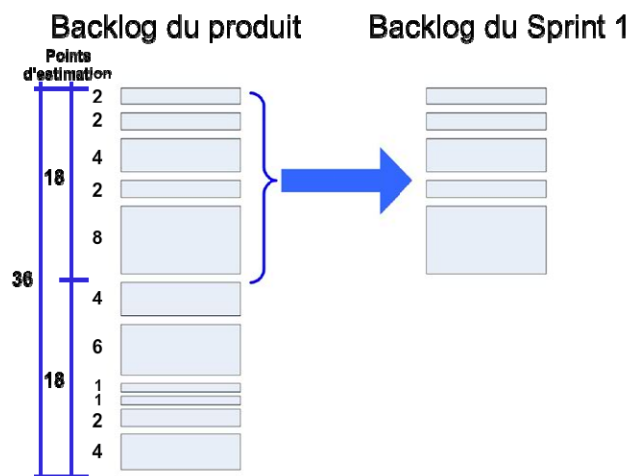
5

#### Le « Sprint planning »



- Le « **Sprint planning** » ou réunion de planification d'un Sprint vise à définir le **but** et le **plan** du Sprint. Normalement tous les membres de l'équipe Scrum sont présents à cette réunion, **qui ne doit pas durer plus de 4 heures**.
  - Le « **Sprint Planning** » consiste d'abord à définir un **but** pour le Sprint, puis à choisir les items du « **Backlog du produit** » qui seront réalisés dans ce **sprint**, soit le « **Backlog de sprint** ». Cette première partie du « **Sprint planning** » représente l'**engagement de l'équipe**. Les items retenus pour chaque sprint prennent en considération l'**estimation globale** des items et le **nombre de sprints** pour réaliser le projet.

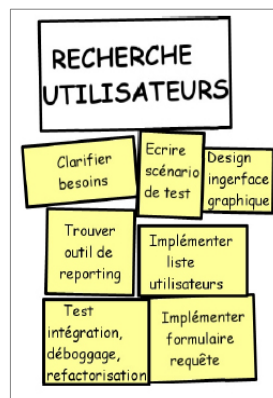
#### Exemple pour un projet de 2 Sprints



## Cours 3

### Planification d'un projet avec Scrum

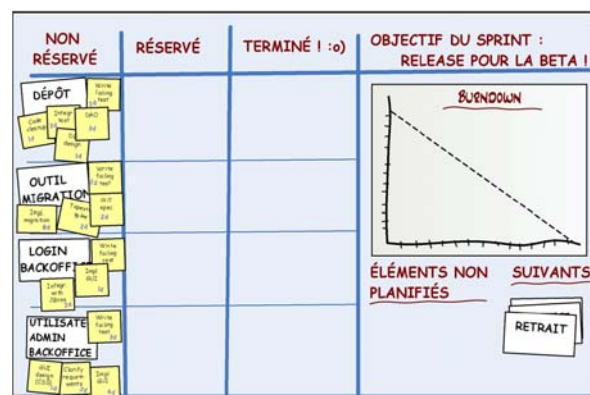
- Dans un second temps, l'équipe décompose chaque item en liste de **tâches élémentaires** (techniques ou non). Une **tâche élémentaire**, selon sa définition, c'est un **travail** qu'on confie pour exécution à **un seul membre de l'équipe sans le subdiviser davantage**. Pour obtenir un plan d'exécution efficace, il est important de ne pas multiplier indûment le nombre de tâches élémentaires. Il est par contre dangereux de définir une tâche élémentaire de trop grande importance et dont la durée est trop longue, car il sera difficile de mesurer l'état de son avancement en phase de réalisation. Avec Scrum, chaque tâche est estimée en heures (toujours avec la technique du *Planning Poker*) et ne doit pas durer plus de 2 jours (16 heures). Auquel cas, il convient de les décomposer en deux ou plusieurs sous-tâches.



Exemple de décomposition d'un item en tâches

#### Le « Backlog du sprint »

- Dès que la planification du **sprint** est complétée (« **Sprint planning** »), le « **ScrumMaster** » doit créer la **version numérique** du « **Backlog de sprint** » avec Excel ou d'autres outils logiciels comme *IceScrum* (logiciel libre). Les items du « Backlog du produit » réalisés pendant le sprint et leurs tâches associées sont normalement **transcrits du haut vers le bas de la liste selon leur ordre d'importance**. Par ailleurs, le « **Backlog de sprint** » est aussi représenté physiquement ou concrètement. Le format jugé le plus efficace, en termes de visibilité et de diffusion, est simplement un **tableau des tâches sur le mur**!

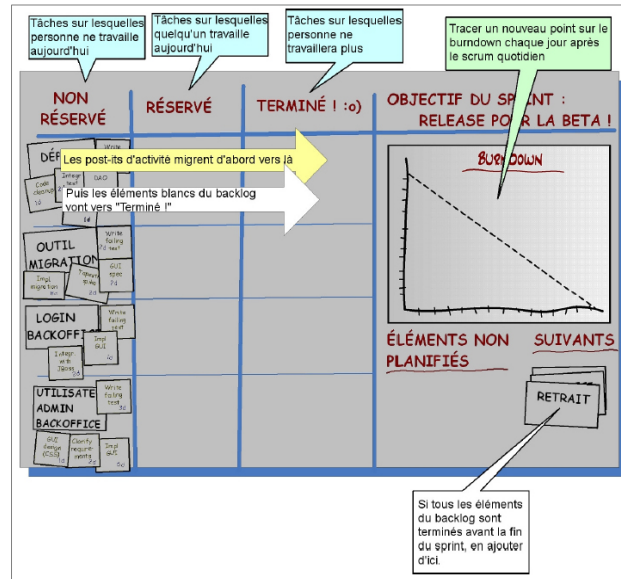




## Cours 3

### Planification d'un projet avec Scrum

- Pendant le déroulement d'un sprint, **chaque équipier s'affecte des tâches du « Backlog de sprint »** et les réalise. Il met à jour régulièrement dans le « **Backlog du sprint** » le reste à faire de chaque tâche. **Il est important de noter que les tâches ne sont pas réparties initialement entre tous les équipiers, elles sont prises au fur et à mesure que les précédentes sont terminées.** Idéalement selon l'ordre de priorité! (Malheureusement, cela ne pourra pas être votre cas ici...). Nous verrons plus en détail les activités de suivi des sprints au prochain cours.



#### Médiagraphie

MESSAGER ROTA, Véronique. *Gestion de projet – Vers les méthodes agiles*, Paris, Édition Eyrolles, 2008, 247 pages, ISBN : 978-2-212-12165

WIKIPEDIA. *Scrum*, [En ligne] <http://fr.wikipedia.org/wiki/Scrum> (Page consultée le 27 août 2010).