



GPI1024

Gestion de projet informatique

La méthode « SCRUM »



Qu'est ce qu'une histoire utilisateur?

Histoire utilisateur

Une **histoire utilisateur** décrit une fonctionnalité qui sera utile soit pour un utilisateur, soit pour l'acheteur d'un système ou d'un logiciel.

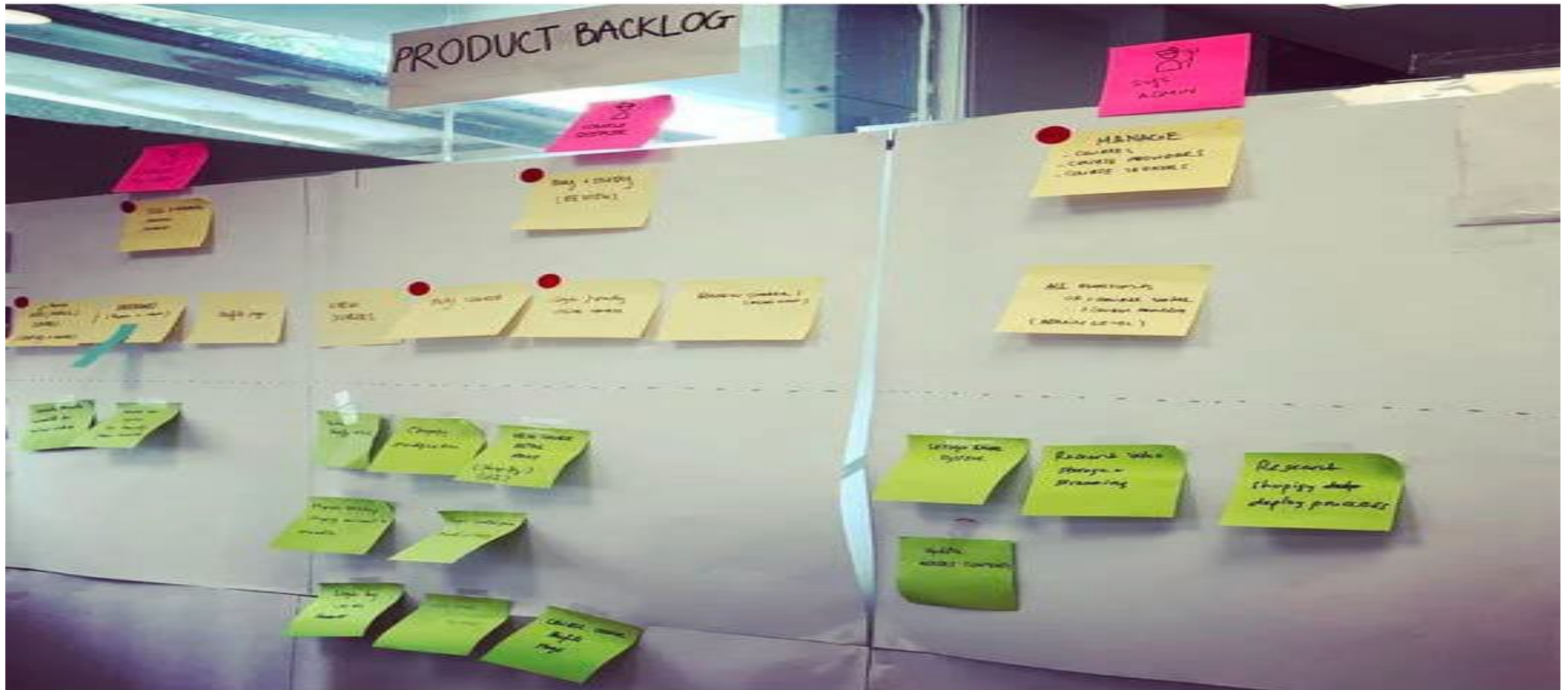
Les histoires utilisateur sont une manière puissante et populaire d'exprimer les besoins des utilisateurs.

Carte histoire utilisateur

Une carte histoire utilisateur est la manifestation physique d'une histoire utilisateur.

La description des histoires utilisateur sont traditionnellement rédigées à la main sur des cartes en papier, mais peut également être un enregistrement numérique dans un outil logiciel.

