



# Créer un Product Backlog : Le Guide du Scrum Product Owner

Lorsqu'une entreprise décide d'implanter l'agilité et plus précisément Scrum, le Product Owner nouvellement formé comprend que la responsabilité qui lui incombe maintenant représente tout un défi. Il doit créer le product backlog initial et ensuite le maintenir. Cet artefact sera l'essence du projet.

## La création du product backlog

Les étapes de l'élaboration du product backlog présentées ici permettront de répondre aux 3 questions suivantes :

- Quelles seront les fonctionnalités à créer ?
- Dans quel ordre devront-elles être livrées ?
- À qui sont-elles destinées ?

Tout au long de ce guide, vous allez découvrir :

- les réponses aux questions ci-dessus ;
- la méthode, pas à pas, de ce que doit construire le Scrum Product Owner avant le premier Sprint planning d'un projet pour un nouveau produit.
- l'explication des projets itératifs qui émergent durant le processus Scrum.

Bien qu'il soit recommandé de faire ce travail en équipe, idéalement une équipe composée de personnes du marketing, des gens de métier et de stakeholders (sponsors), fréquemment le Product Owner fera le travail seul. S'il est issu du marketing, les premières étapes ne devraient pas poser de problèmes particuliers.

Lorsque le travail est effectué en équipe, il n'est pas toujours évident de démarrer les séances de brainstorming. Cet article présente une solution permettant d'éviter le syndrome de la page blanche lors de travail collaboratif. Cette même technique peut être appliquée à toutes les étapes de la création du product backlog.

L'important à retenir de cette technique, est qu'il faut définir des délais suffisamment longs (sans exagérer) pour permettre de passer outre les réponses évidentes. Afin d'illustrer les propos, un exemple simple sera pris : celui d'un système de gestion des locations vidéos.

## Première étape : établir la vision du projet

Il s'agit en quelques lignes de définir quel est l'objectif que le projet devra atteindre avant que le Product Owner n'en arrête le développement. Il est important que cette vision soit acceptée et comprise de tous, elle doit faire consensus puisque l'équipe va devoir se rallier derrière le PO pour réaliser ce projet selon cet objectif.

Exemple de vision pour notre système de gestion des locations vidéos:  
*« Offrir une solution de gestion complète, incluant suivi, statistiques et facturation ainsi qu'un module de prélocation en ligne et d'analyse statistique pour les vidéoclubs. »*

Bien que cela puisse paraître simple il n'est pas évident d'obtenir de la part des sponsors l'information permettant l'élaboration de cette vision, cela nécessitera bien des rencontres.


## Deuxième étape : lister les acteurs / personas

Il va falloir identifier puis détailler tous les intervenants, utilisateurs du système dans tous ses aspects. Pour chaque intervenant on voudra préciser les informations suivantes :

- Surnom : donner un Surnom aux acteurs rendra plus agréable leur utilisation et il sera plus simple de s'y identifier.
- Icône / image : ajouter une représentation graphique de l'acteur rend l'identification encore plus facile.
- Rôle : c'est en fait une description courte souvent juste un mot ou nom commun.
- Description : on décrit pourquoi et/ou comment cet acteur utilisera le système.
- Critères de satisfaction : ce qui rendra cet acteur satisfait de l'utilisation qu'il fait du système.
- Valeur commerciale : élevée, moyenne, basse, bloquante (les autres acteurs ne pourront utiliser le système).
- Fréquence d'utilisation : permanente, quotidienne, occasionnelle, rare.
- Nombre d'instances : combien d'intervenants comme celui-ci utiliseront le système. 1, 10, 100+.
- Niveau de connaissance technologique : élevée, moyenne, basse.
- Niveau de connaissance métier : élevée, moyenne, basse.

Deux exemples de personas de notre système de gestion de location de vidéos:

- *Bob*
- *Une image d'homme*
- *Commis à la location de vidéo*
- *Bob utilise le système pour enregistrer les locations de vidéos, ainsi que les retours de location.*
- *Comme Bob utilise le système en présence de clients qui attendent, Bob veut que le système soit rapide et facile à utiliser, avec une ergonomie limitant les clics.*
- *Élevée*
- *Permanente*
- *10 par installation*
- *Moyenne (souvent des jeunes à ce poste)*
- *Élevée*

