

## **Le product backlog et les users stories**

**Aymen Sioud**  
**Département Informatique et Mathématique**  
**Courriel : [aymen.sioud@uqac.ca](mailto:aymen.sioud@uqac.ca)**

# Tables des matières

---

- ▶ Le product backlog
- ▶ Le user story
  - ▶ Introduction
  - ▶ Rédaction
  - ▶ Identifier les fonctionnalités
- ▶ Outils de découverte du produit
  - ▶ Product Box
  - ▶ Le Story mapping
  - ▶ Affiner le PB
- ▶ Priorisation du PB
  - ▶ Principes
  - ▶ Thèmes
  - ▶ Kano
  - ▶ Moscow
  - ▶ WSJF



# Le product backlog



# Le product backlog (PB)

---

- ▶ Épine dorsale de tout projet Scrum.
- ▶ **Liste priorisée de tout ce qui doit être fait dans le projet, définie par le Product Owner.**
- ▶ Les éléments du product backlog, ou "user stories :
  - ▶ fonctionnalités
  - ▶ Exigences
  - ▶ améliorations
  - ▶ corrections

# Pourquoi investir dans le PB

---

- ▶ **Clarté des objectifs du projet**
  - ▶ Vue d'ensemble claire des objectifs du projet
  - ▶ Chaque élément du backlog a un but précis
- ▶ **Priorisation efficace**
  - ▶ Les éléments les plus importants et les plus précieux pour le produit final sont toujours traités en premier.
  - ▶ Travailler d'abord sur ce qui rapporte plus de valeur
- ▶ **Planification flexible**
  - ▶ PB régulièrement revu et réajusté en fonction des retours d'information des stakeholders et des changements
- ▶ **Transparence**
  - ▶ Tous les membres de l'équipe y ont accès et peuvent voir quel travail a été accompli, quel travail est en cours et quel travail est prévu.
- ▶ **Communication améliorée**
  - ▶ Point central pour les discussions sur le produit.



# Les user stories



# User story

---

- ▶ Élément fondamental du product backlog.
- ▶ Description simple et compréhensible d'une fonctionnalité vue par l'utilisateur du produit.
- ▶ Aide à comprendre le besoin de l'utilisateur et à concevoir une solution qui y répond.

# User story

## Description

---

- ▶ Brève description du besoin d'un utilisateur.
- ▶ Formulation
  - ▶ "En tant que [type d'utilisateur], je veux [faire quelque chose] afin de [bénéficier de quelque chose]".
  - ▶ Par exemple, dans le contexte de l'application de fitness de notre équipe, une user story pourrait être :
  - ▶ "En tant qu'utilisateur abonné, je veux pouvoir suivre mes progrès quotidiens afin de voir si je m'améliore".

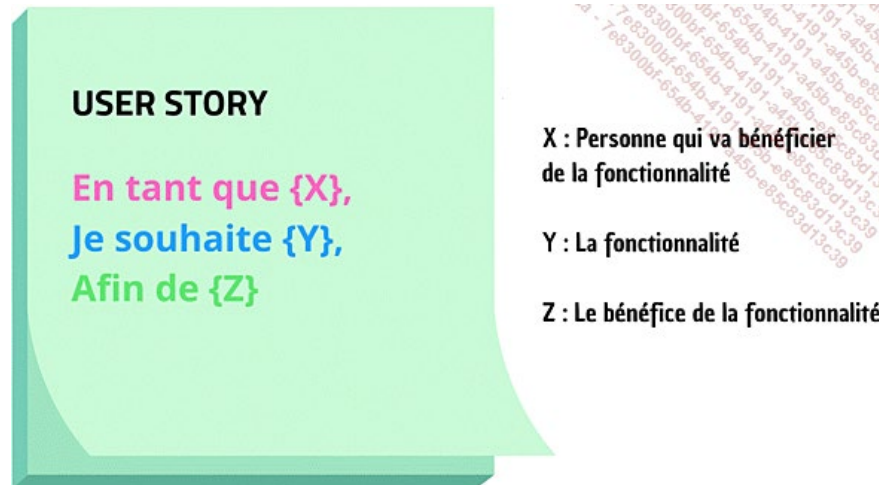


# User story

## Formulation

---

- ▶ Cette formulation aide l'équipe à se concentrer sur l'utilisateur et sur la valeur que la fonctionnalité apporte.
- ▶ Elle aide également à maintenir la fonctionnalité à une taille gérable.
- ▶ Si la user story devient trop complexe, elle peut être décomposée en plusieurs user stories plus petites.



# Importance des user stories

---

- ▶ **Centrées sur l'utilisateur**
  - ▶ Penser du point de vue de l'utilisateur et à concevoir des fonctionnalités qui répondent à leurs besoins.
- ▶ **Simplicité**
  - ▶ Pas de détails techniques
- ▶ **Facilité de priorisation**
  - ▶ En fonction de la valeur qu'elles apportent à l'utilisateur pour se concentrer sur le plus importantes
- ▶ **Flexibilité**
  - ▶ Peuvent être ajoutées, modifiées ou supprimées du product backlog à tout moment (adaptation aux changements)
- ▶ **Promotion de la collaboration**
  - ▶ Encouragent la collaboration entre le Product Owner et l'équipe de développement

# Epics

---

- ▶ Un epic est une grande unité de travail qui est trop volumineuse pour être réalisée en un seul sprint.
- ▶ Une large fonctionnalité ou d'un objectif majeur qui doit être décomposé en plusieurs petites tâches plus gérables, connues sous le nom de "user stories".

# User stories et PB

---

- ▶ La rédaction des user stories et des epics est une étape cruciale de la gestion du product backlog.
- ▶ Il s'agit de transformer les besoins des utilisateurs et les grandes idées en tâches concrètes que l'équipe de développement peut réaliser.

# Rédaction des user stories et epics

---

- ▶ "En tant que [rôle], je veux [action] pour que [bénéfice]".
  - ▶ Formulation permettant de garder le focus sur l'utilisateur et la valeur que la fonctionnalité lui apporte.
- ▶ Exemple
  - ▶ "En tant qu'utilisateur, je veux pouvoir enregistrer mes séances d'entraînement pour pouvoir suivre mes progrès".
- ▶ Rester concis et de se concentrer sur l'essentiel.
- ▶ Les détails techniques et les solutions spécifiques ne doivent pas être inclus dans la user story.
  - ▶ Discutés plus tard avec l'équipe de développement lors de la planification du sprint.