Carte histoire utilisateur



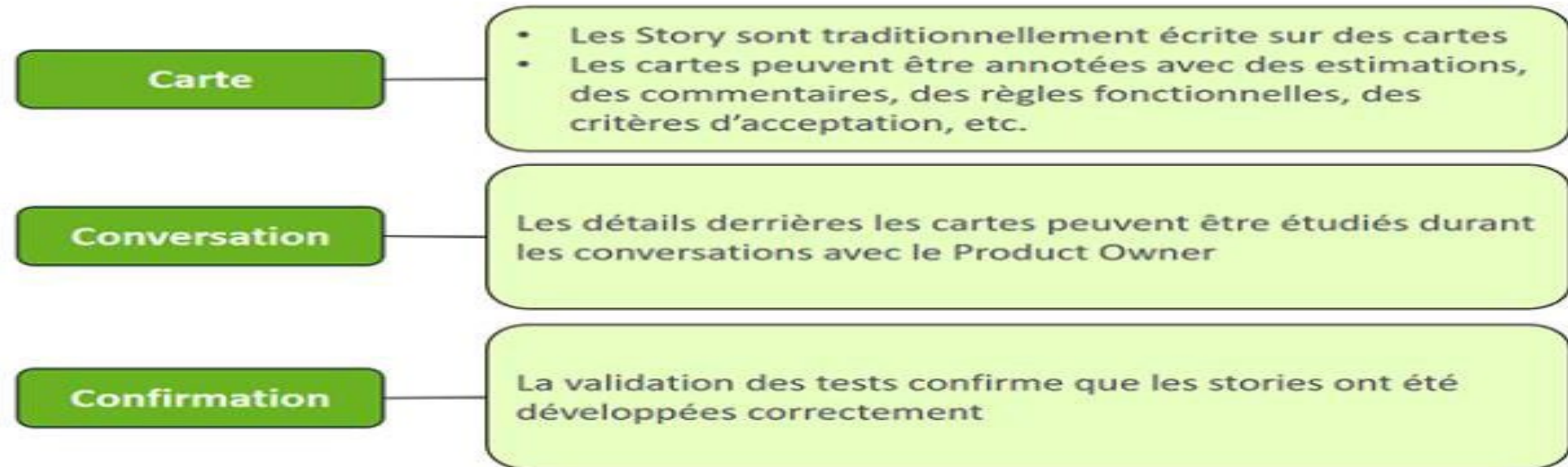
User Story Mapping

Le modèle 3C d'histoire utilisateur

Les 3 C de Ron JEFFRIES



Ron JEFFRIES

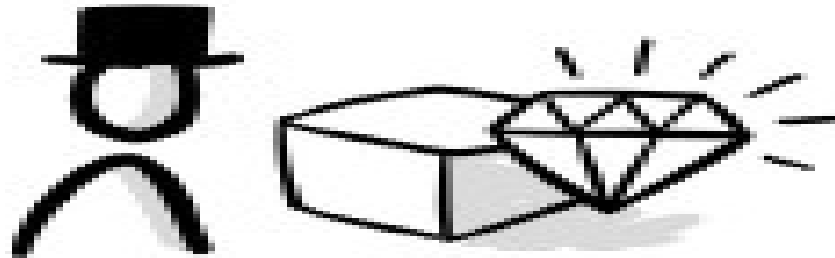


Ron JEFFRIES – Règle des 3C pour les User Stories

Comment écrire une histoire utilisateur?

Vision du produit

C'est le **Product Owner** qui a la **vision du produit** et apporte de la valeur au produit. Il est responsable de la rédaction des « Users Stories », dont l'ensemble constitue le « Product Backlog ».



Idéalement, le Product Owner partage sa vision avec **l'équipe client**.

L'équipe client

L'équipe client comprend ceux qui veillent à ce que le logiciel réponde aux besoins des utilisateurs. Cela peut inclure des testeurs, un chef de produit, de vrais utilisateurs.

L'équipe client rédige les "story cards" (cartes d'histoire) parce qu'elle est la mieux placée pour exprimer les fonctionnalités souhaitées et parce qu'elle doit ensuite être en mesure de discuter des détails de l'histoire avec les développeurs et de prioriser les histoires.

L'équipe client

L'équipe client, plutôt que les développeurs, rédige les user stories pour deux raisons principales:

- Premièrement, chaque histoire doit être écrite dans le **langage des affaires (du métier)**, et non dans un jargon technique, afin que l'équipe client puisse hiérarchiser les histoires à inclure dans les itérations et les versions.
- Deuxièmement, en tant que principaux visionnaires du produit, l'équipe client est la mieux placée pour **décrire le comportement du produit**.

Exemple de “User Story ”

En mettant en avant le rôle, une User story s'écrit comme suit:

En tant que "rôle"
Je veux "fonctionnalité"
Afin de "valeur ajoutée ou résultat"

Exemple:

En tant que client de la banque,
Je veux retirer des billets à l'aide d'un distributeur automatique de billet,
Afin d'obtenir de l'argent facilement.

Exemple de “User Story ”

En mettant en avant la valeur, une User story peut s'écrire comme suit:

Afin de "valeur ajoutée ou résultat"

En tant que "rôle"

Je veux "fonctionnalité"

Exemple:

Afin d'obtenir de l'argent facilement,

En tant que client de la banque,

Je veux retirer des billets à l'aide d'un distributeur automatique de billet.

Exemple de “User Story ”

En mettant en avant le comportement, une User story peut s'écrire comme suit:

Étant donné que "condition initiale"

Quand un "évènement" apparaît

Alors "le résultat attendu est"

Exemple:

Étant donné que j'ai 20\$ sur mon compte en banque ET que je me suis identifié avec succès

Quand je demande un retrait de 10\$

Alors je reçois deux billets de 5\$ et un reçu imprimé.

Exemple d'une "Macro User Story"

- Une macro user story est communément appelée **Epic**.
- L'Epic peut être décomposée en plusieurs User stories.

Exemple:

Authentification de l'utilisateur

- En tant que "utilisateur"
Je veux :
 - "Un écran de **connexion** utilisateur"
 - "Une procédure de **mot de passe oubliée** "
 - "Un bouton de **déconnexion**"
- En tant que "responsable de la sécurité "
Je veux :
 - "verrouiller le compte après trois tentatives infructueuses"

INVEST

- Un acronyme des critères de qualité pour une histoire utilisateur, basé sur une idée originale de Bill Wake.

	I	Indépendant
	N	Négociable
	V	Valeur
	E	Estimable
	S	Small
	T	Testable

Conditions de satisfactions

Pour chaque user story, on doit définir des critères de satisfactions ou d'acceptation.

Les critères d'acceptation (ou acceptance criterias en anglais) sont rédigés par le Product Owner (parfois en collaboration avec l'équipe de développement) et accompagnent chaque user story.

Ils représentent un ensemble de conditions que la story doit satisfaire pour être considérée comme complète et terminée.

