

La planification et l'estimation dans Scrum

Aymen Sioud
Département Informatique et Mathématique
Courriel : aymen.sioud@uqac.ca

Tables des matières

- ▶ La planification
- ▶ Outils d'estimation
 - ▶ T-shirt
 - ▶ Story point
 - ▶ Vitesse
- ▶ La planification Poker
- ▶ Planification du Release

La planification traditionnelle

Un constat d'échec

- ▶ La planification traditionnelle repose sur une approche linéaire et séquentielle de la gestion de projet.
 - ▶ Pas de retour en arrière ou d'itération.
- ▶ Cependant, cette approche présente plusieurs défis qui peuvent mener à l'échec :
 - ▶ Rigidité
 - ▶ Difficile de modifier le plan établi sans causer de perturbations majeures.
 - ▶ Manque de feedback
 - ▶ Produit fini n'est généralement pas vu avant la toute fin du projet.
 - ▶ Estimation imprécise
 - ▶ On suppose que tous les aspects du projet peuvent être entièrement compris et estimés dès le début.
 - ▶ Mauvaise gestion des risques
 - ▶ Problèmes ne sont souvent détectés que tardivement, lorsqu'ils sont plus coûteux et plus complexes à résoudre.
 - ▶ Démotivation de l'équipe
 - ▶ On laisse peu de place à la créativité et à l'autonomie de l'équipe, ce qui peut conduire à un manque de motivation et d'engagement.

La planification agile

Scrum

- ▶ L'accent est mis sur la flexibilité, l'adaptabilité et la satisfaction du client plutôt que sur le respect strict d'un plan prédéterminé.
- ▶ Cette perspective de planification est basée sur plusieurs principes clés :
 - ▶ Planification adaptative
 - ▶ Processus continu et itératif
 - ▶ Priorisation
 - ▶ Les éléments les plus importants ou ceux qui apportent le plus de valeur sont abordés en premier.
 - ▶ Estimation relative
 - ▶ Estimer le travail de manière relative
 - ▶ Planification de la capacité
 - ▶ Tenir compte du volume de travail de l'équipe
 - ▶ Révision et amélioration
 - ▶ Rétrospective pour améliorer

Importance de la planification et l'estimation

- ▶ Dans la méthodologie Scrum, la planification et l'estimation sont des pratiques incontournables pour garantir la bonne marche du projet et la satisfaction des parties prenantes.
- ▶ Ces pratiques permettent d'établir une vision à moyen et long terme du projet, tout en restant flexible pour s'adapter aux changements.
- ▶ L'estimation en Scrum n'est pas seulement une affaire de délais.
- ▶ Il s'agit également d'évaluer la complexité et l'effort requis pour chaque élément du backlog du produit.
- ▶ Cela aide l'équipe à comprendre ce qu'il faut pour réaliser chaque fonctionnalité, et cela aide le Product Owner à prioriser efficacement les éléments du backlog.

Importance de la planification et l'estimation

- ▶ La planification, est essentielle pour définir le cap et fixer des objectifs réalisables pour l'équipe.
- ▶ En Scrum, cela se produit à plusieurs niveaux
 - ▶ la planification du sprint, où l'équipe s'accorde sur ce qui sera réalisé lors du prochain sprint, et
 - ▶ la planification de la release, où l'équipe et le Product Owner discutent de l'orientation générale du produit et des objectifs à atteindre.
- ▶ Pas des tâches faciles, et il est important de noter qu'elles nécessitent une bonne dose de collaboration, de communication et de transparence pour être efficaces.
- ▶ Tâches qui requièrent également de l'expérience, car elles impliquent souvent de l'incertitude et nécessitent une compréhension des capacités de l'équipe et des exigences du produit.

Estimer relative de l'effort

- ▶ Imaginez que vous deviez courir en équipe sur une distance de quelques kilomètres, le plus rapidement possible et répondre à cette « Combien de temps vous prendra cette course ? »
- ▶ Réponses différentes
- ▶ Il y a bien un point sur lequel tout le monde est d'accord, c'est la distance à parcourir.
 - ▶ Donc si nous attribuons une valeur arbitraire en points pour l'effort nécessaire sur une course de 5 km, nous pouvons alors aisément estimer un effort double pour une course de 10 km

Facteurs à prendre en compte pour l'estimation de l'effort

- ▶ La quantité de travail
- ▶ La complexité
- ▶ Le risque à faire
- ▶ La maîtrise en matière de compétences

Exemple

(Mike Cohn)

- ▶ Imaginons que nous avons une équipe de deux personnes composée d'un chirurgien du cerveau et d'un enfant.
- ▶ Nous avons deux stories à réaliser : une opération du cerveau courante (histoire utilisateur 1) et le collage de 1 000 timbres sur 1 000 enveloppes (histoire utilisateur 2).
- ▶ Nous supposons ici que les deux activités prennent autant de temps.
- ▶ Si vous estimez ces deux histoires en fonction de leur complexité, l'opération du cerveau aura une estimation beaucoup plus élevée que l'affranchissement des enveloppes.
 - ▶ Pourtant, si votre but est de planifier et de savoir quand l'équipe aura terminé, étant donné que le temps de réalisation est quasiment le même, l'estimation devrait être la même.
- ▶ En fait, on peut dire que si la complexité varie, l'effort à fournir dans les deux cas est comparable avec des compétences spécifiques différentes (l'un nécessite une grande concentration et une grande dextérité, alors que l'autre nécessite surtout de la patience).



Outils d'estimation



Tailles de t-shirt

Notre exemple

- ▶ Notre product backlog contient les éléments suivants :
 - ▶ ajouter une nouvelle fonctionnalité de suivi d'entraînement ;
 - ▶ améliorer l'interface utilisateur pour rendre le suivi des progrès plus clair ;
 - ▶ corriger un bug mineur dans l'interface utilisateur ;
 - ▶ intégrer une nouvelle API pour le partage des résultats sur les réseaux sociaux.

Déroulement du Tailles t-shirt

1. Le Product Owner, présente chaque élément du backlog en détail et répond à toutes les questions que l'équipe pourrait poser.
 - ▶ C'est un moment clé pour clarifier les attentes et éviter toute confusion future.
2. Le Scrum Master, joue le rôle de facilitateur, veillant à ce que la réunion reste centrée sur l'objectif et à ce que toutes les voix soient entendues.
3. Les développeurs examinent chaque élément, discutent de la complexité de l'implémentation, des dépendances potentielles et des risques associés.
 - ▶ À partir de ces discussions, l'équipe attribue une taille de t-shirt à chaque élément du backlog.