# Rédaction epics

---

- ▶ Les epics sont souvent décomposés en plusieurs user stories plus petites.
- ▶ Se rappeler qu'ils ne sont pas destinés à être réalisés tels quels.
  - ▶ Ils servent de grandes idées qui seront décomposées en user stories plus petites et plus gérables.
- ▶ Exemple
  - ▶ En tant qu'utilisateur, je veux pouvoir suivre et analyser toutes mes activités physiques pour améliorer ma condition physique".
  - ▶ Cet epic pourrait être décomposé en plusieurs user stories, comme
    - ▶ "En tant qu'utilisateur, je veux pouvoir enregistrer mes séances de course à pied"
    - ▶ "En tant qu'utilisateur, je veux voir des graphiques de mes progrès".

# Qui rédige les user stories et les epics

---

- ▶ Compétence clé pour le Product Owner.
- ▶ Transformer les besoins des utilisateurs et les grandes idées en tâches concrètes que l'équipe de développement peut réaliser, tout en maintenant une vision globale du produit.

# La règle des 3C

---

- ▶ Pratique courante lors de la rédaction de user stories dans le cadre de la méthode Scrum.
  - ▶ la Carte (Card)
  - ▶ la Conversation
  - ▶ la Confirmation.



# La carte

---

- ▶ C'est la user story en elle-même.
- ▶ Généralement écrite sur une carte physique ou un post-it, ou elle peut être numérique, dans un outil de gestion de projet.
- ▶ La carte doit contenir l'essence de la fonctionnalité souhaitée, généralement en utilisant le format "En tant que (rôle), je veux (action) pour que (bénéfice)".
- ▶ Généralement assez succincte, laissant de la place pour la discussion et l'élaboration.
- ▶ Exemple
  - ▶ "En tant qu'utilisateur, je veux pouvoir enregistrer mes séances d'entraînement pour suivre mes progrès."

# La conversation

---

- ▶ C'est la discussion qui se tient autour de la carte.
- ▶ C'est lors de cette conversation que l'équipe Scrum, y compris le Product Owner, le Scrum Master et l'équipe de développement, discute des détails de la user story, comme sa faisabilité, les obstacles potentiels et comment elle sera réalisée.
- ▶ C'est un moment crucial pour clarifier les attentes et partager les connaissances.

# La confirmation

---

- ▶ C'est l'accord sur la façon dont l'équipe saura que la user story a été complétée avec succès.
- ▶ Cela inclut généralement des critères d'acceptation qui définissent précisément ce qui est attendu du point de vue fonctionnel.
- ▶ Exemple
  - ▶ Pour la user story sur l'enregistrement des séances d'entraînement, un critère d'acceptation pourrait être :
    - ▶ "L'utilisateur peut entrer le type d'activité, la durée, la distance parcourue et l'intensité de l'effort lors de l'enregistrement d'une séance d'entraînement".

# Exemple 3C

---

## ▶ Card

- ▶ "En tant qu'utilisateur, je veux pouvoir partager mes progrès sur les réseaux sociaux pour pouvoir motiver mes amis à s'entraîner avec moi.»

## ▶ Conversation

- ▶ Cela signifie quoi "partager des progrès"
  - ▶ Est-ce que cela signifie partager des statistiques d'entraînement spécifiques ?
  - ▶ Ou peut-être partager des photos de l'entraînement lui-même ?
  - ▶ Quels réseaux sociaux devraient être pris en compte ?
  - ▶ Facebook ? Instagram ? Twitter ? Ou tous les trois ?
  - ▶ Comment cela devrait-il être mis en œuvre techniquement ?
  - ▶ Y a-t-il des problèmes de confidentialité à prendre en compte ? Tout cela est discuté au cours de la conversation.

