

Chapitre 4

Gérer le Backlog de produit

1. Organiser le Backlog de produit

Le chapitre De la vision stratégique au plan opérationnel décrit le Backlog de produit comme un référentiel ayant pour vocation d'organiser, de planifier et de communiquer :

Le Backlog produit est une liste ordonnée et émergente de ce qui est nécessaire pour améliorer le produit.

(Guide Scrum 2020)

Ce même chapitre décrit également comment la vision du produit est progressivement traduite en une stratégie et un plan opérationnel. Tout cela doit contribuer à planifier des actions dont le Backlog de produit doit refléter l'organisation. En ce sens, c'est donc un outil de communication.

Il ne s'agit pas de trier en fonction de l'urgence ou d'une priorité reposant sur une règle, mais bien d'organiser les éléments constituant le Backlog selon la stratégie établie.

Pour mieux comprendre l'organisation du Backlog, imaginez une liste de courses que l'on rédige avant d'aller réellement au supermarché. Il y a autant de manières de le faire que de ménages, et parfois deux manières différentes existent au sein du même ménage. Vous pouvez écrire les articles les uns après les autres en fonction des idées qui surviennent ou bien de ce que vous avez envie de manger. Il est possible de commencer par les promotions sur les prospectus, puis de compléter par les éléments pouvant s'accommoder avec ces premiers choix. Vous pouvez écrire en commençant par ce qui manque le plus et en terminant par les éléments les moins utiles, ou bien en fonction de la disposition des rayons dans le supermarché pour optimiser votre parcours. C'est votre organisation selon votre stratégie et votre vision des courses. Si vous revenez à la maison avec des articles qui ne sont pas sur votre liste, ce n'est pas forcément un signe d'agilité. En revanche, si vous vous rendez compte que vous avez oublié des articles, c'est que vous avez assurément besoin d'inspecter vos processus lors d'une rétrospective.

Cette simple organisation montre immédiatement au Product Owner les éventuelles défaillances pouvant exister dans sa stratégie, et affirme ou fragilise sa vision produit et la valeur du produit. S'il a des objectifs ambitieux mais peu de moyens pour y parvenir, il identifie facilement le risque.

Là encore, imaginez-le comme une liste de courses. Si vous prévoyez un barbecue avec des amis et que votre liste se résume à cinq sortes de boissons plus ou moins alcoolisées, vous aurez sûrement faim, mais vous avez identifié plus clairement une exigence d'une fête réussie selon vous. Cela illustre vos valeurs et votre vision. Il est probable qu'après la première release vous révisiez votre vision pour y incorporer un peu plus d'éléments solides.

À l'opposé, s'il dispose d'un grand nombre de moyens divergents permettant d'atteindre le même objectif, le Product Owner devra faire des choix, parfois difficiles.

Faut-il préparer une grande salade composée ou bien des pommes de terre pour le barbecue ? Il est possible de noter les deux sur la liste et de choisir en fonction de ce qui sera disponible dans le magasin.

Par défaut, et en priorité, le Backlog de produit est organisé en fonction de la valeur apportée par chaque élément selon les objectifs à atteindre. Le Product Owner privilégie les éléments apportant potentiellement le plus de valeur. Si, à la suite d'une release, la valeur potentielle n'est pas avérée, il dispose d'autres solutions qu'il pourra reprioriser, ce qui conduira à une réorganisation du Backlog.

Dans tous les cas, l'organisation du Backlog de produit reflète la vision du produit et la stratégie du Product Owner. Si vous avez un doute sur ce sujet, consultez vos récentes listes de courses.

L'organisation grossière du Backlog de produit est réalisée en fonction du plan de release du Product Owner. Les éléments placés tout en haut du Backlog de produit sont ceux qui seront traités dans les prochains Sprints de la release en cours. Ils doivent donc permettre d'atteindre l'objectif business de cette release.

1.1 Organiser en fonction de la valeur

Organiser le Backlog de produit en fonction de la valeur potentielle de chaque élément du produit nécessite bien évidemment d'être très clair sur la définition de la valeur ou des caractéristiques qui la constituent. De cette manière, l'organisation du Product Owner sera péremptoire, quelle que soit la position de celui-ci dans la hiérarchie de l'organisation. Lorsqu'elle ne repose pas sur une vision solide de la valeur du produit, l'organisation du Backlog dépend alors d'une succession de négociations ou d'argumentations plus émotionnelles que factuelles. Le Product Owner ayant passé peu de temps à définir une vision et une stratégie claires en passera beaucoup plus à gérer son Backlog et devra parfois négocier chacune de ses décisions.