Après discussion et échanges

- ▶ Fonctionnalité "Ajouter une nouvelle fonctionnalité de suivi d'entraînement" - Estimation : **XL**.
 - ▶ C'est un gros travail qui implique de modifier l'interface utilisateur, d'ajouter une nouvelle logique côté serveur et de tester soigneusement pour éviter les bugs.
- ▶ Fonctionnalité "Améliorer l'interface utilisateur pour rendre le suivi des progrès plus clair" - Estimation : **M**.
 - ▶ Cela nécessite des changements dans l'interface utilisateur, mais la logique de l'application ne devrait pas être affectée.
- ▶ Fonctionnalité "Corriger un bug mineur dans l'interface utilisateur" - Estimation : **S**.
 - ▶ C'est une tâche simple qui ne devrait prendre que peu de temps.
- ▶ Fonctionnalité "Intégrer une nouvelle API pour le partage des résultats sur les réseaux sociaux" - Estimation : **L**.
 - ▶ C'est une tâche complexe qui nécessite de comprendre une nouvelle API et d'intégrer cette fonctionnalité dans l'application.

Et après ?

- ▶ Les estimations ne sont pas fixes et peuvent être revues et ajustées au fur et à mesure que l'équipe acquiert plus d'informations ou si des changements se produisent.
- ▶ L'important est d'impliquer toute l'équipe dans le processus d'estimation pour obtenir un consensus et une compréhension claire de la taille du travail à venir.
- ▶ Les avis divergents sont essentiels 😊





Outils d'estimation

Les points de story



Les points de story

Les points de story

- ▶ Les points de story représentent une mesure de la complexité relative d'une user story, prenant en compte les efforts, le risque et la complexité impliqués.
- ▶ Contrairement à d'autres méthodes d'estimation, les points de story ne sont pas directement liés au temps.
 - ▶ Il s'agit plutôt d'un moyen pour l'équipe de comparer les user stories les unes aux autres et d'estimer la quantité de travail que chaque histoire nécessite.

Fonctionnement des points de story

1. Cette méthode repose sur l'idée que l'estimation du temps peut être imprécise et sujette à des erreurs, tandis que l'estimation de la complexité relative peut être plus précise et utile pour la planification.
2. L'équipe attribue à chaque user story un nombre de points de story sur une échelle prédéfinie.
 - ▶ Cette échelle est souvent la séquence de Fibonacci (1, 2, 3, 5, 8, 13, 21, etc.), qui permet d'illustrer l'incertitude croissante avec des user stories de plus grande taille.

Fonctionnement des points de story

Le PO en rôle central

3. Les points de story sont déterminés lors d'une réunion d'estimation de l'équipe de développement.
4. Le Product Owner présente chaque user story et répond aux questions de l'équipe.
5. Chaque membre de l'équipe propose une estimation en points de story et, après une discussion, une estimation finale est convenue.

Les points de story

Unique à chaque équipe

- ▶ En utilisant cette méthode, l'équipe peut estimer combien de points de story elle peut livrer dans un sprint (sa "vélocité") et utiliser cette information pour la planification future.
- ▶ **Il est important de noter que les points de story sont spécifiques à chaque équipe et ne doivent pas être comparés entre différentes équipes.**

Pour quoi la suite de Fibonacci ?

Ou encore (0, 0.5, 1, 2, 3, 5, 8, 13, 20, 40, 100) ?

- ▶ Reflète l'incertitude croissante avec la taille des tâches.
- ▶ Montre que l'écart entre les valeurs augmente pour les grandes tâches, absorbant ainsi l'incertitude liée à leur estimation.
- ▶ Reconnaît également que l'estimation du temps ou de l'effort n'est pas précise, surtout pour les grandes tâches, où il est plus difficile de faire des prévisions exactes, comme le montre l'augmentation des écarts entre les nombres consécutifs de la suite.
- ▶ Aide à éviter les faux débats sur des détails mineurs de l'estimation 😊.
 - ▶ Si l'on utilise une échelle continue, les membres de l'équipe pourraient débattre de la question de savoir si une user story est un "7" ou un "8". En utilisant la suite de Fibonacci, le choix est limité à "5" ou "8", ce qui simplifie la discussion.

Exemple points de story

- Nous avons une table de référence des user stories déjà estimées pour notre application.

User Story	Story Point
En tant qu'utilisateur, je veux pouvoir me connecter via mon compte Facebook afin de faciliter l'accès à l'application.	3
En tant qu'utilisateur, je veux pouvoir suivre mes progrès hebdomadaires pour savoir si j'atteins mes objectifs.	8
En tant qu'utilisateur, je veux recevoir des recommandations personnalisées pour mes exercices en fonction de mon historique de fitness.	13
En tant qu'utilisateur, je souhaite pouvoir ajouter des photos de mes repas pour tenir un journal alimentaire.	5

Exemple points de story

Autres estimations

- ▶ En tant qu'utilisateur, je veux pouvoir synchroniser l'application avec mon Apple Watch pour suivre mes activités.
- ▶ En tant qu'utilisateur, je veux avoir accès à des routines d'exercices personnalisées en fonction de mon objectif de fitness (perte de poids, gain musculaire, etc.).
- ▶ En tant qu'utilisateur, je souhaite recevoir des notifications de rappel pour boire de l'eau tout au long de la journée.

User Story	Story Point
En tant qu'utilisateur, je veux pouvoir me connecter via mon compte Facebook afin de faciliter l'accès à l'application.	3
En tant qu'utilisateur, je veux pouvoir suivre mes progrès hebdomadaires pour savoir si j'atteins mes objectifs.	8
En tant qu'utilisateur, je veux recevoir des recommandations personnalisées pour mes exercices en fonction de mon historique de fitness.	13
En tant qu'utilisateur, je souhaite pouvoir ajouter des photos de mes repas pour tenir un journal alimentaire.	5

Exemple points de story

Nouvelles estimations

- ▶ Pour la synchronisation avec l'Apple Watch, l'équipe pourrait estimer que cette user story est plus complexe que celle de suivre les progrès hebdomadaires, mais moins complexe que celle de recevoir des recommandations personnalisées d'exercices.
 - ▶ Ils pourraient donc lui attribuer un score de **8** points de story.
- ▶ Pour les routines d'exercices personnalisées, l'équipe pourrait considérer cette user story comme plus complexe que la user story de recommandations personnalisées d'exercices, en raison de la diversité des routines qui doivent être créées.
 - ▶ Ils pourraient donc lui attribuer un score de **13** points de story.
- ▶ Pour les notifications de rappel pour boire de l'eau, l'équipe pourrait estimer que cette user story est moins complexe que celle d'ajouter des photos de repas au journal alimentaire.
 - ▶ Ils pourraient donc lui attribuer un score de **3** points de story.

Les points de story

Estimations relatives

- ▶ Estimations réalisées en équipe se basant sur la complexité relative et l'effort de travail perçu, et non sur un temps spécifique ou une durée de travail.
- ▶ Ce n'est pas une unité de mesure universelle et qu'ils ne peuvent pas être comparés entre différentes équipes.
- ▶ Chaque équipe définira son propre tableau de référence, et l'estimation en points de story sera spécifique à cette équipe.
- ▶ Il serait incorrect de comparer directement les points de story entre deux équipes ou d'essayer d'établir une moyenne ou une somme des points de story sur plusieurs équipes.