# Exemple 3C

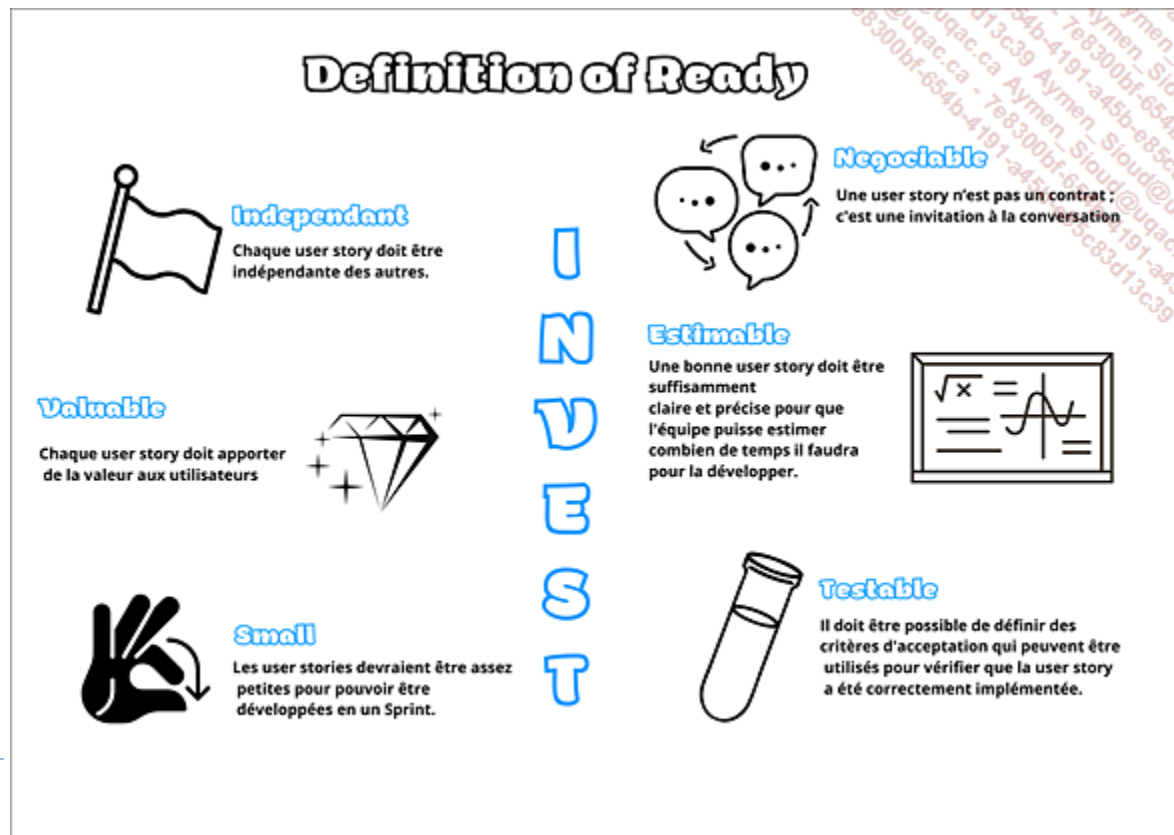
---

## ► Confirmation

- Un ensemble de critères d'acceptation qui définissent précisément ce qui doit être fait pour que la user story soit considérée comme "terminée" :
  - L'utilisateur peut sélectionner un bouton Partager après avoir terminé un entraînement.
  - L'utilisateur peut choisir quels détails de l'entraînement partager (par exemple, le type d'entraînement, la durée, les calories brûlées).
  - L'utilisateur peut sélectionner sur quel(s) réseau(x) social(aux) il souhaite partager ces détails.
  - L'application doit demander la permission de l'utilisateur pour accéder à son compte sur le réseau social sélectionné avant de publier.
  - L'application doit publier les détails sélectionnés de l'entraînement sur le compte du réseau social de l'utilisateur, avec un message préétabli qui peut être personnalisé par l'utilisateur.

# Les principes INVEST

- ▶ Pour qu'une user story soit efficace et exploitable, elle doit suivre les 6 principes INVEST.
- ▶ Permettre d'assurer que les user stories sont bien écrites et qu'elles apportent de la valeur à l'équipe et au projet.



# 6 principes assurant la qualité

---

- ▶ Ces principes permettent d'avoir des US de qualité, sont claires, gérables et axées sur la fourniture de valeur à vos utilisateurs et à vos parties prenantes..
- ▶ Exemple
  - ▶ "En tant qu'utilisateur, je veux pouvoir partager mes progrès de fitness sur les réseaux sociaux afin que mes amis puissent voir mes progrès."

# Indépendant (I)

---

- ▶ Peut être développée, testée, et livrée séparément, sans dépendre d'autres user stories.
- ▶ Cette indépendance permet une plus grande flexibilité dans la planification et la priorisation.
- ▶ Exemple
  - ▶ User story est indépendante car peut être développée et testée séparément des autres fonctionnalités de l'application.
  - ▶ Sa mise en œuvre ne dépend pas d'autres user stories.



# Négociable (N)

---

- ▶ Une user story est pas un contrat ; c'est une invitation à la conversation.
- ▶ Les détails exacts de ce qui doit être fait ne sont pas définis à l'avance et sont négociables, généralement entre le Product Owner et l'équipe de développement.
- ▶ Exemple
  - ▶ Quelle plateforme de médias sociaux sera intégrée ?
  - ▶ Quels aspects des progrès de l'utilisateur seront partagés ?
  - ▶ Ces détails peuvent être discutés et définis par le Product Owner et l'équipe de développement.

# Valeur (V)

---

- ▶ Chaque user story doit apporter de la valeur aux utilisateurs ou aux parties prenantes de l'application.
  - ▶ Si une user story n'apporte pas de valeur, pourquoi la développer ?
- ▶ Exemple
  - ▶ Apporte de la valeur à l'utilisateur, car elle lui permet de partager ses réussites avec ses amis, ce qui peut l'encourager à continuer ses efforts de fitness.

# Estimable (E)

---

- ▶ Une bonne user story doit être suffisamment claire et précise pour que l'équipe puisse estimer combien de temps il faudra pour la développer.
- ▶ Si une user story est trop vague ou trop grande, il peut être nécessaire de la diviser en plusieurs user stories plus petites.
- ▶ Exemple
  - ▶ L'équipe de développement a suffisamment d'informations pour estimer le travail nécessaire à la réalisation de cette user story.
  - ▶ Ils connaissent les plateformes de médias sociaux, et ils ont déjà développé des fonctionnalités liées aux progrès de fitness.

# Small (Petite) (S)

---

- ▶ Les user stories devraient être assez petites pour pouvoir être développées en un sprint.
- ▶ Si une user story est trop grande, elle devrait être décomposée en plusieurs user stories plus petites.
- ▶ Exemple
  - ▶ Est assez petite pour être développée en un Sprint.
  - ▶ Si ce n'était pas le cas, nous pourrions la décomposer en plusieurs user stories plus petites (par exemple, une pour chaque plateforme de médias sociaux).

# Testable (T)

---

- ▶ Une user story doit être testable.
- ▶ Il doit être possible de définir des critères d'acceptation qui peuvent être utilisés pour vérifier que la user story a été correctement implémentée.
- ▶ Exemple
  - ▶ "L'utilisateur peut choisir de partager ses progrès sur Facebook",
  - ▶ "L'utilisateur peut choisir quels aspects de ses progrès partager",
  - ▶ "Les progrès partagés apparaissent correctement sur la plateforme de médias sociaux".

# Quelques pièges à éviter

---

- ▶ User stories trop grandes (épique)
- ▶ Absence de l'utilisateur
- ▶ Détails techniques dans la user story
- ▶ Absence de critères d'acceptation
- ▶ Pas assez de discussion

# Les stories techniques 😊

---

- ▶ Une story technique se concentre sur les aspects techniques ou non fonctionnels du système.
  - ▶ Exemple
    - ▶ Optimiser de la performance, à la maintenance du code ou à la mise à niveau de certaines infrastructures.
- ▶ Pas de valeur métier et pas orientée utilisateur
- ▶ Mais permet d'aborder les aspects techniques du projet et permet d'intégrer ces tâches dans le processus Scrum de manière transparente...
- ▶ Mais...