	<b>Surnom</b>	<b>Bob</b>
	Rôle :	Commis à la location de vidéos
	Utilisation :	Bob utilise le système pour enregistrer les locations de vidéos, ainsi que les retours de location.
	Critères de satisfaction :	Comme Bob utilise le système en présence de clients qui attendent, Bob veut que le système soit rapide et facile à utiliser, une ergonomie limitant les clics sera désirée.
	Valeur commerciale :	Élevée
	Fréquence d'utilisation :	Permanente
	Nombre d'instances :	10 par installation
	Niveau de connaissance technologique :	Moyenne (souvent des jeunes à ce poste)
	Niveau de connaissance métier :	Élevée

- *Caroline*
- *Une image de femme*
- *Administratrice système*
- *Caroline aura la responsabilité d'installer et de configurer et de supporter le système au niveau technique.*
- *Caroline désire une installation sans soucis, une configuration souple et simple à la fois et avant tout un logiciel fiable et sans crash.*
- *Bloquante*
- *Rare*
- *1 par installation*
- *Élevée*
- *Basse*

En créant les personas, on crée également les éléments de réponse à la troisième question : À qui sont-elles destinées ?

## Troisième étape : les thèmes ou regroupement de fonctionnalités

Si on a accordé aux deux premières étapes tout le temps et le soin nécessaire, celle-ci sera facile à accomplir. De la vision du produit, on extraira facilement les thèmes principaux. Ils forment l'objectif central du produit.

Dans notre exemple, la vision est « *Offrir une solution de gestion de location vidéo complète, incluant suivi, facturation ainsi qu'un module de pré-location en ligne et d'analyse statistique pour les vidéoclubs* ».

On en tire donc facilement les thèmes suivants :

- location et suivi
- statistiques
- facturation
- pré-location en ligne

À partir des définitions des personas on peut identifier d'autres thèmes. Par exemple, la description de Bob dit « *utiliser le système pour enregistrer les locations de vidéos, ainsi que les retours de location* » et son critère de satisfaction est : « *que le système soit rapide et facile à utiliser avec une ergonomie limitant les clics* »

On peut donc extraire :

- retour de location (en considérant que le suivi concerne les retards)
- ergonomie simplifiée

La description de Caroline indique que « *Caroline aura la responsabilité d'installer et de configurer et de supporter le système ...* »

On arrive donc facilement à extraire les thèmes:

- installation du système
- configuration du système

Pour chacun de ces thèmes, il va maintenant falloir préciser certains éléments:

- Description : plus de détail concernant le thème, peut être même une liste de sous thèmes
- Valeur commerciale : c'est un indicateur très important qui guidera vos choix et priorités
- Critère de satisfaction : les critères qui feront de ce thème un succès
- Couleur de post-it : permettra de facilement identifier les stories associées
- Utilisateurs : la liste des utilisateurs que ce thème rejoint

# Comment définir la valeur commerciale ?

La méthode suivante est utilisée afin de définir la valeur commerciale de chacun des thèmes :

1. Compter le nombre de thèmes et attribuer 300 points par thème.
2. Pour chaque thème, accorder une valeur commerciale sur 1000 points.
3. Rapidement, on remarque qu'il n'y a pas assez de points pour tout. Cela oblige à différencier ce qui est vraiment une valeur ajoutée de ce qui l'est moins.

On utilise alors une évaluation comparative et on remarque rapidement 3 groupes :

- les incontournables,
- les bons thèmes
- et ceux dont on pourrait se passer.

Cette valeur permet de rapidement déterminer une priorité dans l'élaboration du projet. Cette valeur sera réutilisée plus tard pour calculer un ROI par thème.

 ROI N/A	<b>Thème</b>	<b>Locations et suivi</b>
	Description :	Enregistrement des locations de vidéofilms et suivi des retards de retours.
	Valeur commerciale :	750
	Critères de satisfaction :	Ergonomie axée sur la saisie ultra rapide. Performance et disponibilité maximales.
	Utilisateurs :	Bob Sophie ...

À la fin de cet exercice, la liste des thèmes obtenue est plus que satisfaisante et fournira la base de la création du product backlog.

# Quatrième étape: la vision de la première release

C'est le moment de commencer à prioriser sa vision et déterminer ce qui serait une solution comportant suffisamment de valeur pour être utile. Dans notre exemple, une vision de la release initiale pourrait être « *Offrir un système de location et de retour pour vidéoclub* ».

Cette vision va permettre de prioriser les thèmes pour lesquels il faudra écrire des user stories prochainement.

On comprend très bien que dans notre exemple, les thèmes statistiques ou prélocation en ligne ne feront pas partie de la release initiale. On pourra donc ne pas s'attarder dessus pour le moment.