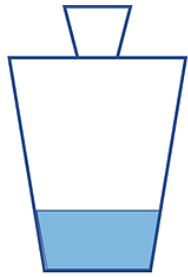
Étalonner

- ▶ Le Story Point n'est ni du jour/homme, ni du kilo-dollars.
- ▶ Pour une équipe, il faut par conséquent définir un étalon et cet étalon sera utilisé comme référence pour estimer en relatif les choses à faire et s'accorder sur leur taille.

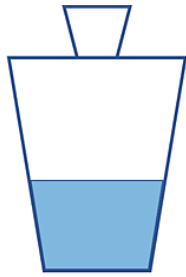
Où est l'étalon ?



Trouver l'étalon



Quantité de référence

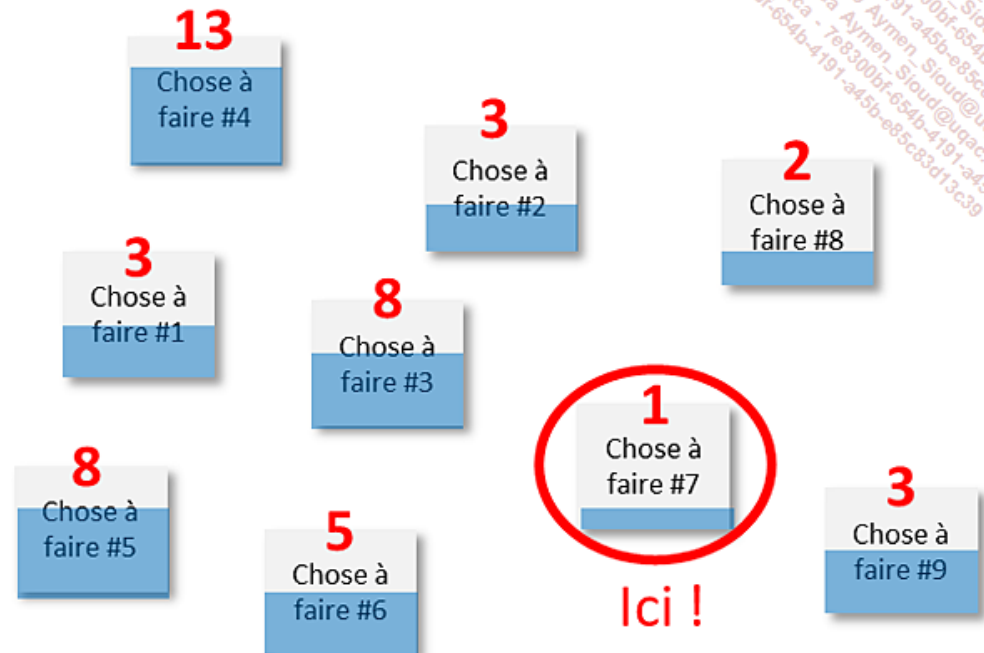



Quantité supérieure
à la quantité de référence



Quantité très supérieure
à la quantité de référence

Où est l'étalon ?





Outils d'estimation

Les points de story ou heure-homme



Un débat ?

- ▶ Faut-il estimer le travail à faire en termes de points de story ou d'heure-homme ?
- ▶ Dépend de
 - ▶ la nature du projet,
 - ▶ la composition de l'équipe et
 - ▶ les exigences du client.

Une comparaison

Points de story

- ▶ Mesure abstraite prenant en compte la complexité du travail à faire, le risque associé et les incertitudes.
- ▶ Peut aider à absorber les variations naturelles de la productivité et à éviter les problèmes liés aux estimations basées sur le temps, comme le padding (ajout de temps supplémentaire à une estimation par précaution).
- ▶ Facilitent l'adaptabilité, qui est l'un des principaux avantages de l'agilité.

Une comparaison

Heure-homme

- ▶ Quantifier le temps nécessaire pour réaliser une tâche.
- ▶ Familière et plus précise.
- ▶ Peut aussi conduire à des erreurs d'estimation, en particulier pour les tâches complexes ou celles qui comportent beaucoup d'inconnues.

Comparaison

Verdict

- ▶ Dans le contexte agile, il est généralement recommandé d'utiliser
 - ▶ les points de story pour les estimations à long terme (par exemple, pour les releases ou les grandes epics)
 - ▶ les heures-hommes pour les estimations à court terme (par exemple, pour les sprints).
- ▶ Le choix doit être guidé par ce qui fonctionne le mieux pour l'équipe et le projet en question.
- ▶ Il est essentiel d'adopter une approche flexible et d'être prêt à ajuster la méthode d'estimation si nécessaire.



La vitesse



La vélocité

- ▶ Concept clé dans le monde agile, plus précisément dans la méthodologie Scrum.
- ▶ Elle désigne la quantité de travail qu'une équipe est capable de réaliser pendant un sprint, généralement mesurée en points de story.
- ▶ La vélocité se calcule en additionnant les points de story de toutes les user stories que l'équipe a terminées au cours d'un sprint.
- ▶ Exemple
 - ▶ Si une équipe a terminé 5 user stories de 3 points chacune pendant un sprint, la vélocité de l'équipe pour ce sprint serait de 15.

La vélocité

Quelques caractéristiques

- ▶ Ce n'est pas un indicateur de performance, mais plutôt un outil de planification.
 - ▶ Elle aide l'équipe à comprendre combien de travail elle peut réalistiquement accomplir lors des prochains sprints, basé sur ce qu'elle a pu accomplir par le passé.
- ▶ Lorsque l'équipe a terminé quelques sprints, on peut calculer une vélocité moyenne.
 - ▶ Cette moyenne donne une indication sur le volume de travail que l'équipe est en mesure de réaliser durant un sprint, et peut être utilisée pour une planification plus précise des sprints à venir.

La vélocité

Quelques caractéristiques

- ▶ La vélocité est spécifique à chaque équipe.
 - ▶ Deux équipes travaillant sur le même projet peuvent avoir des vélocités différentes en raison de facteurs tels que la composition de l'équipe, l'expérience des membres et le contexte de travail. C'est pourquoi il n'est pas conseillé de comparer la vélocité de deux équipes différentes.
- ▶ Ce n'est pas une fin en soi.
 - ▶ Elle doit être utilisée comme un outil pour aider l'équipe à s'améliorer et à planifier de manière plus efficace, et non comme un moyen de pression ou de comparaison.