Le « Product Backlog »

Le Product Backlog

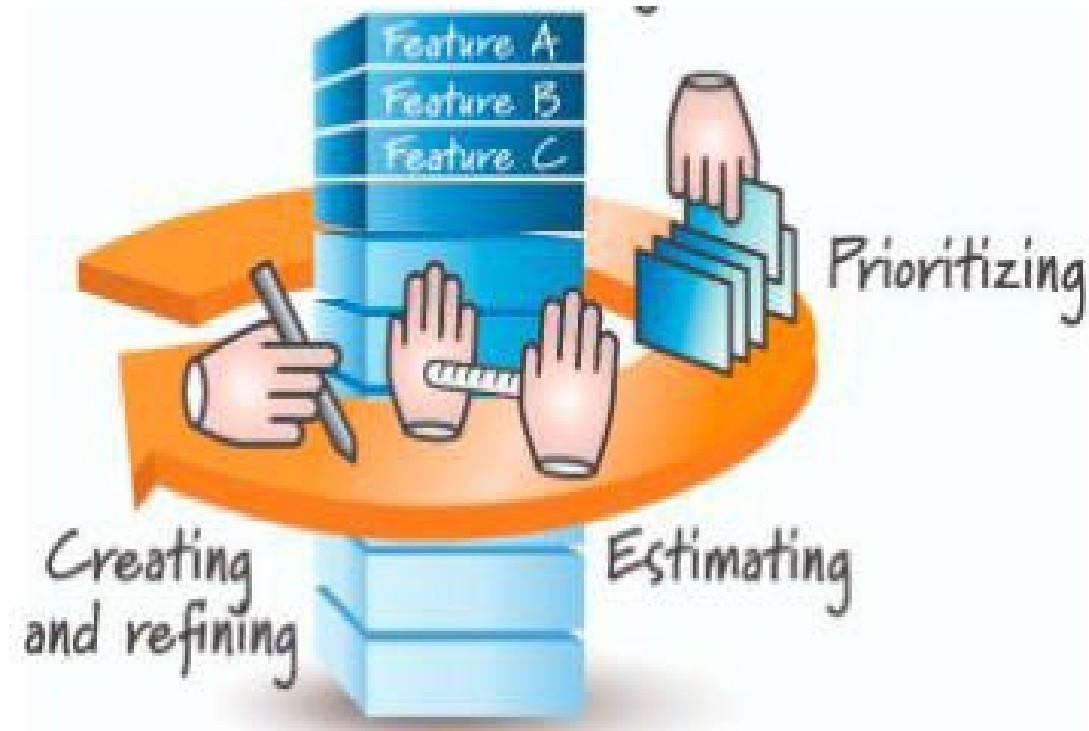
Le Product Backlog est une liste de priorisés de fonctionnalités.

Le Product Backlog existe et évolue tout au long du cycle de vie du produit, c'est la feuille de route du produit.

Tout au long du projet, le Product Backlog centralise la liste de tout ce qui peut être fait par l'équipe, par ordre de priorité.

Il n'existe qu'un seul Product Backlog pour un produit.

Le Product Backlog



Priorisation des éléments

Stories	Priorités
En tant que joueur, je veux pour utiliser mon épée	1
En tant que joueur, je veux pouvoir sauter	1
En tant que joueur, je veux pouvoir sélectionner des items dans mon inventaires et les équipés	2

Il existe différentes façons de désigner des priorités. La classique est l'utilisation d'adjectifs : faible/moyenne/forte. On peut aussi utiliser des chiffres de 1 à 5 ou à 10. Plusieurs références Agile suggèrent d'utiliser plutôt la méthode MoSCoW. Au final, le choix a peu d'importance. L'important est d'avoir un système de priorisation efficace.

Critères de priorisation

Valeur : Est-ce que l'histoire est importante pour l'utilisateur?

Coût : Quel est l'effort nécessaire pour compléter l'histoire?

Risque : Est-il possible de réaliser cette histoire? Incertitude de valeur et coût.

Connaissance : Est-ce que les connaissances acquises avec cette histoire seront utiles pour d'autres histoires?