# Stories techniques

## Un juste équilibre

---

- ▶ Peuvent être perçus comme une solution car elles permettent de prendre en compte des éléments essentiels du projet qui ne sont pas directement liés à une fonctionnalité utilisateur.
- ▶ Mais on ne se concentre pas sur la valeur apportée à l'utilisateur.
  - ▶ Si nous avons trop de stories techniques, cela peut indiquer que nous ne parvenons pas à relier suffisamment notre travail technique aux besoins de l'utilisateur.
  - ▶ Peut être le signe que l'équipe n'est pas assez "auto-organisée" ou "cross-fonctionnelle"



# Outils de découvertes du produit

# Product box ☺

---



# Product Box

---

- ▶ Technique de visualisation couramment utilisée dans les projets agiles pour aider à identifier et à prioriser les fonctionnalités essentielles d'un produit.
- ▶ Concevoir la "boîte" de votre produit, comme s'il s'agissait d'un produit physique vendu en magasin.

# Pourquoi le Product Box

---

- ▶ Permettre à l'équipe de développement de comprendre clairement quels sont les aspects du produit qui apportent le plus de valeur aux utilisateurs.
  - ▶ C'est une excellente occasion pour l'équipe de se mettre à la place des utilisateurs et de voir le produit du point de vue de l'utilisateur final.
- ▶ Aider à prioriser les fonctionnalités du produit.
  - ▶ En visualisant, l'équipe peut identifier rapidement quelles sont les fonctionnalités qui méritent d'être mises en avant sur la boîte et donc, quelles sont les plus importantes pour les utilisateurs.

# Fonctionnement du Product Box

---

- ▶ Session de brainstorming.

1. Chaque membre de l'équipe dispose d'une "boîte" en carton et de divers matériaux de décoration, comme des marqueurs, des stickers, des images, etc.
2. Chaque membre de l'équipe doit alors concevoir sa propre "boîte de produit", en se concentrant sur la représentation des fonctionnalités qu'ils estiment être les plus importantes pour le produit.
  - ▶ Ils peuvent utiliser des mots, des images, des symboles, etc., pour représenter ces fonctionnalités sur leur boîte.
3. Une fois que tout le monde a terminé, chaque membre de l'équipe présente sa boîte aux autres, expliquant pourquoi ils ont choisi de mettre en avant certaines fonctionnalités et comment ils ont représenté ces fonctionnalités sur leur boîte.
4. L'équipe discute ensuite des différentes boîtes et des fonctionnalités qui ont été mises en avant.
  - ▶ Cette discussion peut aider à identifier les fonctionnalités qui sont vraiment essentielles pour le produit et celles qui peuvent être de moindre priorité.



# Le story mapping



# À la découverte du product backlog

---

- ▶ Le story mapping est une technique de visualisation qui aide à comprendre la structure et la fonction d'un produit en décomposant les fonctionnalités en user stories.
- ▶ Cette technique peut être particulièrement utile pour planifier les releases et identifier les fonctionnalités minimales indispensables pour une première version du produit.
- ▶ L'approche travaille sur 2 axes :
  - ▶ horizontal (nouvelles fonctionnalités)
  - ▶ vertical (affinage des fonctionnalités)