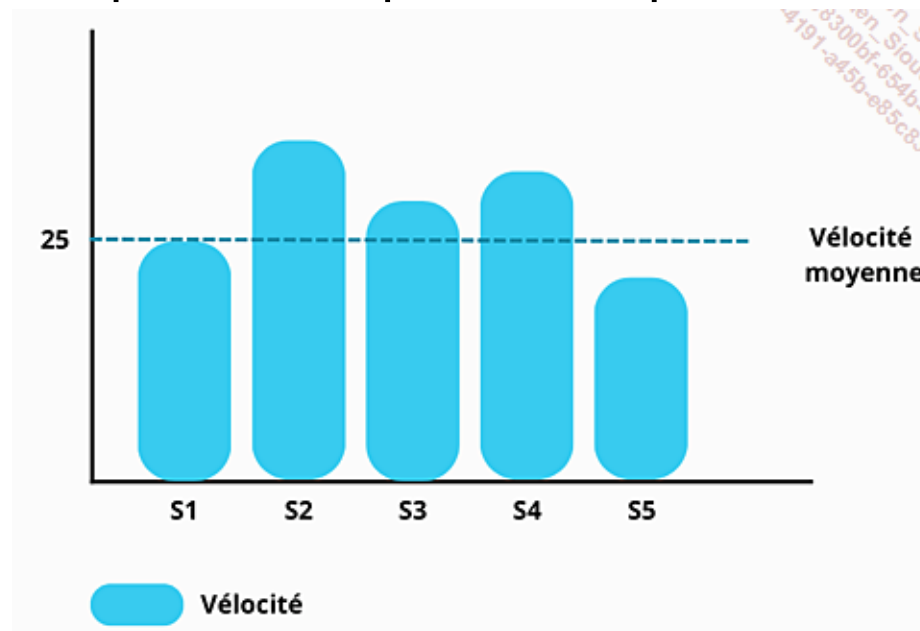
La vélocité

Le suivi

- ▶ La vélocité d'une équipe peut être suivie grâce à plusieurs types de graphiques.
- ▶ Les deux plus courants sont :
 - ▶ le graphique de vélocité et
 - ▶ le burndown chart.

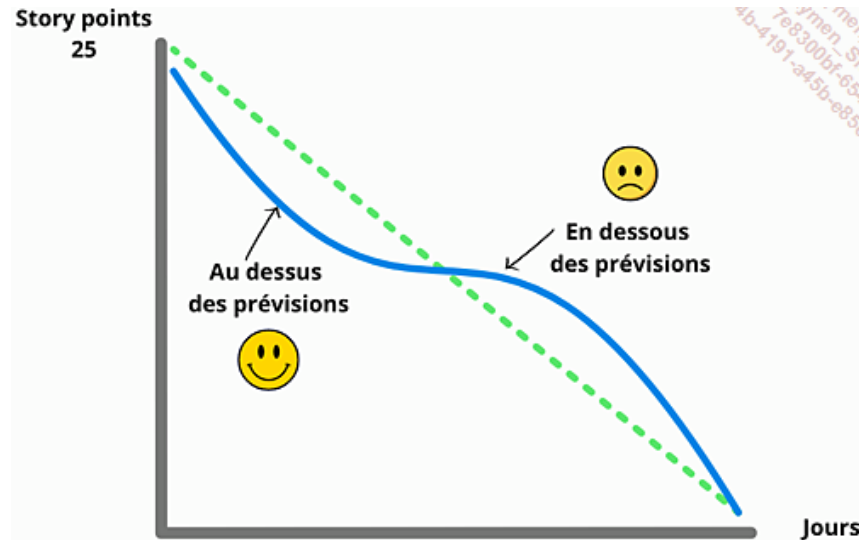
Le graphique de vélocité

- ▶ Graphique à barres qui affiche la vélocité de l'équipe pour chaque sprint.
- ▶ Les barres peuvent être divisées en deux couleurs pour distinguer les points de story terminés de ceux qui étaient prévus mais n'ont pas été accomplis.
- ▶ Cela permet de visualiser la stabilité de la vélocité d'une équipe sur plusieurs sprints et peut aider à prévoir la capacité future de l'équipe.



Le Burndown Chart

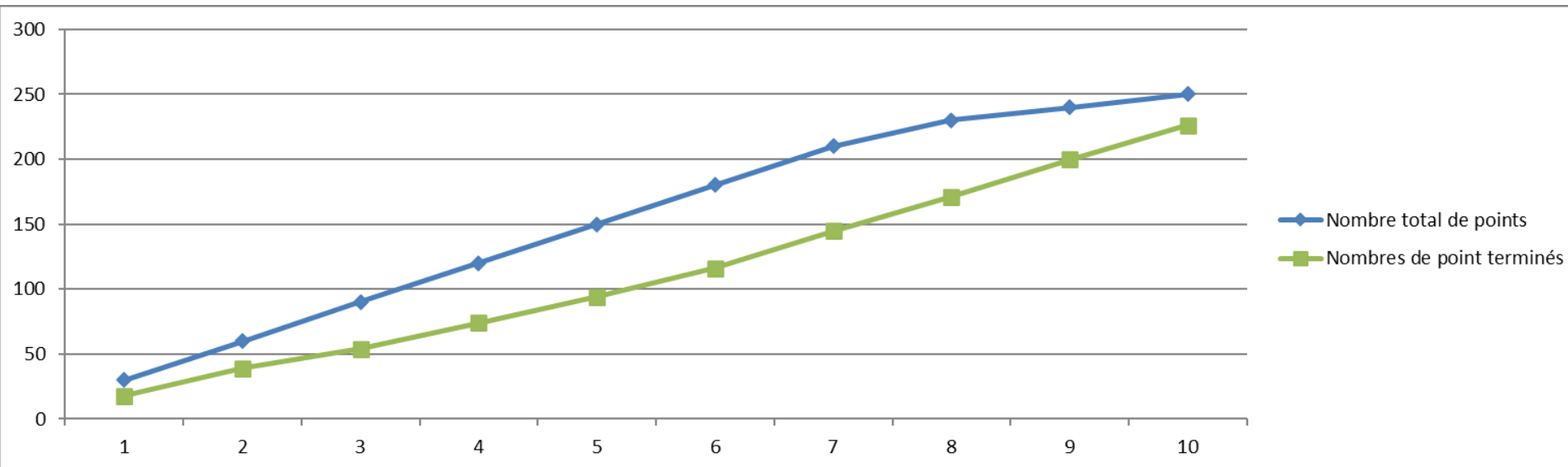
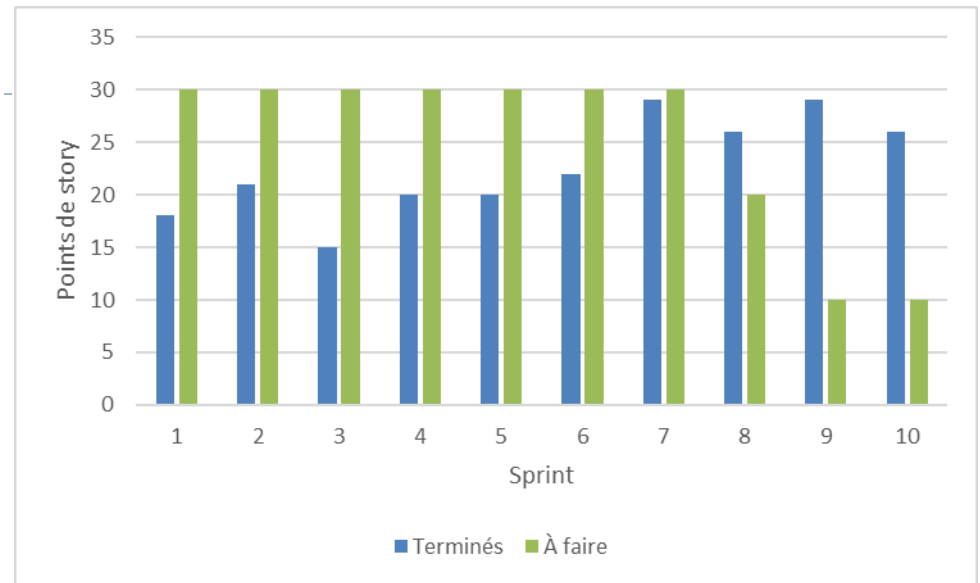
- ▶ Illustre la quantité de travail restante au fur et à mesure que le temps avance pendant un sprint.
- ▶ Le but est de voir la ligne descendre vers zéro d'ici la fin du sprint. Si la ligne est au-dessus de la ligne idéale (diagonale du graphique), cela signifie que l'équipe est en retard par rapport à son plan initial.



