

# Scrum

Support de cours de la formation Certified Scrum Product Owner

[@BrunoSbille](#)

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

# Table of contents

<b>Scrum: le framework .....</b>	<b>3</b>
Qu'est-ce que Scrum ? .....	4
Scrum: un aperçu du framework.....	5
Scrum: les rôles .....	6
Scrum: les rituels .....	7
Scrum: les artéfacts .....	8
<b>Scrum et les principes “Agile” .....</b>	<b>9</b>
Les principes de Scrum.....	10
Les valeurs de Scrum .....	13
Historique de Scrum .....	14
Agile .....	15
<b>Le Product Backlog et les user stories .....</b>	<b>16</b>
Un Product Backlog ‘complet’ .....	17
User stories .....	18
INVESTir dans les User Stories .....	19
Les pièges des User Stories .....	20
<b>Agile: estimation et planning .....</b>	<b>21</b>
Relative estimations .....	22
Vélocité.....	24
Représentation graphique : Burndown chart .....	25
<b>Scrum: les Rituels .....</b>	<b>26</b>
Redéfinition du Product Backlog .....	27
Sprint Planning.....	28
Daily Scrum .....	30
Sprint Review .....	32
Sprint Retrospective .....	34

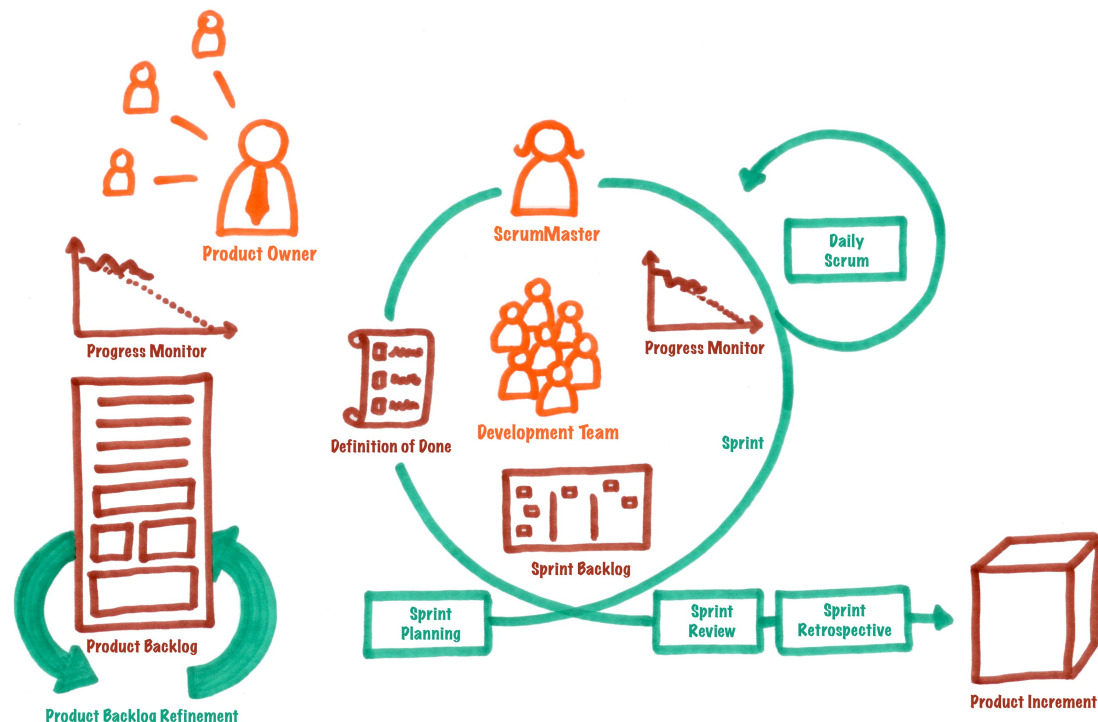
# Scrum: le framework

# Qu'est-ce que Scrum ?

Scrum est un cadre de travail (**framework**) permettant de créer des produits. L'objectif de Scrum est de permettre de créer des produits avec une équipe auto-organisée qui utilise des cycles de temps court (itérations). Chaque cycle permettant de délivrer un incrément de produit potentiellement livrable. De cette façon, Scrum maximise le contrôle, la flexibilité, la qualité et le retour sur investissement.

Scrum est un framework qui comporte un nombre limité de rôles, de rituels et d'artéfacts qui fonctionnent ensemble en respectant certaines règles. En plus de cela, Scrum requiert et fait émerger un ensemble de valeurs et de principes.