Après cette étape on a des éléments de réponse pour la question: *Quelles seront les fonctionnalités à créer ?*, même si pour le moment il ne s'agit que de thèmes. On commence aussi à voir une idée de réponse pour: *Dans quel ordre devront-elles être livrées ?*

Parlons maintenant des éléments du product backlog à proprement dit, les « User Stories ».

## Que sont les users stories ?

Ce sont les spécifications du projet présentées sous forme d'histoires utilisateurs qui décrivent les interactions de l'utilisateur avec le système de façon plus ou moins détaillées.

### **Comment se présente une user story ?**

La partie principale est l'histoire elle-même. Elle s'écrit sous la forme suivante:  
« En tant que ...,  
Je peux ...  
Afin de ... »

Si on reprend notre ami Bob, on pourrait donc avoir une user story comme celle-ci :

*« En tant que Bob,  
Je peux créer un compte client,  
Afin de pouvoir leur louer des films »*

ou bien

*« En tant que Bob,  
Je peux entrer une location,  
Afin d'avoir une trace »*

La rédaction d'une user story se doit de répondre aux critères INVEST (independant, negotiable, valuable, estimable, small, testable).

- **Independant** (indépendante) : Une user story se doit d'être indépendante de toutes les autres stories. En prenant cette définition à l'envers, on la comprend mieux. Une user story se doit de n'avoir aucune dépendance envers les autres stories, elle doit pouvoir être réalisée individuellement. Ce besoin découle du fait qu'à tout moment, on veut pouvoir réorganiser, reprioriser le Product Backlog. Si les stories ont des dépendances, cette repriorisation est limitée.
- **Negotiable** (négociable): Toute story qui se trouve dans le backlog de produit, peut être réécrite, repensée ou même supprimée à tout moment en raison de changements du marché, de modification de l'orientation de besoins techniques ou tout autre motif valable provenant de membre de l'équipe ou des sponsors.
- **Valuable** (valeur métier) : Toutes les stories se doivent de générer de la valeur pour l'utilisateur final. À cet effet, les stories techniques, bien que fun à coder ou parfois nécessaires, n'apportent aucune valeur métier et ne devraient donc pas exister ou être incluses dans des stories régulières. On essaiera donc de les limiter.
- **Estimable** : Une user story doit pouvoir être évaluée en terme de complexité par l'équipe. Si une user story ne peut être évaluée, elle ne sera jamais planifiée ni incluse dans un sprint backlog ou découpée en tâches. Habituellement elle ne peut l'être lorsqu'elle manque d'informations de support, ou lorsque la description faite pas le Product Owner est insuffisante ou vague.

Une catégorie particulière de story : Les épics ne peuvent être évaluées puisqu'elles représentent en fait un ensemble de stories. Dans ce guide, les Thèmes remplissent le même besoin. Vous pouvez toutefois les utiliser comme pense-bête.