Mise à jour des graphiques

- ▶ Mis à jour quotidiennement pendant les daily scrums ou les réunions de suivi, permettant ainsi à l'équipe de comprendre son avancement et d'ajuster son plan de travail si nécessaire.
- ▶ Ils contribuent à la transparence, à la communication et à l'amélioration continue, qui sont des éléments clés des méthodologies agiles.
- ▶ Aussi burn up et ajout de la tendance

Autres graphiques



Initialisation de la vélocité

Mise en place d'un projet test

I. Mise en place d'un projet test

- ▶ Travailler sur un projet réel, mais de moindre envergure, qui permettra de mesurer la capacité de travail de l'équipe.
- ▶ Estimer les stories, travailler dessus et les compléter comme dans un véritable projet.
- ▶ Les points de story complétés au cours de ce projet test vont aider à déterminer la vélocité initiale de l'équipe.
- ▶ Le projet test doit être représentatif du travail que l'équipe va effectuer par la suite.
 - ▶ Il est donc crucial de choisir des user stories qui reflètent le type de travail que l'équipe sera amenée à faire.

Initialisation de la vélocité

Choisir sur la base du sentiment

2. Choisir sur la base du sentiment

- ▶ Demander aux membres de l'équipe d'estimer, sur la base de leur intuition et de leur expérience, combien de points de story ils pensent être capables d'accomplir lors d'un sprint.
- ▶ L'idée est de créer un environnement où l'équipe peut s'exprimer librement, sans crainte de jugement ou de répercussions.
- ▶ Utiliser la moyenne des estimations
- ▶ Bonne approche lorsque l'équipe est très nouvelle et n'a pas encore eu l'occasion de travailler ensemble.
- ▶ Réévalue la vélocité régulièrement

Initialisation de la vélocité

Estimation de la vélocité à partir de l'historique

3. Estimation de la vélocité à partir de l'historique

- ▶ Pertinente pour les équipes qui ont déjà travaillé ensemble sur plusieurs sprints.
- ▶ Utiliser une moyenne des de points de story l'équipe a été capable de terminer lors des précédents sprints.
 - ▶ Par exemple, si l'équipe a terminé 10, 12, et 8 points de story lors des trois derniers sprints, la vélocité moyenne serait de 10 points de story ($(10+12+8)/3$).
- ▶ Pas applicable pour une toute nouvelle équipe sans historique de travail ensemble.
- ▶ Considérer l'historique des sprints avec prudence.
 - ▶ Vacances, des maladies ou d'autres événements
 - ▶ Ajuster la vélocité en continu en fonction des performances récentes.

Qui effectue les estimations ?

- ▶ Responsabilité collective de l'équipe de développement Scrum.
 - ▶ C'est l'équipe entière qui possède la connaissance détaillée nécessaire pour réaliser une estimation précise du travail à faire.
- ▶ Chaque membre de l'équipe a un rôle à jouer.
 - ▶ Les développeurs, par exemple, peuvent estimer le temps qu'il leur faudrait pour coder une fonctionnalité.
 - ▶ Les testeurs, quant à eux, peuvent estimer le temps nécessaire pour tester cette fonctionnalité.
 - ▶ Les concepteurs peuvent fournir des estimations sur le temps nécessaire pour concevoir l'interface utilisateur.
- ▶ Le PO peut participer à ces réunions d'estimation, mais son rôle n'est pas d'influencer ou de diriger l'estimation, mais plutôt de clarifier les exigences et de répondre aux questions.
- ▶ L'estima toujours être basée sur la discussion et le consensus

Le planning poker

Une autre méthode d'estimation

C'est quoi le planning poker

- ▶ Ou encore Scrum Poker, est une technique d'estimation agile largement utilisée qui assure une estimation précise du travail et favorise le consensus au sein de l'équipe.



Déroulement du planning poker

Mise en place

- ▶ Chaque membre de l'équipe de développement reçoit un jeu de cartes.
- ▶ Ces cartes portent généralement des valeurs correspondant à la séquence de Fibonacci (1, 2, 3, 5, 8, 13, 21, etc.) ou autre suite

Déroulement du planning poker

Le consensus comme finalité

1. Le Product Owner commence par présenter la user story ou l'élément du backlog à estimer.
2. Il en décrit les fonctionnalités, les attentes et les résultats attendus.
3. Une fois que tous les détails sont clairs, chaque membre de l'équipe choisit une carte de son jeu qui, à son avis, représente l'effort nécessaire pour réaliser la tâche.
4. Une fois que tout le monde a choisi sa carte, tous les membres de l'équipe révèlent simultanément leur estimation.
5. Si toutes les estimations sont similaires, la valeur est acceptée.
6. Cependant, si les estimations diffèrent considérablement, une discussion est nécessaire.
7. Les membres de l'équipe aux estimations les plus élevées et les plus basses expliquent leur raisonnement.
8. Après la discussion, le processus est répété jusqu'à ce qu'un consensus soit atteint.

Avantages du planning poker

- ▶ Encourage la communication et la transparence, permet de prendre en compte les perspectives de tous les membres de l'équipe, et aide à éviter les erreurs d'estimation.
- ▶ Favorise également un consensus et garantit que chaque membre de l'équipe est investi dans le processus d'estimation.