Lorsque la valeur du produit est purement financière, l'organisation est assez simple : le Product Owner a intérêt à privilégier les éléments permettant potentiellement de dégager les plus gros bénéfices. Mais quand la valeur du produit est constituée de multiples caractéristiques, l'organisation peut nécessiter un peu plus de finesse. Ce facteur n'est toutefois pas exclusif, et il est possible d'en prendre d'autres en considération pour organiser le Backlog de produit, comme la complexité ou la prise de risque.

1.1.1 La valeur liée aux gains

Il peut s'agir de gains financiers, mais également de gains d'image, de confort ou de tous gains pouvant rapprocher le produit de ses objectifs.

En identifiant les gains potentiels de chaque élément de son Backlog de produit, le Product Owner peut choisir d'en privilégier un afin de maximiser les chances d'accroître la valeur ajoutée du produit.

1.1.2 La valeur liée aux coûts

Dans certains cas, la valeur du produit peut porter davantage sur une maîtrise des coûts qu'une maximisation potentielle des gains. Il peut également s'agir de la valeur d'une partie du produit, ou bien d'une release particulière ayant pour objectif de réduire les coûts de fonctionnement.

Lorsque le Product Owner doit gérer de la valeur portant à la fois sur les gains et sur les coûts, il peut simplifier son approche et communiquer sur les bénéfices nets de chaque élément de son Backlog, mais il perd alors en finesse d'analyse.

1.1.3 La valeur liée aux délais

Une organisation basée sur la valeur peut également prendre en compte les délais de mise sur le marché ou d'appropriation des nouveautés par le marché. Le Product Owner aura alors tendance à privilégier les éléments qui apportent le plus rapidement de la valeur ajoutée.

1.1.4 La valeur liée à des objectifs ou contraintes secondaires

Enfin, le Product Owner peut organiser son Backlog de produit en fonction de la valeur liée à l'atteinte d'objectifs ou au respect de contraintes. Cette approche est particulièrement intéressante lorsqu'il existe plusieurs solutions permettant d'atteindre un objectif. Si les notions de coût et de délai ne les départagent pas franchement, le Product Owner peut privilégier la solution promettant de mieux atteindre l'objectif ou respectant davantage les contraintes. Bien sûr, cette notion de valeur peut être prioritaire sur les coûts, auquel cas le Product Owner organisera son Backlog en priorité par rapport à cet indicateur et affinera en fonction des coûts ou des bénéfices escomptés.

Quelle que soit la manière dont le Product Owner choisit d'organiser son Backlog de produit, il ne doit pas céder à l'usage intempestif d'algorithmes. Il est préférable d'organiser le Backlog en fonction de plusieurs indicateurs, dont le poids peut varier dans la durée, plutôt que d'avoir un seul indicateur basé sur une équation mathématique.

1.1.5 La valeur de quel point de vue ?

Lors des estimations de valeur, il faut veiller à ne pas confondre plusieurs points de vue. Par exemple, l'ajout d'encarts publicitaires dans une solution digitale apporte une valeur financière à la compagnie, mais cela réduit l'espace alloué au contenu et intègre un contenu ayant une bien plus faible valeur ajoutée du point de vue du consommateur. Dans certains contextes, le Product Owner peut définir plusieurs attributs de valeur en fonction de différentes cibles.

1.2 Organiser en fonction de la complexité

La valeur que peut apporter chaque élément au produit est à prendre en considération pour maximiser la valeur potentielle du produit final, et c'est pour cela que cet aspect est considéré comme le premier et principal critère de l'organisation du Backlog de produit. Mais les estimations de l'équipe de développement peuvent jouer un rôle déterminant dans l'organisation du Backlog. Ces estimations se présentent généralement sous la forme d'une complexité, bien que l'équipe puisse utiliser un autre référentiel de son choix (durée, charge, effort...).