# Scrum: un aperçu du framework



Scrum commence lorsque différentes parties prenantes (**stakeholders**) ont besoin d'un certain produit. Le **Product Owner** façonne une vision pour le produit et collecte les objectifs et spécifications du produit dans le **Product Backlog**. Il est responsable pour que l'on puisse extraire un maximum de valeur du produit, et ce en prenant en compte le temps, le budget et d'autres contraintes. Il ou elle accomplit cela en **raffinant** continuellement le **Product Backlog** à l'aide de la Development Team.

La **Development Team** est une équipe pluridisciplinaire et auto-organisée qui délivre un incrément de produit qui est potentiellement livrable et qui amène de la valeur. Elle réalise l'incrément de produit lors de courtes itérations à durée fixe, appelées **Sprints**. Le **ScrumMaster** agit en tant que guide et coach pour l'équipe et l'organisation, il les supporte en mettant en place Scrum correctement et en gérant tous les obstacles qui pourraient les ralentir dans la livraison. Le Product Owner, ScrumMaster et la Development team sont appelés la **Scrum Team**.

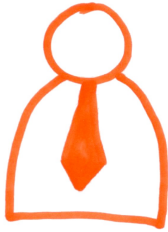
Un Sprint débute par le **Sprint Planning**. C'est un rituel à durée fixe pendant lequel la Development Team prend des éléments du Product Backlog afin de les réaliser dans le Sprint, tout tenant compte des priorités du Product Owner. Le résultat du Sprint Planning s'appelle le **Sprint Backlog**, qui est un plan de réalisation pour le Sprint, il contient les prévisions de l'équipe pour le **Product Increment** qui seront délivrées dans ce Sprint. La Scrum Team utilise le **Definition of Done** pour planifier le travail et pour décider si un élément du Product Backlog est vraiment finalisé et amène de la valeur métier.

Chaque jour, la Scrum Team se réunit pour une courte réunion à durée fixe, appelée **Daily Scrum**, elle permet de vérifier leur avancement, relever les problèmes et s'assurer que l'objectif du Sprint sera atteint. La Scrum Team utilise des **Progress Monitors** pour suivre et visualiser l'avancement du travail.

A la fin du Sprint, la Scrum Team organise le **Sprint Review** pour présenter le Product Increment au Product Owner et aux stakeholders, pour générer du feedback à propos du produit et pour discuter de son évolution. A la suite du Sprint review, la Scrum team prend du recul sur le Sprint écoulé et évalue leur process dans la **Sprint Retrospective**, avec pour résultat, un plan d'action concret pour le prochain Sprint.

# Scrum: les rôles

Scrum définit trois rôles différents: le Product Owner, le ScrumMaster and la Development Team. Ensemble ils sont appelés la Scrum Team.



**Product Owner**

- ✓ Sa responsabilité: maximiser la valeur produit en tenant compte des contraintes de temps et de budget.
- ✓ Représente les clients, les utilisateurs et stakeholders.
- ✓ Gère et ordonne le Product Backlog.



**Development Team**

- ✓ Une équipe multi-disciplinaire avec les personnes nécessaires (compétences et connaissance) pour délivrer des incréments de produit "terminés" (done).
- ✓ L'équipe est auto-organisée.
- ✓ Responsabilité: accomplir l'objectif du sprint et délivrer un incrément de produit "terminé" (done).



**ScrumMaster**

- ✓ Un leader au service des autres. Il ou elle supporte la Scrum Team et l'organisation en mettant en place Scrum de manière correcte.
- ✓ Sa responsabilité: s'assurer que les obstacles sont identifiés et résolus rapidement.

# Scrum: les rituels

Scrum définit 4 activités ayant une durée fixe:

## Sprint Planning

Au début du Sprint, La Scrum Team se réunit pour prévoir et planifier le Sprint. Les membres de la Scrum Team décident de ce qui sera réalisé, du nombre d'éléments à prendre dans le Product Backlog pour l'inclure dans le Sprint Backlog, et de la façon dont le travail sera réalisé.

## Daily Scrum

Chaque jour, Les membres de la Development Team se réunissent pour une courte réunion (max. 15 minutes). Ils discutent l'avancement du Sprint en cours, les éventuels obstacles qui ralentissent leur progression et ce qui est nécessaire pour atteindre le Sprint goal.

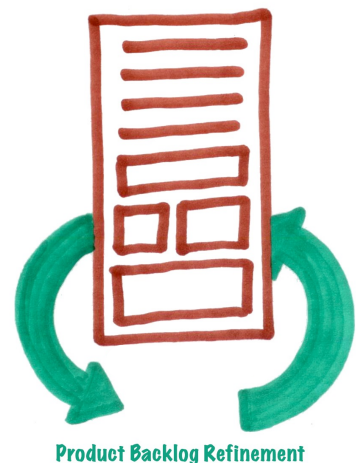
## Sprint Review

A la fin du Sprint, la Scrum Team et les stakeholders intéressés se réunissent pour passer en revue l'Incrément de Produit délivré. La Development Team effectue la démo des éléments du Product Backlog réalisés et interagit avec les stakeholders afin d'avoir du feedback. Le Product Owner accepte le travail terminé ('done').