Exemple planning poker

Mise en situation

- ▶ Vous êtes le PO
- ▶ L'équipe a déjà réussi à mettre en place une fonctionnalité de suivi de la course à pied, qui permet aux utilisateurs de suivre leur distance parcourue, leur rythme, le temps écoulé et les calories brûlées.
- ▶ Maintenant, vous voulez développer une nouvelle fonctionnalité pour suivre les séances d'entraînement de musculation.
 - ▶ Cela nécessite que les utilisateurs puissent entrer le type d'exercice, le poids utilisé, le nombre de séries et de répétitions, et voient alors apparaître une estimation des calories brûlées. De plus, vous voulez que l'application propose des séances d'entraînement de musculation prédéfinies.
- ▶ Vous convoquez une réunion de Planning Poker avec l'équipe de développement pour estimer l'effort nécessaire pour réaliser cette tâche. Vous expliquez la fonctionnalité en détail et répondez aux questions des développeurs.

Exemple planning poker

Choix des cartes

- ▶ Vous demandez à chaque membre de l'équipe de choisir une carte de leur jeu qui, à leur avis, représente l'effort nécessaire pour développer cette fonctionnalité.
- ▶ Une fois que tout le monde a choisi sa carte, tous les membres révèlent simultanément leur estimation.V
 - ▶ Vous constatez qu'il y a une certaine divergence dans les estimations - certains développeurs ont choisi le chiffre 13, d'autres le 21, et un a même choisi le 34.
- ▶ Vous demandez alors à ceux qui ont choisi les chiffres les plus élevés et les plus bas d'expliquer leur raisonnement.
 - ▶ Le développeur qui a choisi 34 explique qu'il pense qu'il y aura des défis techniques à surmonter pour estimer correctement les calories brûlées pendant les séances de musculation.
 - ▶ Ceux qui ont choisi 13 pensent que, puisque l'application a déjà une fonctionnalité de suivi pour la course à pied, cela pourrait simplifier le développement de la fonctionnalité de musculation.

Exemple planning poker

Discussions et finalisation

- ▶ Après une discussion approfondie, l'équipe est prête à refaire une estimation.
- ▶ Cette fois, la plupart des membres de l'équipe s'accordent sur le chiffre 21, ce qui semble être une estimation réaliste de l'effort nécessaire.
- ▶ Vous notez cette estimation pour la fonctionnalité de suivi de la musculation dans le product backlog.

À vous en équipe...

- ▶ **Étape 1 : choisissez un projet hypothétique**
 - ▶ Pour commencer, imaginez un projet hypothétique, par exemple, la création d'un site web pour une petite entreprise locale.

- ▶ **Étape 2 : estimez les tâches**

- ▶ **Étape 3 : comparez vos estimations**

- ▶ **Étape 4 : revoyez vos estimations**

Avantages du planning poker

- ▶ **Discussion ouverte et productive**
 - ▶ Chaque membre de l'équipe a la possibilité de donner son avis, ce qui conduit à une meilleure compréhension du travail à réaliser.
- ▶ **Estimations plus précises**
 - ▶ Permet d'obtenir des estimations plus précises, car elle prend en compte les perspectives et l'expertise de toute l'équipe.
- ▶ **Renforcement de l'esprit d'équipe**
 - ▶ Crée un environnement de travail plus positif.

Inconvénients du planning poker

- ▶ **Processus potentiellement long**
 - ▶ Il est important de gérer le temps de manière efficace et d'éviter de se perdre dans les détails.
- ▶ **Désaccords**
 - ▶ Il est important d'aborder ces désaccords de manière constructive et de parvenir à un consensus.
- ▶ **Complexité pour les grands projets**
 - ▶ Il peut être nécessaire de décomposer les travaux en tâches plus petites et gérables.
- ▶ **Risque de sur ou sous-estimation**
 - ▶ Il est donc crucial de réévaluer régulièrement les estimations à mesure que l'équipe acquiert de l'expérience.

Erreur fréquente lors du planning poker

- ▶ Trop de détails
- ▶ Le biais de l'expert
- ▶ Se focaliser uniquement sur la difficulté technique
- ▶ Oublier de réestimer
- ▶ Mauvaise interprétation des estimations
- ▶ Manque de cohérence dans les estimations
- ▶ Utilisation de la méthode pour des tâches trop petites ou trop grandes

Atelier Carpaccio

- ▶ Technique utilisée pour décomposer les fonctionnalités ou les exigences du projet en petites tâches gérables
- 1. **Définition du but**
 - ▶ Définir clairement l'objectif. Cela peut être l'estimation du coût de développement d'une nouvelle fonctionnalité, la décomposition d'une tâche complexe en sous-tâches plus petites, etc.
- 2. **Présentation de la fonctionnalité ou de l'exigence**
 - ▶ Doit être suffisamment détaillée pour que tous les participants puissent comprendre ce qui est attendu.
- 3. **Découpage de la fonctionnalité**
 - ▶ L'objectif est de découper la fonctionnalité en tâches suffisamment petites pour être estimées et mises en œuvre indépendamment les unes des autres.
- 4. **Estimation des sous-tâches**
 - ▶ Cela permet d'obtenir une estimation plus précise de la fonctionnalité dans son ensemble.
- 5. **Discussion et ajustement**
 - ▶ Cette discussion permet d'identifier les points de désaccord et de trouver un consensus sur l'estimation finale.

Atelier Carpaccio

Exemple

- ▶ Création d'un parcours personnalisé d'entraînement en fonction des objectifs spécifiques de l'utilisateur.
- ▶ Définition du but
 - ▶ Estimer le coût de développement de cette nouvelle fonctionnalité et de comprendre comment nous pouvons la décomposer en tâches plus petites.
- ▶ Présentation de la fonctionnalité
 - ▶ Permettre à un utilisateur de sélectionner son objectif de fitness (par exemple, perdre du poids, gagner de la masse musculaire, améliorer son endurance, etc.), puis l'application générera un parcours d'entraînement personnalisé adapté à cet objectif.

Atelier Carpaccio

Exemple (suite)

▶ Découpage de la fonctionnalité

- ▶ conception de l'interface utilisateur pour la sélection des objectifs ;
- ▶ mise en place d'une base de données d'exercices et de parcours d'entraînement ;
- ▶ développement de l'algorithme de génération de parcours d'entraînement personnalisé ;
- ▶ intégration de la nouvelle fonctionnalité avec le reste de l'application ;
- ▶ test de la fonctionnalité et correction des bugs.

▶ Estimation des sous-tâches

- ▶ Par exemple, la conception de l'interface utilisateur pourrait être estimée à 5 points, la mise en place de la base de données à 8 points, le développement de l'algorithme à 13 points, l'intégration à 3 points et les tests à 5 points.

▶ Discussion et ajustement

- ▶ S'accorder sur une estimation finale pour chaque sous-tâche.



Planification d'une release



Une release, c'est quoi ?