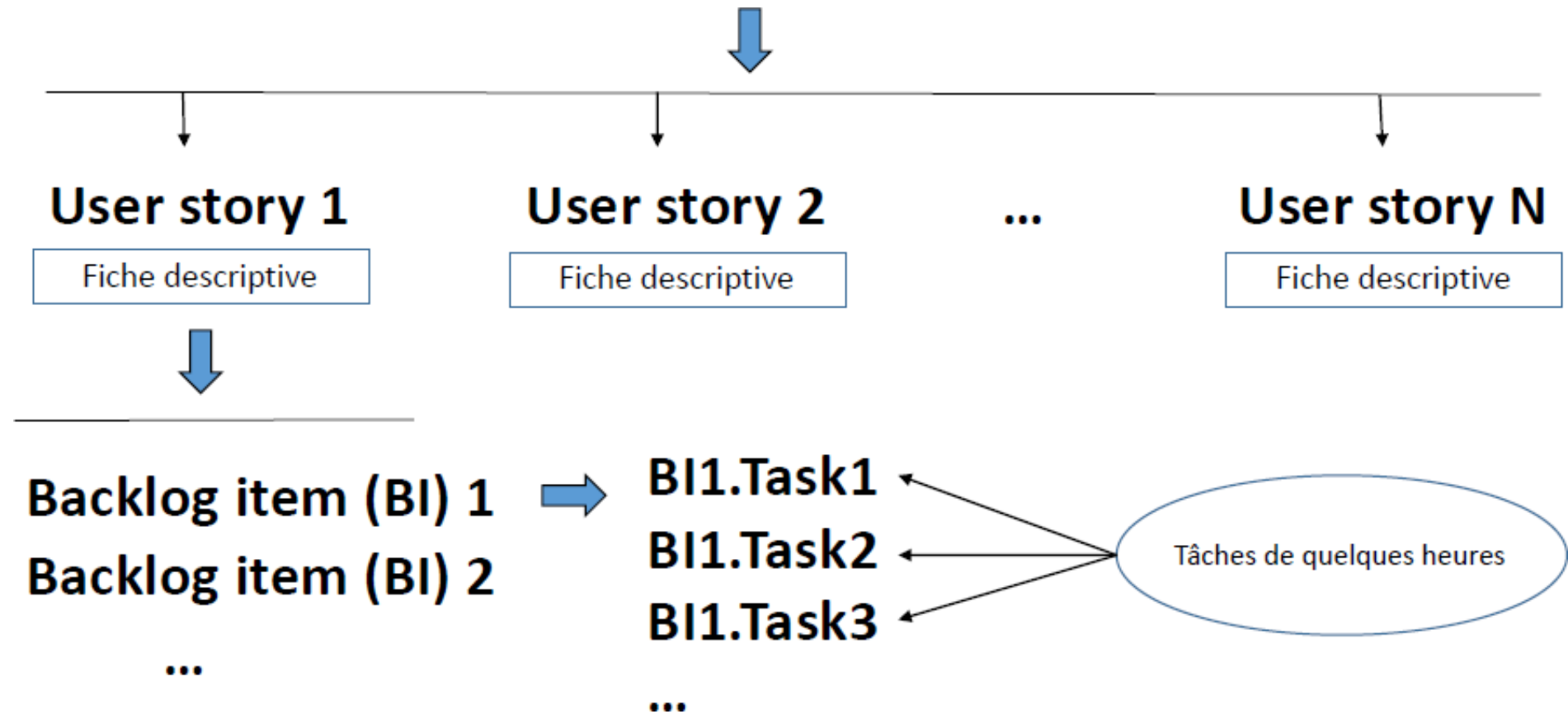
La méthode MoSCoW pour prioriser

Une des techniques utilisées pour prioriser les besoins dans un contexte itératif est celle de **MoSCoW**. L'avantage de la méthode MoSCoW réside dans la signification de l'acronyme, qui est plus compréhensible que d'autres techniques de priorisation comme **élevé/moyen/faible**.

- **M (must have)** : doit être fait (**vital**).
- **S (should have)** : devrait être fait dans la mesure du possible (**essentiel**).
- **C (could have)** : pourrait être fait dans la mesure où cela n'a pas d'impact sur les autres tâches (**souhaitable**).
- **W (won't have)** : ne sera pas fait cette fois mais sera fait plus tard (**luxé, optimisation budgétaire**).

Le “Sprint Backlog”

Le “Sprint Backlog”

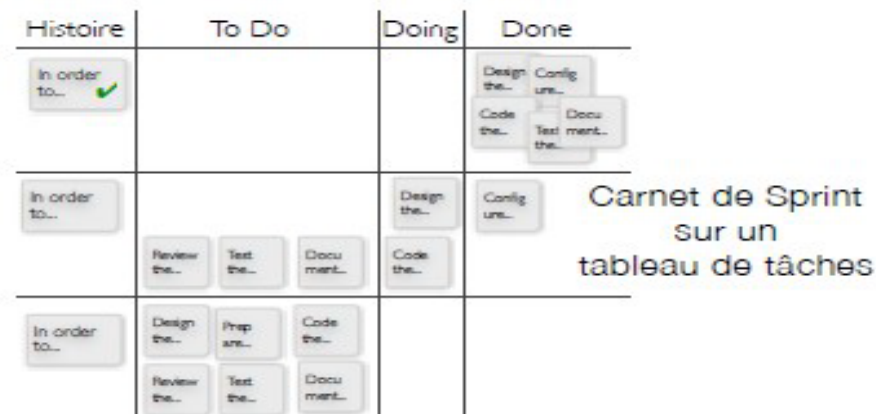


Le “Sprint Backlog”

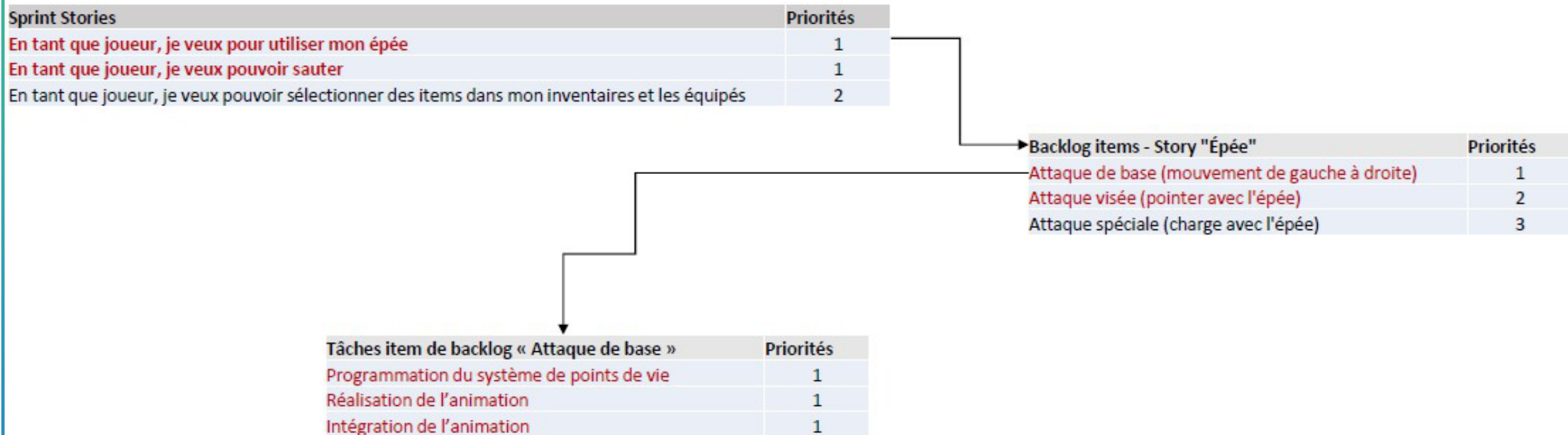
La plupart des Equipes de développement **visualisent** leur Carnet de Sprint sur un **tableau de tâches**, qui est la liste des éléments que l'Equipe s'est engagée à faire pendant le Sprint.