C'est particulièrement vrai lorsque plusieurs solutions permettent potentiellement d'atteindre le même objectif. Dans ce cas, le Product Owner peut choisir de privilégier la solution apportant potentiellement le plus de valeur, mais si cette solution est trois fois plus complexe à mettre en œuvre que sa concurrente, il pourrait revoir sa priorisation.

La complexité peut également influencer l'organisation du Backlog de produit, en reportant la réalisation d'un élément qui n'aurait pas encore été suffisamment affiné par l'équipe Scrum.

Enfin, et surtout, la complexité influencera l'organisation du Backlog de produit par rapport à la capacité de production de l'équipe.

Si le Product Owner a identifié que dans trois Sprints, il faudrait livrer vingt éléments pour atteindre l'objectif, mais que l'équipe n'est capable de n'en livrer que cinq ou six par Sprint, il devra prioriser son Backlog en sélectionnant ceux qui apportent le plus de valeur.

Lorsque la complexité des éléments varie, le Product Owner peut alors affiner son choix et privilégier un plus grand nombre d'éléments plus faciles à réaliser à un nombre restreint d'éléments plus complexes.

1.2.1 La complexité de mise en œuvre

La complexité de mise en œuvre est toujours et exclusivement estimée par la ou les équipes de développement. Toute l'équipe Scrum peut consulter des experts pour affiner ses estimations, mais c'est l'équipe de développement qui aura toujours le dernier mot. L'estimation de complexité peut être réalisée en jours/homme, en points de complexité ou en toute autre unité de mesure compréhensible par un maximum de personnes impliquées par le produit. De nombreuses équipes utilisent des Story Points, ou points de complexité, permettant de comparer la complexité des différents éléments du Backlog de produit. Si l'élément n° 5 a une complexité estimée à 10 points et que celle de l'élément n° 7 est estimée à 20 points, alors tout le monde s'accorde à dire qu'il est deux fois plus complexe, ce qui ne signifie pas forcément qu'il sera deux fois plus long à réaliser. Ces pratiques seront détaillées au prochain chapitre.

1.2.2 La complexité d'exploitation

Dans certains cas, le Product Owner peut s'appuyer sur une complexité d'exploitation ou de déploiement. C'est particulièrement intéressant lorsqu'il doit gérer des produits devant être déployés sur des sites multiples, à l'international ou auprès d'un nombre important de partenaires.

Il cherchera alors à identifier la complexité d'exploitation ou de déploiement sur les différents marchés pour affiner sa stratégie. Commencer par les marchés les plus complexes ou au contraire par les plus simples dépend de la stratégie du Product Owner. L'ordonnancement du Backlog de produit n'en sera que le reflet.

1.3 Organiser en fonction de la prise de risque

La valeur et la complexité sont deux critères essentiels d'organisation du Backlog de produit, mais dans certains cas il est préférable d'en ajouter d'autres. La notion de risque ou de prise de risque est un critère de qualification fréquemment utilisé.

Par exemple, un Product Owner devant déployer un produit vers une cible réticente aura tendance à privilégier les fonctionnalités les moins risquées pour atténuer la défiance initiale, plutôt que de déployer les fonctionnalités apportant une plus grande valeur mais présentant un risque de rejet total. Ainsi, il peut décider de privilégier le déploiement des fonctionnalités gratuites en premier, puis d'intégrer progressivement des éléments payants ou nécessitant une identification puis un abonnement. C'est la stratégie adoptée par plusieurs plateformes de diffusion de contenus culturels.

À l'inverse, certains acteurs préfèrent proposer des fonctionnalités très innovantes, à forte valeur ajoutée, impliquant une précommande. C'est le cas de nombreux studios de développement de jeux vidéo.

Il n'existe pas de recette miracle, c'est une question de stratégie produit par rapport à un marché. Le fait de fixer des attributs aux composants du produit et d'en identifier l'ampleur permet au Product Owner de faire accepter sa stratégie sur la base de données mesurables plutôt que sur sa seule intuition.

1.4 Organiser selon d'autres critères

Le délai d'appropriation par un marché peut être un critère à prendre en compte par le Product Owner. Dès la première release, cette notion de délai, initialement estimée, pourra être mesurée de manière empirique et servir d'indicateur fiable pour les releases suivantes.