- ▶ Une "release", également appelée version ou livraison, représente une version spécifique d'un produit logiciel qui est prête à être distribuée à ses utilisateurs finaux.
- ▶ Dans le cadre du développement agile, une release est un livrable majeur qui contient plusieurs fonctionnalités complètes, testées et prêtes à être utilisées par les clients.

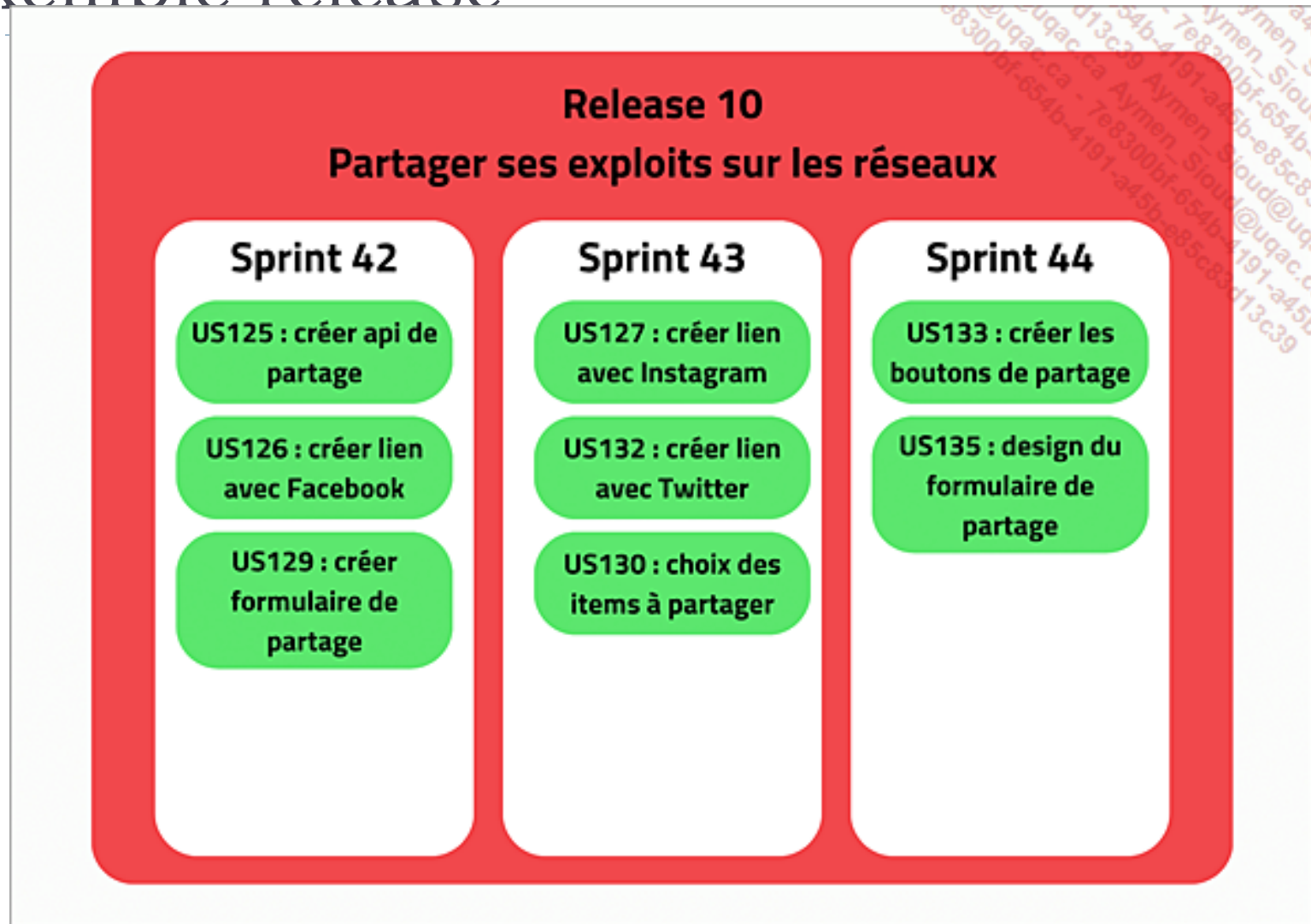
Création d'une release

- ▶ Le processus de création d'une release inclut
 - ▶ la conception,
 - ▶ le développement,
 - ▶ le test et
 - ▶ la validation de nouvelles fonctionnalités ou améliorations.
- ▶ L'ensemble de ces fonctionnalités est assemblé dans une version cohérente du produit qui est ensuite déployée dans l'environnement de production.
- ▶ Chaque release est dotée d'un numéro de version unique qui permet de suivre l'évolution du produit.

Une release, un objectif précis

- ▶ Chaque release peut également comporter des notes de version, qui sont des documents fournissant des détails sur les nouvelles fonctionnalités, les corrections de bugs et les améliorations apportées depuis la dernière version.
- ▶ La "release planning" ou planification de version, est une activité qui permet à l'équipe de déterminer quelles user stories du product backlog seront développées et livrées dans les prochains sprints pour composer une release.

Exemple release



Un PB priorisé

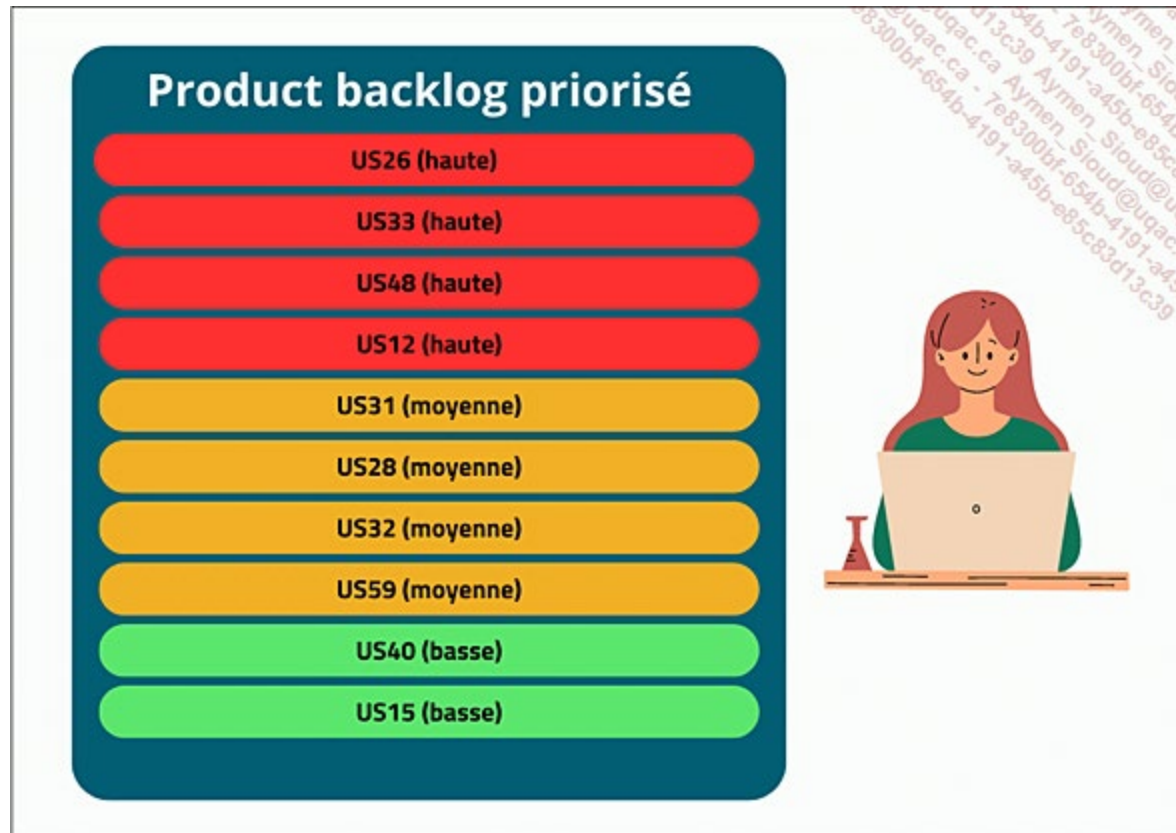
- ▶ La planification de release dépend fortement d'un product backlog correctement priorisé.
- ▶ Le product backlog est une liste de toutes les fonctionnalités, améliorations et corrections nécessaires pour le produit, ordonnées par ordre de priorité.
- ▶ La priorisation permet de s'assurer que les fonctionnalités les plus critiques et celles qui apportent le plus de valeur aux utilisateurs sont développées en premier lieu.