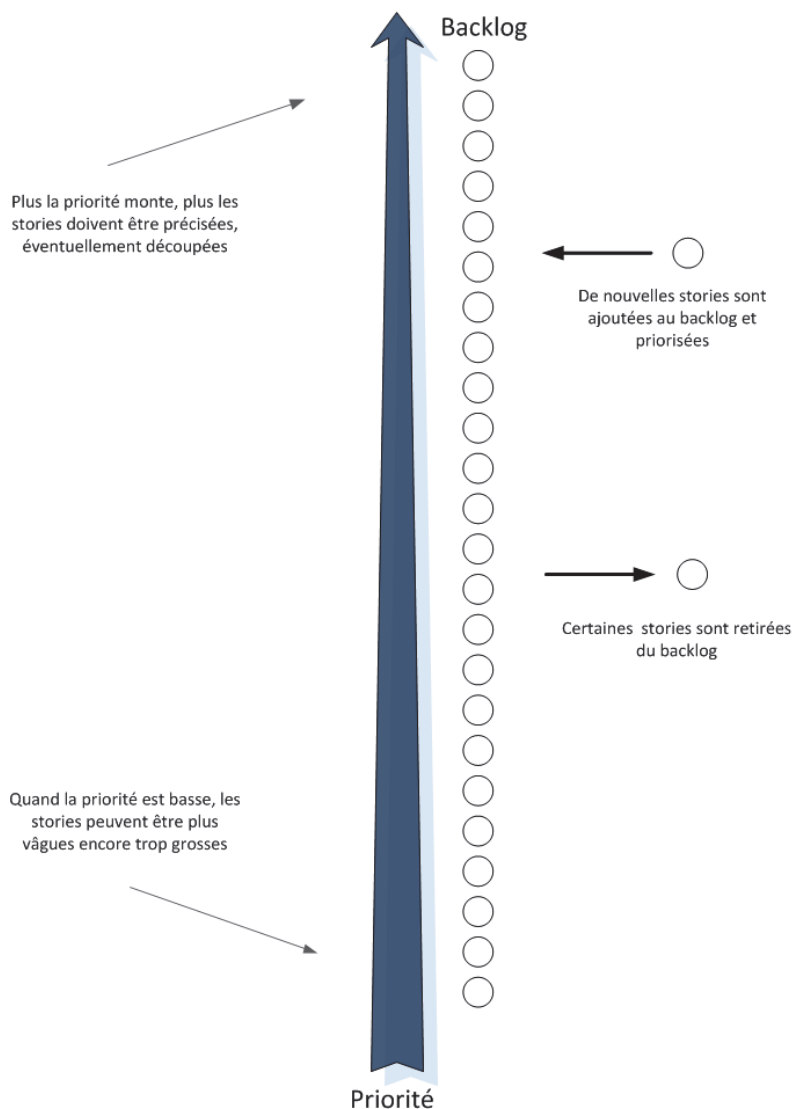
- **Small** (Petite ou de taille appropriée): Une user story doit être suffisamment petite pour permettre d'être évaluée avec le plus de précision et certitude possible. Si l'équipe évalue la complexité relativement élevée, cela démontre en général un niveau d'incertitude élevé et cela implique qu'elle devrait être découpée en plusieurs stories.
- **Testable** : Garder en tête qu'une story ne peut être considérée finie sans qu'elle ait été testée avec succès. S'il n'est pas possible de tester la story due à un manque d'information, elle ne devrait surtout pas être considérée pour être incluse dans un sprint Backlog. Ceci est encore plus vrai pour les équipes intégrant les techniques XP et particulièrement le TDD (développement piloté par les Tests).

Si une user story répond à tous ces critères sa place dans le product backlog est tout à fait justifiée. Il ne restera donc plus qu'à la prioriser. En plus de l'histoire elle-même, certaines informations utiles peuvent être ajoutées pour la définition de la story, ce qui donnera comme ensemble :

- **Thème** : le thème auquel cette story est reliée. Utiliser un post-it de la couleur définie dans le thème permet de rendre le backlog plus visuel.
- **Acteur** : l'acteur ou personas utilisé pour rédiger la user story. Là encore, si c'est possible, utiliser l'icône rend l'ensemble plus visuel.
- **Story** : utilisée lorsqu'il s'agit d'une user story.
- **Description** : utilisée lorsqu'il s'agit d'une story technique ou d'un bug.
- **Valeur** : la valeur métier. On pourra utiliser le même genre de principe que celui présenté pour les thèmes.
- **Complexité** : sera évaluée par l'équipe lors des sprints plannings ou des rencontres de travail du backlog produit (optionnelles).
- **ROI** : le retour sur investissement se calcule simplement = Valeur/Complexité. C'est un indicateur intéressant pour prioriser le product backlog.
- **Tests d'acceptation** : permet de définir quels seraient les tests d'acceptation de la story.
- **Commentaires**.

# Aspect itératif de la création et définition du Product Backlog

Plus un élément du product backlog est priorisé et donc proche d'être inclus dans un sprint, plus il se doit d'être détaillé, estimable et petit. A contrario, tant qu'un élément du product backlog est de faible priorité, il nécessite moins de détail et peut rester plus vague. On va donc au fur et à mesure de la montée en priorité définir et découper les stories en éléments plus clairs et plus petits.



Donc, en basse priorité on pourrait retrouver « *gérer les clients* ». Une fois upgradée en plus haute priorité, la story sera découpée par exemple, en « *ajouter un client, modifier un client, et supprimer un client* ».

Pour l'aspect émergence itérative, on pourra identifier les transactions centrales du système. Dans notre exemple, ce pourrait être l'écran de location de films dans lequel tous les éléments devraient être saisis manuellement. Puis on créera la gestion de la bibliothèque de films et on en utilisera les données dans la location, puis la gestion des clients et on l'utilisera dans les transactions de location.

C'est ce qu'on appelle un développement à itérations centrées autour des fonctionnalités majeures du système.

Les informations contenues dans ce guide devraient vous permettre de vous préparer pour votre premier sprint planning. Et le logiciel [Nutchache](#) sera le support idéal pour créer votre Product Backlog.



[www.nutchache.com](http://www.nutchache.com)