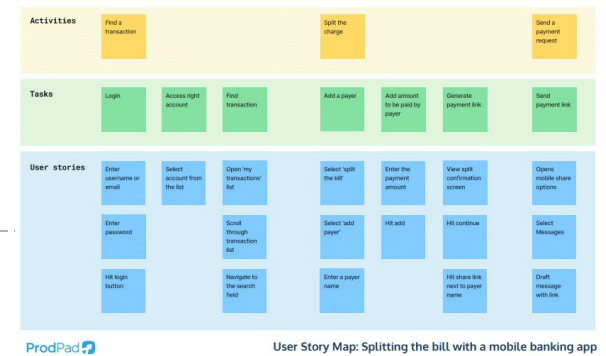
# Exemple





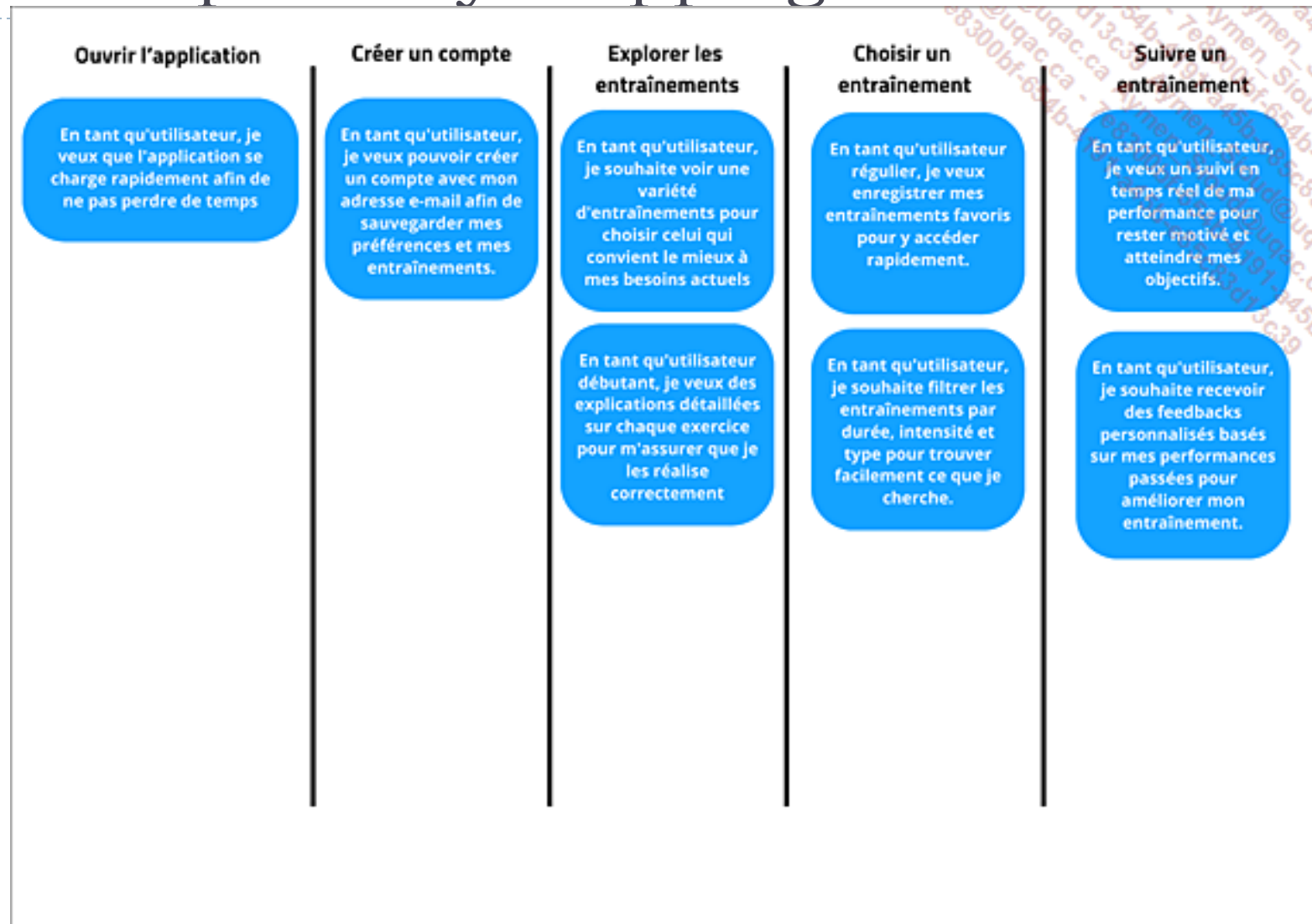
# Le story mapping

- ▶ Inventé par Jeff Patton



- ▶ Permet de créer une représentation physique des user stories pour aider à la compréhension et la planification du produit.
- ▶ Les user stories sont organisées en plusieurs rangées horizontales, généralement selon leur ordre d'importance pour le produit.
- ▶ Chaque colonne verticale représente une étape du parcours de l'utilisateur, de l'ouverture de l'application à la réalisation de la tâche finale.

# Exemple story mapping





# Affinage du PB



# L'affinage c'est quoi ?

---

- ▶ *L'affinage du Backlog de produit consiste en l'ajout de détails, d'estimations et de l'ordonnancement des éléments du Backlog de produit. Il s'agit d'une activité régulière dans laquelle le Product Owner et l'équipe de développement collaborent pour détailler les éléments du Backlog de produit.*

(Guide Scrum 2017)

# Pourquoi affiner un backlog ?

---

- ▶ La planification du sprint repose sur un backlog affiné, et l'affinement du backlog permet de passer au travail le plus prioritaire pour le sprint suivant.
- ▶ Le PO devrait affiner le backlog produit avant chaque réunion de planification du sprint.
- ▶ Dans certains cas, on introduit des réunions d'affinage/affinement

# Affinage : deux activités

---

- ▶ L'organisation ou l'ordonnancement des éléments du Backlog de produit dans celui-ci.
  - ▶ Réalisée sous la responsabilité exclusive du Product Owner, mais elle peut être partiellement ou totalement déléguée à n'importe qui dans l'équipe Scrum
- ▶ La clarification de chaque élément du Backlog de produit avec au minimum :
  - ▶ sa description,
  - ▶ son périmètre,
  - ▶ sa valeur,
  - ▶ une estimation (de délai, de charge ou de complexité) réalisée par l'équipe de développement après avoir parfaitement compris les trois points précédents.

# Affiner, on commence où ?

---

- ▶ Pourquoi cet élément devrait-il être réalisé ?
  - ▶ Tant que la réponse à cette question est ambiguë, contestable ou suscite d'autres questions, il faut affiner ce point.
  - ▶ S'il subsiste des doutes ou que cet élément ne fait pas sens avec le reste, malgré les efforts de l'équipe Scrum pour l'affiner, alors il faut le sortir du Backlog de produit, tout simplement.
- ▶ Cette phrase est essentielle et chaque Product Owner devrait se l'approprier pleinement, la mémoriser, en faire son credo.

# Qui affine ?

---

- ▶ Le PO
- ▶ L'équipe
- ▶ Les parties prenantes



# Quand affiner ?

---

- ▶ À tout moment et dès que possible.
- ▶ La réunion de planification de Sprint (Sprint Planning) est un évènement consacré à cette activité
  - ▶ Masi pas exclusive et ne suffit pas toujours à réaliser toutes les activités nécessaires pour arriver à un niveau de transparence idéal.
  - ▶ Il est préférable d'avoir rendu les éléments le plus clairs possible avant d'entamer le Sprint Planning.

# Jusqu'à quel niveau affiner ?

---

- ▶ Le raffinement n'occupe généralement pas plus de 10 % de la capacité de l'équipe de développement. Toutefois, les éléments du Backlog de produit peuvent être modifiés à n'importe quel moment par le Product Owner ou à sa discrétion.

(Guide Scrum 2017)

- ▶ Pour les éléments les plus lointains, qui seront probablement réalisés au-delà des deux ou trois prochains Sprints, il suffit que leur description et leur raison d'être soient claires.
- ▶ Les estimations sont grossières et le périmètre peut même être encore assez flou.
- ▶ Pour les éléments les plus lointains, l'équipe Scrum n'a peut-être pas encore statué précisément sur les moyens qu'elle mettra sûrement en œuvre.