Un PB priorisé pour de meilleures estimations

- ▶ Une fois que le product backlog a été priorisé, l'équipe de développement peut alors estimer l'effort nécessaire pour chaque élément du backlog.
- ▶ Ces estimations, combinées à la vélocité de l'équipe (le nombre moyen de points de story que l'équipe peut réaliser dans un sprint), permettent de prévoir combien de sprints seront nécessaires pour développer une certaine quantité de travail.

Un PB priorisé

Exemple



Le point final

- ▶ Moment où la release sera prête à être déployée
- ▶ Permet de fixer un objectif réaliste et d'aligner les attentes de toutes les parties prenantes.
- ▶ Calculer le nombre de sprints nécessaires pour accomplir les items prioritaires du backlog.
- ▶ En utilisant la vélocité de l'équipe comme mesure, on peut estimer le temps nécessaire pour compléter le travail.
- ▶ Exemple
 - ▶ Si l'équipe a une vélocité moyenne de 20 points de story par sprint, et que le backlog priorisé totalise 100 points de story, on peut estimer qu'il faudra environ cinq sprints pour atteindre le point final.

Le point final

Estimation changeante

- ▶ L'estimation peut changer en fonction de divers facteurs, tels que la complexité des tâches, les imprévus, les changements de priorité, etc.
- ▶ Il est nécessaire de revoir régulièrement le point final et de l'ajuster en fonction de l'évolution du projet.

Déterminer la durée des sprints

- ▶ Dans le cadre de Scrum est généralement comprise entre une et quatre semaines.
- ▶ Le choix de la durée dépend de nombreux facteurs :
 - ▶ le type de projet,
 - ▶ la complexité des tâches,
 - ▶ la taille de l'équipe,
 - ▶ la disponibilité des membres de l'équipe,
 - ▶ le degré d'incertitude,
 - ▶ etc.

Déterminer la durée des sprints

À la recherche d'un rythme soutenu

- ▶ Favoriser un rythme de travail soutenable pour l'équipe, tout en permettant un feedback rapide et des ajustements réguliers.
- ▶ Une durée de sprint trop courte peut augmenter la pression et conduire à des réunions plus fréquentes, ce qui peut réduire le temps consacré au travail réel.
- ▶ Une durée de sprint trop longue peut retarder le feedback et la détection des problèmes, ce qui peut rendre les ajustements plus difficiles.

Établir le plan de release

- ▶ Une fois que tous les éléments sont en place, y compris la définition du produit minimum viable (MVP) et les stratégies de déploiement, il est temps d'établir le plan de la release.
- ▶ Ce processus se réalise généralement en collaboration avec toutes les parties prenantes du projet, y compris le Product Owner, l'équipe de développement et les parties prenantes clés.

Le plan de release

Contenu

- ▶ **Le plan de release doit intégrer**
 - ▶ les fonctionnalités à inclure dans le MVP, qui représente la version la plus simplifiée du produit pouvant être déployée,
 - ▶ les étapes de déploiement, qui impliquent la manière et le timing de la mise en production du produit ou des fonctionnalités auprès des utilisateurs finaux.
- ▶ Cette planification permet de s'assurer que le produit est développé et déployé de manière efficace, en répondant aux besoins des utilisateurs tout en maximisant la valeur ajoutée pour l'entreprise.

Le plan de release

Déroulement

1. Déterminer les objectifs de la release
2. Définir les sprints
3. Assigner les user stories aux sprints
4. Établir le plan

Vive la technologie

▶ Plusieurs outils sur le marché :

- ▶ Jira
- ▶ Trello
- ▶ Microsoft Azure DevOps
- ▶ VersionOne
- ▶ Rally

Pièges à éviter dans l'estimation

-
- ▶ **Piège #1 : Penser qu'une estimation est exacte**
 - ▶ Ne pas oublier que c'est une approximation

 - ▶ **Piège #2 : Penser pouvoir tout estimer en une seule fois**
 - ▶ Répartir l'activité d'estimation en plusieurs courtes séances plutôt que de vouloir traiter l'intégralité d'une liste d'artefacts en une seule passe.

 - ▶ **Piège #3 : Penser que l'estimation ne repose que sur une seule personne**
 - ▶ Il faut s'assurer que toutes les personnes ont eu droit à la parole et ont pu exprimer leur point de vue.

 - ▶ **Piège #4 : Passer plus de temps à discuter du contenu d'un artefact qu'à l'estimer**
 - ▶ Assurez-vous d'abord que les User Stories soient suffisamment comprises lors des séances d'affinage du Product Backlog. Cela aidera à fluidifier et à rendre efficaces les séances d'estimation. Si ce n'est pas le cas, alors cela signifie que les User Stories ne sont pas prêtes à être estimées.

 - ▶ **Piège #5 : Continuer à utiliser les mêmes unités de mesure**
 - ▶ En agile, les estimations doivent être des estimations relatives entre les artefacts, d'une part sur la valeur (en Value Points) et d'autre part sur l'effort de réalisation (en Story Points). Chaque équipe doit donc se mettre d'accord sur un étalon, et c'est sur la base de cet étalon que seront réalisées toutes les autres estimations.