Le tableau de tâches est un exemple de **kanban**, un mot japonais signifiant "signal visible".

Il montre à l'équipe Scrum et à quiconque, le travail prévu pendant le Sprint et l'état d'avancement actuel.



Le “Sprint Backlog”



Dans ce cas, le Sprint consisterait à développer les user-story de l'épée et du saut. Par contre, seulement certains éléments du backlog de ces histoires seront développés, en fonction de leur priorité. De même, seul les tâches reliés au développement de ces éléments (items) seront effectuées.

Estimation d'histoires utilisateurs

La pratique du « Planning Poker »

La méthode du **Planning Poker** est une technique basée sur le consensus d'experts. Elle est la principale technique utilisée pour **estimer l'effort relative des histoires utilisateurs** dans les approches Agile/ Scrum.

Le Planning Poker a été défini et nommé par James Grenning en 2002 et popularisé plus tard par Mike Cohn dans son livre "Agile Estimating and planification ", dont la société commerciale a marqué le terme.

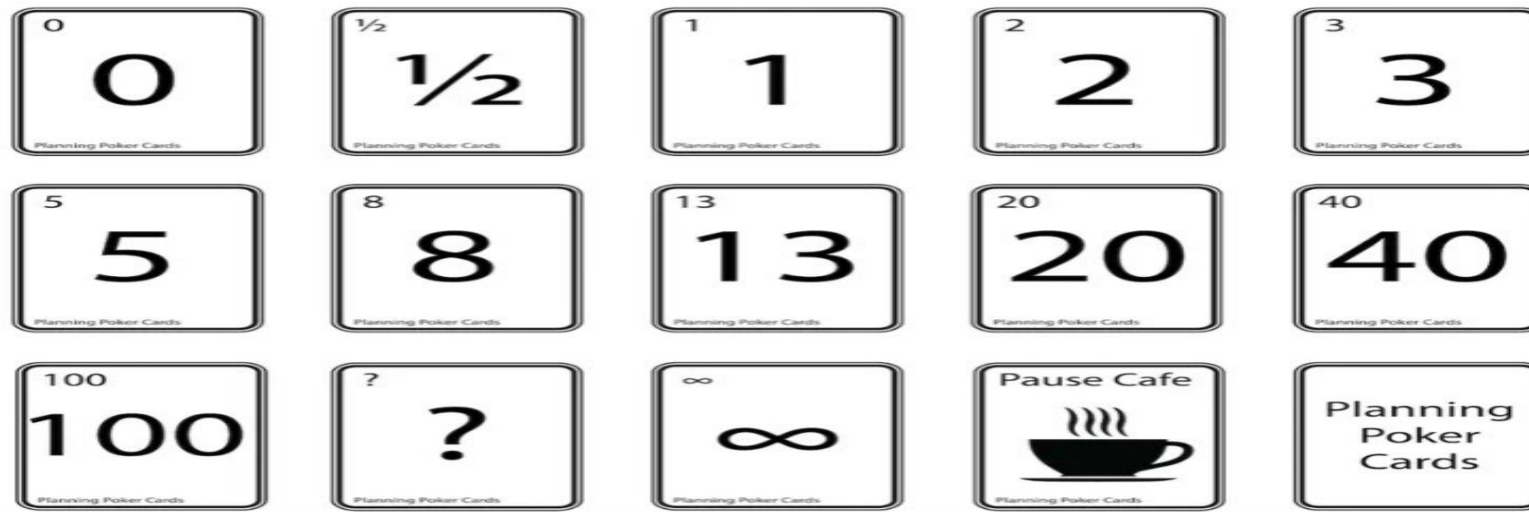
La pratique du « Planning Poker »

Dans la technique du Planning Poker, les estimations des histoires utilisateurs sont dérivées en jouant au poker de la planification avec des cartes à jouer. Toute l'équipe Scrum est impliquée et il en résulte des estimations rapides, mais fiables.

Planning Poker se joue avec un jeu de cartes. Comme la suite de Fibonacci est utilisée, les cartes ont des numéros 1, 2, 3, 5, 8, 13, 21, 34, etc. Ces chiffres représentent les «Points Histoire». Chaque estimateur a un jeu de cartes. Les chiffres sur les cartes devraient être assez grands pour être visibles à tous les membres de l'équipe, lorsque l'un des membres de l'équipe détient une carte. De façon générale, il faut comprendre que les points représentent un ordre de grandeur pour le développement informatique lié à l'histoire estimée.