Cette mesure peut amener le Product Owner à réviser en cours de route sa planification initiale. Si un marché n'est pas prêt à recevoir un produit, il est vraisemblablement plus intéressant de porter les efforts à court terme vers d'autres marchés ou d'autres secteurs.

Chapitre 5

Construire et prioriser le Product Backlog

1. Pourquoi investir dans le Product Backlog ?

Que l'on soit en contexte Agile ou pas, il est assez évident que pour réussir un projet nous devons posséder une bonne vision du système ou produit que l'on va développer, par le biais d'une définition pertinente du besoin auquel il répond.

En Scrum donc, pas de projet réussi sans un Product Backlog bien construit et intelligemment priorisé. Pour ce faire, évidemment, le Product Owner joue un rôle fondamental, par sa connaissance du métier et/ou du marché, mais aussi par sa capacité à transcrire efficacement, découper et ordonner ce qui est attendu par les utilisateurs du logiciel.

Là où nous avons l'habitude en tant que « Client » d'écrire les spécifications fonctionnelles détaillées faisant office d'expression des besoins, nous allons devoir nous familiariser avec une nouvelle méthode. Cette familiarisation demande un temps d'apprentissage et se doit d'être partagée par l'ensemble des parties prenantes du projet, ce qui signifiera formation, assimilation et mise en pratique des nouvelles techniques.

Dans ce chapitre, nous allons donc parcourir les concepts et méthodes permettant de construire, ordonner, affiner et gérer au quotidien le Product Backlog.

2. La brique de base du Product Backlog : la User Story

Aussi étonnant que cela puisse paraître Scrum ne prescrit pas un format ou un contenu précis du Backlog ! Pour autant, un formalisme s'est imposé et on le retrouve quasiment systématiquement sur tous les projets Scrum : il s'agit de la notion de **User Story**, qui est empruntée à XP.

Une User Story est une description simple et compréhensible d'un élément de fonctionnalité à valeur « métier » du système. Elle est donc bien exprimée du point de vue de l'utilisateur. Et elle est guidée par la réponse à ces trois questions :

- Qui fait la demande ou qui bénéficie de la demande ? (rôle utilisateur)
- Quelle est la demande ? (le besoin)
- Quelle valeur métier découle de la réalisation de ce besoin ?

Ci-dessous quelques exemples (nous verrons plus loin comment bien rédiger les User Stories) :

« Je souhaite que la TVA soit automatiquement calculée sur les factures ».

« Je souhaite pouvoir supprimer les clients n'ayant pas passé de commandes depuis plus d'un an ».

En complément, on emploie aussi la notion d'**Epic Story**. On peut considérer une Epic comme une « **macro User Story** », c'est-à-dire qu'elle englobe dans sa définition un sous-ensemble de User Stories.

Par exemple, en relation avec les User Stories décrites précédemment, nous pourrions avoir les Epics suivantes :

« Je souhaite que mes factures soient établies automatiquement ».

« Je souhaite avoir une gestion de mes clients ».

C'est donc l'ensemble des Epic et User Stories que constitue le Product Backlog.

Il est en pratique utile de regrouper les Epic ou User Stories (en particulier pour la priorisation) : pour ce faire, on emploie communément les concepts de **Thèmes** ou **Activités**, qui sont des regroupements thématiques de Stories.

3. Comment rédiger les User Stories et Epics ?

3.1 Règle des 3C

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 :

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

En pratique, on écrira donc la Story sur une petite carte de couleur colorée épinglée ou collée sur un tableau (principe hérité du Kanban, si vous vous souvenez bien), sous la forme suivante :

- On commence avec un titre.
- On ajoute une description synthétique de la « tâche utilisateur » et de ses objectifs.
- On peut y ajouter toutes les notes, dessins ou informations utiles. Exemple : chiffre, indicateur de priorité ou valeur métier.
- On y ajoute idéalement les critères de validation (par exemple au dos si on n'a plus de place...).

Cela donnera, dans la forme la plus simple, quelque chose qui ressemble à ceci :



La carte est un concept de partage de l'information extrêmement synthétique et efficace, mais dès que des données additionnelles s'ajoutent, au fur et à mesure de l'analyse, attention à ne pas surcharger celle-ci. On voit assez vite venir la nécessité de passer par un outillage plus sophistiqué (et informatisé) pour gérer l'information : nous parlerons de cela plus loin...