# Priorisation du product backlog

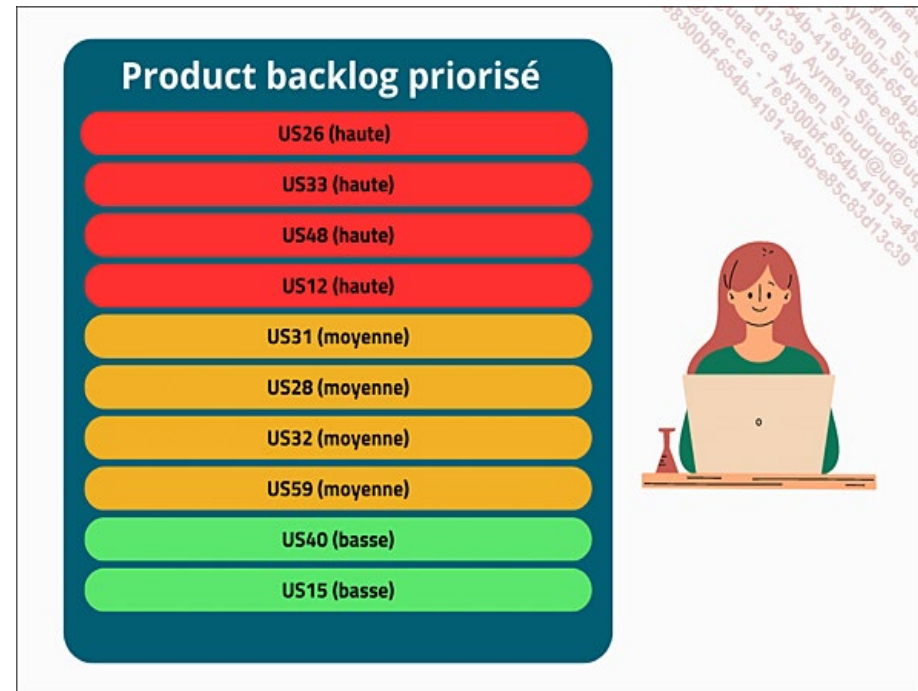
# Priorisation du PB

---

- ▶ Étape essentielle pour garantir que l'équipe se concentre sur les tâches qui offrent le plus de valeur à l'organisation et aux utilisateurs.
- ▶ La priorisation nécessite une compréhension approfondie des besoins des utilisateurs, du marché, ainsi que des objectifs et contraintes commerciales de l'organisation.

# Pourquoi prioriser le PB ?

- ▶ La priorisation aide à répondre à des questions clés :
  - ▶ quelle fonctionnalité ou quel aspect du produit devrait être développé en premier ?
  - ▶ Qu'est-ce qui apportera le plus de valeur à l'entreprise et aux utilisateurs ?



# Pourquoi prioriser le PB ?

---

- ▶ **Maximisation de la valeur**
  - ▶ Fournir le maximum de valeur dès le début du projet.
- ▶ **Gestion des ressources**
  - ▶ Utiliser les ressources de manière optimale pour développer les fonctionnalités les plus importantes.
- ▶ **Gestion des risques**
  - ▶ S'attaquer plus tôt aux parties les plus risquées du projet.
- ▶ **Satisfaction du client**
  - ▶ Augmenter la satisfaction du client et obtenir des retours plus rapidement.

# Approches de priorisation

---

- ▶ Une bonne priorisation du product backlog est une tâche collaborative qui nécessite l'implication de l'ensemble de l'équipe Scrum, en particulier du Product Owner.
- ▶ Une priorisation efficace exige une compréhension
  - ▶ des utilisateurs,
  - ▶ des objectifs de l'entreprise,
  - ▶ des contraintes techniques, et
  - ▶ une évaluation du retour sur investissement (ROI).

# Évaluation de la valeur de l'utilisateur

---

- ▶ Le premier critère de priorisation est la valeur qu'une fonctionnalité apporte aux utilisateurs finaux.
- ▶ Responsabilité du Product Owner
- ▶ Les personas utilisateurs peuvent être utilisées pour mieux comprendre les utilisateurs et leurs besoins.
  - ▶ Un persona utilisateur est une représentation semi-fictive d'un utilisateur idéal basée sur des données réelles et des hypothèses éclairées.
  - ▶ <https://agilemodeling.com/artifacts/personas.htm>



# Estimation de l'effort de développement

---

- ▶ Responsabilité de l'équipe de développement
- ▶ Exemple
  - ▶ Le Planning Poker est une technique ludique et collaborative utilisée dans les méthodes de développement agile pour estimer l'effort requis pour réaliser différentes tâches ou user stories.

# Analyse ROI

---

- ▶ En utilisant les informations sur la valeur de l'utilisateur et l'effort de développement, une analyse du ROI peut être réalisée pour chaque fonctionnalité.
- ▶ Cela permet de prioriser les fonctionnalités qui offrent le plus grand retour sur investissement.

# Prise en compte des dépendances et des risques

---

- ▶ Il est également important de prendre en compte les dépendances entre les fonctionnalités, ainsi que les risques associés.
- ▶ Par exemple, une fonctionnalité qui dépend d'une autre devrait être priorisée en conséquence.
- ▶ De même, les fonctionnalités qui présentent un risque élevé (par exemple, parce qu'elles sont techniquement complexes ou parce qu'elles sont nouvelles pour l'équipe) peuvent être priorisées plus haut pour s'attaquer à ces risques plus tôt.

# Revue et ajustement réguliers

---

- ▶ **Activité continue**
- ▶ **Le product backlog doit être revu et ajusté régulièrement en fonction des retours des utilisateurs, des changements dans les objectifs de l'entreprise, des contraintes techniques et d'autres facteurs.**

# Qu'est ce qui influe la priorisation du PB ?

---

- ▶ Valeur pour l'utilisateur
- ▶ Valeur pour
- ▶ Effort de développement
- ▶ Risques et incertitudes
- ▶ Dépendances
- ▶ Contraintes légales ou réglementaires

# Quelques méthodes de priorisation

---

- ▶ En évaluant chaque élément du backlog selon les critères RICE (Reach (Portée), Impact, Confidence (Confiance) et Effort (Effort)) , l'ordre de priorité pourrait être le suivant :
  - ▶ Fonctionnalité de suivi des pas
  - ▶ Système de badges de récompense
  - ▶ Intégration avec les réseaux sociaux
  - ▶ Intégration de la nutrition
  - ▶ Interface utilisateur multilingue

# Quelles méthodes utilisées 😊

---

- ▶ Peuvent être utilisées en combinaison pour fournir une vue équilibrée de la priorité.
- ▶ Le choix de la méthode dépend de la nature du projet, de l'équipe et de l'entreprise.