La pratique du « Planning Poker »

La figure ci-dessous nous montre un exemple de cartes à jouer. Elles représentent des unités, que l'on appelle des « **story points** » ou « **points d'histoire** ». Ces points représentent le **niveau d'effort nécessaire pour effectuer une tâche**. Elles ne représentent pas directement des durées.



La pratique du « Planning Poker »

- Le processus du PP

1 – Explications du scénario utilisateur : Le responsable de produit (en Scrum) explique aux autres participants un scénario utilisateur (user story). Il tente de bien expliquer toutes les facettes et items.

La pratique du « Planning Poker »

- Le processus du PP

2 – Questions/réponses sur le scénario utilisateur : Les participants posent des questions au responsable de produit (product owner), discutent du périmètre du scénario (*scope*), évoquent les conditions de satisfaction qui permettront de le considérer ce scénario comme « terminé ». Ils peuvent demander des clarifications du scénario.

La pratique du « Planning Poker »

- Le processus du PP

3 – Estimation individuelle du scénario : Chacun des participants évalue la complexité de ce scénario, choisit la carte qui correspond à son estimation de la complexité et la dépose, face vers le bas, sur la table devant lui.

La pratique du « Planning Poker »

- Le processus du PP

4– Estimation collective : Au signal du facilitateur, les cartes sont retournées en même temps. S'il n'y a pas unanimité, la discussion reprend. On répète le processus d'estimation jusqu'à l'obtention de l'unanimité.

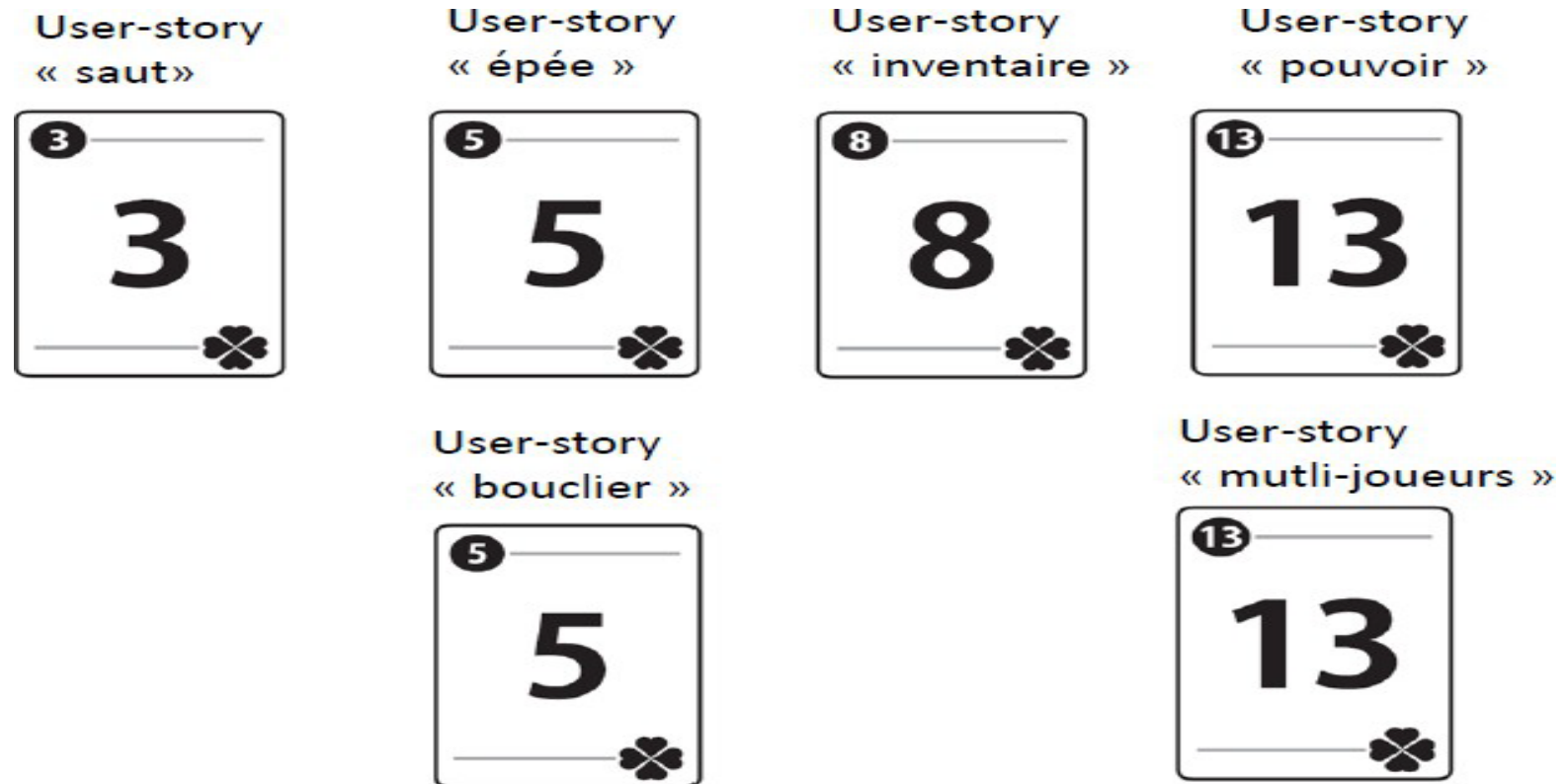
La pratique du « Planning Poker »