3.2 Rédiger une bonne User Story : le principe INVEST

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.
 - Nous verrons dans le chapitre dédié aux tests comment formaliser cela.

3.3 Erreurs courantes

Pour bien rédiger nos Stories, il est aussi très instructif de savoir quelles erreurs éviter :

- Trop détailler la description et rentrer trop tôt dans un niveau de détail fin. N'oublions pas qu'on ne cherche pas à faire les spécifications détaillées complètes préalablement au développement comme dans les méthodes traditionnelles !
- Perdre la notion utilisateur dans la description ou utiliser des acteurs trop génériques : cela peut créer des ambiguïtés et imprécisions

Exemple :

- En tant que client abonné, je veux pouvoir consulter les tarifs des billets d'avion.
- En tant que client standard, je veux pouvoir consulter les tarifs des billets d'avion.

Plutôt que :

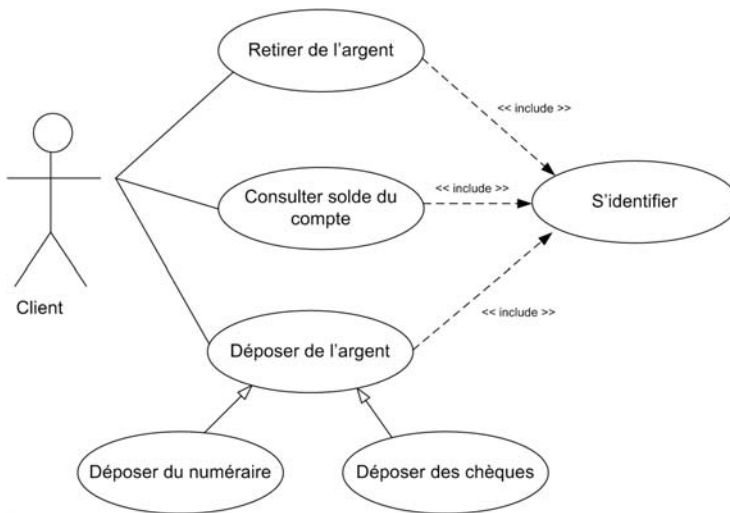
- En tant qu'utilisateur, je veux pouvoir consulter les tarifs des billets d'avion.

En effet, on voudra vraisemblablement proposer des offres ou des tarifs différents aux abonnés par rapport aux autres personnes qui consultent les tarifs, ce qui n'apparaît pas dans la seconde description.

- Partir d'un cahier des charges rédigé en amont et le découper en User Stories en s'appuyant aveuglément sur sa structure textuelle. Cela peut être une approche commode pour une équipe débutante mais n'est pas une pratique recommandée (nous verrons plus loin une méthode pratique pour initialiser le Product Backlog).
- Vouloir avoir un niveau de détail constant. Ce point est très important, car il est au cœur de la démarche Agile. Le contenu des User Stories est amené à évoluer au fil du temps, au fur et à mesure de l'affinage de la compréhension du besoin et de l'analyse. Typiquement, une Story dont la réalisation est prévue dans plusieurs Sprints ne sera décrite que de manière très simple (titre, description synthétique), alors qu'une User Story dont la réalisation est proche doit être complétée par des tests de recette, des exemples, des règles de gestion, des maquettes d'écran, etc.

- Assimiler une User Story à un **Use Case**. Bien que la comparaison entre les deux notions soit souvent utile, ce sont deux approches aux caractéristiques sensiblement différentes.

Sans rentrer dans un niveau de détail trop fin, on peut dire qu'une modélisation à base de Use cases est une description de processus (par définition, un Use case représente une séquence d'actions qu'un système ou toute autre entité peut accomplir en interagissant avec les acteurs du système), qui s'appuie souvent sur des diagrammes UML, comme dans l'exemple ci-dessous :



3.4 La Story technique : solution ou aveu d'échec ?

Disons en préambule que ce thème est éminemment polémique dans la communauté Agile, car certains considèrent que le Product Backlog ne doit contenir que des éléments qui ont de la valeur d'un point de vue utilisateur ou métier alors que d'autres disent que finalement Scrum ne dit rien de précis à ce sujet, donc on peut inclure des sujets à vocation purement technique dans le Backlog.