# Exemple méthodes de priorisation du PB

---

- ▶ Supposons que le PB contienne les items suivants :
  - ▶ Fonctionnalité de suivi des pas
  - ▶ Système de badges de récompense
  - ▶ Intégration avec les réseaux sociaux pour partager les progrès
  - ▶ Intégration de la nutrition pour suivre les repas
  - ▶ Interface utilisateur multilingue



# Exemple méthodes de priorisation du PB

## MOSCOW

---

- ▶ **Must have (Doit avoir) : fonctionnalité de suivi des pas.**
  - ▶ C'est une fonctionnalité essentielle pour toute application de fitness, indispensable pour répondre aux besoins de base des utilisateurs.
- ▶ **Should have (Devrait avoir) : système de badges de récompense.**
  - ▶ Bien que non essentiel, ce système est important car il peut encourager l'engagement des utilisateurs et améliorer l'expérience utilisateur.
- ▶ **Could have (Pourrait avoir) : intégration avec les réseaux sociaux et intégration de la nutrition.**
  - ▶ Ces fonctionnalités sont désirables mais non critiques. Elles pourraient améliorer l'application mais ne sont pas prioritaires par rapport aux catégories 'Must have' et 'Should have'.
- ▶ **Won't have (N'aura pas pour l'instant) : interface utilisateur multilingue.**
  - ▶ Cette fonctionnalité n'est pas nécessaire dans l'immédiat et peut être envisagée pour une version ultérieure. C'est une décision stratégique de reporter ce développement à plus tard.

# Exemple méthodes de priorisation du PB

## Matrice d'Eisenhowe

---

- ▶ **Urgent et important : fonctionnalité de suivi des pas.**
  - ▶ Cette fonctionnalité est essentielle pour une application de fitness et doit être mise en œuvre rapidement pour répondre aux besoins immédiats des utilisateurs.
- ▶ **Non urgent mais important : système de badges de récompense.**
  - ▶ Bien qu'il ne soit pas urgent, ce système est important pour l'engagement et la fidélisation à long terme des utilisateurs. Intégration avec les réseaux sociaux. Cette intégration peut améliorer la visibilité et l'engagement de l'application, mais elle n'est pas urgente.
- ▶ **Urgent mais non important : cette catégorie peut inclure des tâches qui exigent une attention immédiate mais qui n'apportent pas une valeur significative à long terme.**
  - ▶ Par exemple, répondre à des demandes ponctuelles des utilisateurs ou des partenaires qui ne sont pas stratégiques pour le développement du produit.
- ▶ **Non urgent et non important : intégration de la nutrition.**
  - ▶ Bien que cette fonctionnalité puisse ajouter de la valeur, elle n'est ni urgente ni essentielle pour le fonctionnement de base de l'application.
- ▶ **Interface utilisateur multilingue.**
  - ▶ Comme elle n'est pas nécessaire immédiatement, cette fonctionnalité peut être classée comme non urgente et non importante dans le contexte actuel.

# Exemple méthodes de priorisation du PB

## Méthode RICE

---

- ▶ (Reach, Impact, Confidence, Effort) : la méthode RICE consiste à évaluer chaque élément du backlog en fonction de sa portée (combien d'utilisateurs seront touchés), de son impact (quel sera l'effet sur les utilisateurs touchés), de la confiance (quelle est la certitude que l'élément aura l'impact prévu) et de l'effort (combien de travail sera nécessaire pour réaliser l'élément).
- ▶ En évaluant chaque élément du backlog selon les critères RICE, l'ordre de priorité pourrait être le suivant :
  - ▶ Fonctionnalité de suivi des pas
  - ▶ Système de badges de récompense
  - ▶ Intégration avec les réseaux sociaux
  - ▶ Intégration de la nutrition
  - ▶ Interface utilisateur multilingue

# Les thèmes

---

- ▶ Les thèmes sont des groupes d'éléments du backlog qui partagent un objectif commun.
- ▶ Exemple
  - ▶ "Motivation de l'utilisateur", regroupant des éléments tels que le suivi des progrès, le système de badges de récompense, les défis quotidiens, etc.

# Priorisation par thèmes

---

- ▶ La priorisation par thèmes permet de se concentrer sur la réalisation d'objectifs spécifiques plutôt que sur des fonctionnalités individuelles.

# Le sondage des thèmes (theme screening)

---

- ▶ Utile quand de nombreux thèmes sont identifiées et doit décider lesquels sont les plus importants à travailler en premier.
  
- ▶ **Fonctionnement**
  - ▶ Chaque membre de l'équipe attribue une note à chaque thème, généralement sur une échelle de 1 à 5, en fonction de son importance perçue pour l'atteinte des objectifs du projet.
  - ▶ Ensuite, les scores de chaque thème sont totalisés pour obtenir un score global. Les thèmes sont ensuite priorisés en fonction de ces scores.
  
- ▶ **Exemple**
  - ▶ Si le thème "Motivation de l'utilisateur" obtient le score le plus élevé, cela signifierait que l'équipe estime que ce thème est le plus important à travailler en premier.

# La mesure des thèmes (theme scoring)

---

- ▶ Technique est plus objective et quantifiable que le theme screening, et prend en compte des critères spécifiques pour attribuer un score à chaque thème.
- ▶ Fonctionnement
  - ▶ Définir les critères d'évaluation
  - ▶ Attribuer une pondération à chaque critère
  - ▶ Évaluer chaque thème en fonction des critères
  - ▶ Calculer le score total pour chaque thème
- ▶ Exemple
  - ▶ Critères : ROI (pondération 2) / Complexité technique (pondération 1)
  - ▶ Score : "Motivation de l'utilisateur (5x2) + (3x1) = 13
  - ▶ Score : "Suivi de la nutrition" serait (4x2) + (2x1) = 10.

# La priorisation des thèmes par l'utilisation de poids relatifs

---

- ▶ Utile lorsque l'on cherche à comparer le "poids" ou l'importance relative de différents thèmes.
- ▶ **Fonctionnement**
  - ▶ Identification des critères
  - ▶ Attribution de poids à chaque critère (importance relative)
  - ▶ Évaluation de chaque thème
  - ▶ Calcul du poids relatif de chaque thème
- ▶ **Exemple**
  - ▶ Critères : Bénéfice commercial (poids 2) / Coût développement (poids 1)
  - ▶ Score : "Intégration de la réalité augmentée "  $(7 \times 2) + (3 \times 1) = 17$



# Modèle de Kano

---

- ▶ Le modèle de Kano classe les exigences des produits en cinq catégories :
  - ▶ Exigences de base (Must-be)
    - ▶ Une exigence de base peut être une interface conviviale.
  - ▶ Exigences de performance (One-dimensional)
    - ▶ Un nombre plus important de programmes d'entraînement disponibles peut augmenter la satisfaction du client.
  - ▶ Exigences excitantes (Attractive)
    - ▶ L'intégration d'un coach virtuel pour fournir des commentaires en temps réel pourrait être une exigence excitante.
  - ▶ Exigences indifférentes (Indifferent)
  - ▶ Exigences inversées (Reverse)
    - ▶ Un nombre excessif de notifications pourrait nuire à l'expérience utilisateur.