Ajuster par ordre de grandeur

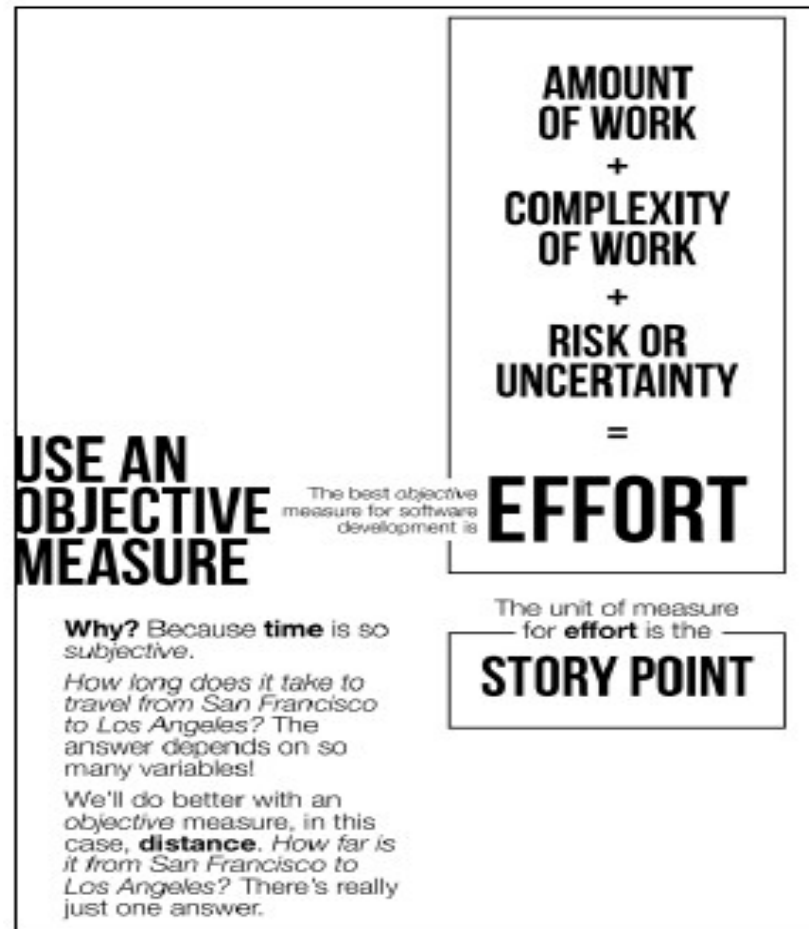
L'objectif est que l'équipe s'entende sur l'ordre de grandeur. Si, par exemple, l'histoire « épée » est évaluée à 5 points, une autre histoire qualifiée de taille comparable devrait aussi faire 5 points. De la même manière, si l'équipe évalue une nouvelle histoire à 5 points, mais qu'en discutant, la majorité s'entendent pour dire que celle-ci est d'envergure plus grande que l'histoire « épée », elle aussi évaluée à 5 points, alors l'équipe devrait l'évaluer à niveau plus élevé, comme 8 points. L'idée est de toujours garder un ordre de grandeur consistant.

La pratique du « Planning Poker »

Ajuster par ordre de grandeur



Estimation de l'effort



Estimation de l'effort

Traduire les story points en temps (en heures ou en jours)

À la fin, on établit la valeur en temps d'un « **story point** ». En anglais, on parle de la « **velocity** » (**vélocité**) qui est le nombre de « story points » de travail que l'équipe de développement peut caser dans une « **timebox** » dont la durée est connue . Ainsi, on va évaluer le nombre de points qui peuvent être effectués par l'équipe à l'intérieur d'un sprint. En bout de ligne, on démarre avec des estimations en ordre de grandeur d'effort pour en extirper, au final, des estimations en temps. Par exemple, l'équipe pourrait statuer que sa vélocité est de 14 points par cycle de deux semaines. Dans ce cas, une histoire utilisateur estimée à 8 points prendrait donc un peu plus d'une semaine.

Avantage du Planning Poker

En synthèse, en termes d'avantages, le Planning Poker tire parti de **l'intelligence collective** en matière de prise de recul et d'idées sur les modalités de réalisation.

Le Product Owner peut **partager sa vision du produit** à cette occasion avec l'équipe qui le réalise. Il bénéficie également d'une **transparence absolue** sur les estimations et peut même ajuster son besoin en séance pour arriver à un **“coût”** de réalisation optimale par rapport à la valeur associée à son besoin.

L'équipe de développement quant à elle a la possibilité de poser ses questions directement, plutôt qu'uniquement à partir d'un cahier des charges papier.

Enfin, une fois la technique rodée, les estimations sont rapides grâce au recours à la suite de Fibonacci et au **principe d'estimation par relativité (ordre de grandeur)**.

Pourquoi une planification AGILE?

Planification AGILE

Livrer les fonctionnalités avec le plus de valeur en premier

Créer un réflexe multidisciplinaire

Meilleure estimation de la progression

Planifier continuellement selon l'émergence des fonctionnalités

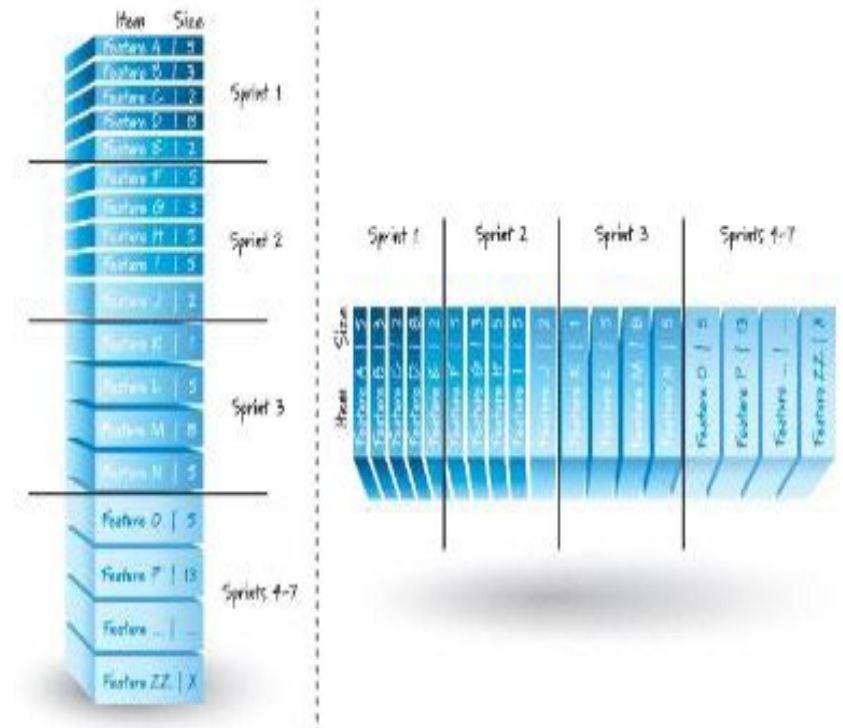
Travailler avec des objectifs clairs et communs

Meilleure communication avec les parties prenantes

Amélioration des prévisions en cours de production

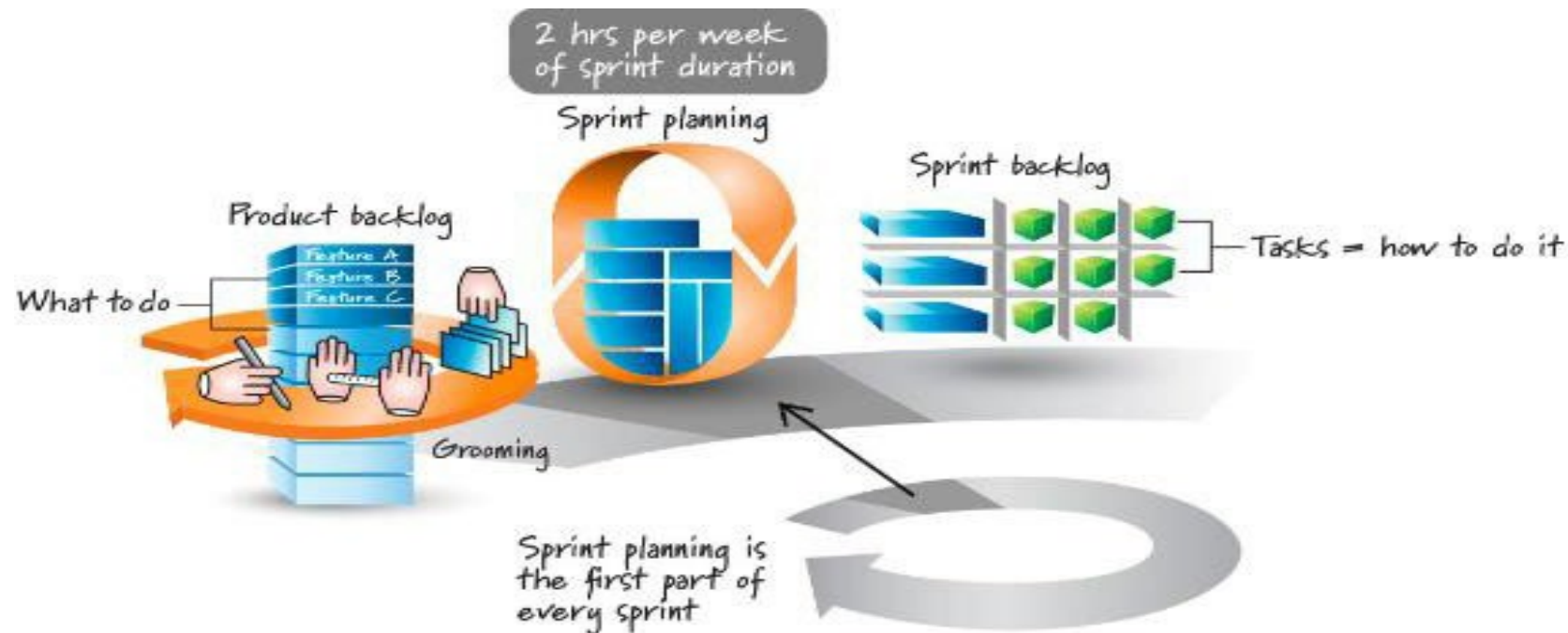
Release Planning

Planification approximative de quelles histoires utilisateurs seront réalisées durant les prochains Sprints.



Sprint Planning

Planifier le travail à faire pour le Sprint courant.



Références

Anny P. Murray, The Complete Software Project Manager: Mastering Technology from Planning to Launch and Beyond, Wiley, 2016.

Andrew Stellman, Jennifer Greene, Applied Software Project Management, O'Reilly, 2006.

PMI, Guide du Corpus des connaissances en management de projet (Guide PMBOK), 5ième édition, Project Management Institute (PMI), 2013.

PMI, Guide du Corpus des connaissances en management de projet (Guide PMBOK), 6ième édition, Project Management Institute (PMI), 2017.

Éléments de gestion de projets, Gilles Boulet PMP, 2009.

Ordre des conseillers en ressources humaines agréés. <http://www.portailrh.org> Kenneth S. Rubin, Essential Scrum: A Practical Guide to the Most Popular Agile Process, Addison-Wesley, 2012.