# Modèle de Kano

## Fonctionnement

---

- ▶ Pour prioriser les items du backlog en utilisant le modèle de Kano, il faut d'abord identifier dans quelle catégorie chaque item se situe.
- ▶ Cela peut être réalisé en menant des interviews avec les clients ou en utilisant d'autres méthodes de recherche utilisateur.
- ▶ Ensuite, nous donnons la priorité aux exigences de base, puis aux exigences de performance et enfin aux exigences excitantes.

# La méthode MoSCoW pour la priorisation

---

- ▶ La méthode MoSCoW est une technique de priorisation qui permet de classer les exigences du product backlog selon leur importance.
  - ▶ M pour "Must have" (doit avoir) : ces éléments sont impératifs pour le projet. Ils sont non négociables et le produit ne peut pas être lancé sans eux.
  - ▶ S pour "Should have" (devrait avoir) : ces éléments sont importants mais non critiques. Si nous ne pouvons pas les inclure dans le produit actuel, ils seront prioritaires pour les versions futures.
  - ▶ C pour "Could have" (pourrait avoir) : ces éléments sont souhaitables mais non essentiels. Ils sont généralement inclus si le temps et les ressources le permettent.
  - ▶ W pour "Won't have" (n'aura pas) : ces éléments sont les moins prioritaires. Ils ne seront pas inclus dans la version actuelle, mais pourront être envisagés pour des versions futures.

# La méthode MoSCoW pour la priorisation

## Fonctionnement

---

- ▶ Permet de hiérarchiser clairement les exigences en fonction de leur importance et aide à déterminer l'ordre dans lequel les exigences doivent être traitées.
- ▶ Utile dans le contexte du développement de produits Scrum car elle facilite la planification des sprints en mettant l'accent sur les fonctionnalités qui apportent le plus de valeur à l'utilisateur final.
- ▶ Peut être combiné à d'autres techniques
  - ▶ Chaque équipe Scrum doit choisir la technique qui convient le mieux à son contexte spécifique.

# Exemple Moscow

---

## ► PB

- Mise en place d'une fonctionnalité de suivi de la fréquence cardiaque en temps réel.
- Ajout d'un système de badges de récompense pour motiver les utilisateurs à atteindre leurs objectifs.
- Intégration d'un système de recommandations personnalisées pour les exercices.
- Ajout d'une fonctionnalité permettant aux utilisateurs de partager leurs progrès sur les réseaux sociaux.
- Mise en place d'une fonctionnalité de rappels pour inciter les utilisateurs à faire de l'exercice régulièrement.

# Exemple Moscow

## Priorisation

---

- ▶ **Must have (doit avoir) :**
  - ▶ Mise en place d'une fonctionnalité de suivi de la fréquence cardiaque en temps réel.
- ▶ **Should have (devrait avoir) :**
  - ▶ Ajout d'un système de badges de récompense pour motiver les utilisateurs à atteindre leurs objectifs.
- ▶ **Could have (pourrait avoir)**
  - ▶ Intégration d'un système de recommandations personnalisées pour les exercices et ajout d'une fonctionnalité de rappels pour inciter les utilisateurs à faire de l'exercice régulièrement.
- ▶ **Won't have (n'aura pas)**
  - ▶ Ajout d'une fonctionnalité permettant aux utilisateurs de partager leurs progrès sur les réseaux sociaux.

# La méthode WSJF

Weighted Shortest Job First (travail le plus court ponderé en premier)

---

- ▶ Technique de priorisation dérivée de la gestion Lean des produits.
  - ▶ Utilisée dans le cadre du framework SAFe (Scaled Agile Framework).
- ▶ Permet de maximiser la valeur économique délivrée en minimisant le temps de réalisation.
- ▶ Cette méthode prend en compte deux aspects principaux :
  - ▶ La valeur du travail : c'est l'importance économique de la tâche pour l'organisation. Elle peut être évaluée en termes de revenus potentiels, de satisfaction client, d'impact sur la marque, ou de tout autre critère pertinent.
  - ▶ La taille du travail : c'est la quantité de travail nécessaire pour réaliser la tâche. Elle peut être évaluée en termes de temps, de ressources ou de complexité.

# Fonctionnement de la méthode WSJF

---

- ▶ La méthode WSJF attribue un score à chaque élément du backlog en divisant sa valeur par sa taille.
- ▶ Les éléments avec les scores les plus élevés sont ceux qui offrent le meilleur retour sur investissement et sont donc prioritaires.
- ▶ Formule de WSJF :
  - ▶  $WSJF = \text{Valeur du travail} / \text{Taille du travail}$
- ▶ Le principe est simple : plus la valeur est grande et la taille petite, plus le score WSJF est élevé et plus l'élément doit être priorisé.



# Exemple WSJF

---

- ▶ Amélioration de l'interface utilisateur
  - ▶ Valeur à 8 (sur une échelle de 1 à 10) et sa taille à 5 (en termes de complexité et de temps nécessaire).
- ▶ Intégration des réseaux sociaux : valeur à 7 et taille à 3.
- ▶ Ajout d'un chat en direct pour le support : 6 et 8
- ▶ Intégration de nouvelles routines d'entraînement : 9 et 4
  
- ▶ Ce qui donne
  - ▶ (3) Amélioration de l'interface utilisateur :  $8/5 = 1.6$
  - ▶ (1) Intégration des réseaux sociaux :  $7/3 = 2.33$
  - ▶ (4) Ajout d'un chat en direct pour le support :  $6/8 = 0.75$
  - ▶ (2) Intégration de nouvelles routines d'entraînement :  $9/4 = 2.25$