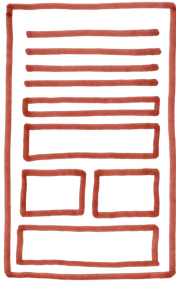
## Sprint Retrospective

Juste après la Sprint Review, La Scrum Team se rencontre afin de discuter de ce qui s'est bien et mal passé durant le Sprint. Ils décident des améliorations les plus importantes à réaliser et définissent des actions concrètes pour réaliser ces améliorations dans le(s) Sprint(s) suivant(s).

Le Product Backlog Refinement n'est pas à durée fixe mais est une activité continue dont le Product Owner a la responsabilité, et qui est exécutée par toute la Scrum Team. Son but est de préparer le Product Backlog pour les Sprints à venir. Cette activité inclut ces différentes actions: ajouter, diviser, détailler, estimer et ordonner des éléments du Product Backlog. Ceci, afin d'être sûr que le Product Backlog soit prêt pour le prochain Sprint Planning.

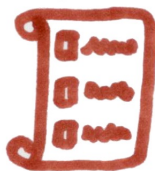


# Scrum: les artéfacts



**Product Backlog**

Une liste ordonnée des éléments du Product Backlog qui représente tous les requirements, besoins et idées pour le produit. Chaque élément doit représenter une partie du produit ayant de la valeur (business). Le Product Backlog est la seule source de travail pour la Development Team.



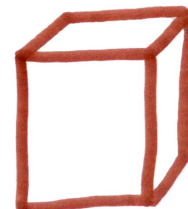
**Definition of Done**

C'est une compréhension commune entre le Product Owner, La Development Team et les stakeholders de ce que signifie le terme 'terminé' (done) pour n'importe quel élément du Product Backlog. Le Definition of Done ne définit pas les critères spécifiques (fonctionnel/technique) d'acceptance pour un élément individuel du Product Backlog, mais est une checklist générale, applicable pour n'importe quel élément du Product Backlog.



**Sprint Backlog**

La liste "raffinée" rassemble les éléments du Product Backlog qui sont sélectionnés pour être délivrés dans le sprint en-cours, et dans le plan de la Development Team pour réaliser le travail. Ce plan est souvent accompagné d'une découpe en tâche des éléments du Product Backlog.



**Product Increment**

Un incrément de produit potentiellement livrable. Chaque Sprint doit se terminer par la livraison d'un nouvel incrément de produit qui est en accord avec le Definition of Done.



**Progress Monitors**

Tout artéfact créé par la Scrum Team pour assurer la transparence sur le statut et l'avancement du produit. Les Progress Monitors classiques incluent des tableaux de tâches et des burndown charts au niveau du Sprint ainsi qu'un release burndown graphs au niveau du produit.

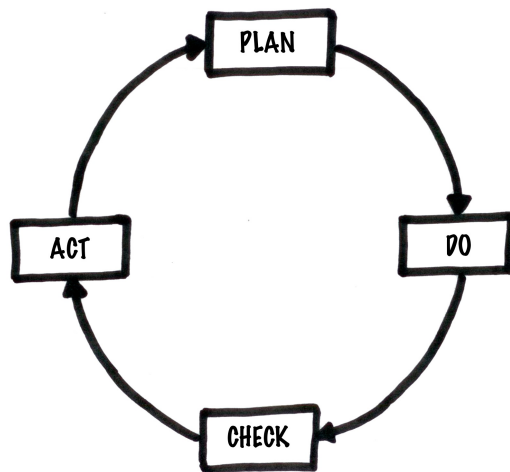


# **Scrum et les principes “Agile”**

# Les principes de Scrum

## “Inspect and adapt” (empirisme)

Scrum est un processus empirique, c’est à dire qu’il déduit la connaissance de l’observation et de l’expérimentation, plutôt que des planning théoriques et des hypothèses. Pour appliquer ce principe **inspect and adapt**, Scrum est basé sur le cycle PDCA cycle, également connu comme la roue de Deming. C’est une méthode itérative qui assure le contrôle et l’amélioration continue quand on gère des processus ou des produits.



✓ **PLAN** On définit l’objectif et les résultats attendus et on décide comment ils seront délivrés (process)

✓ **DO** On réalise le travail en fonction du plan.

✓ **CHECK** On mesure et on analyse les résultats délivrés et on les compare avec le plan initial.

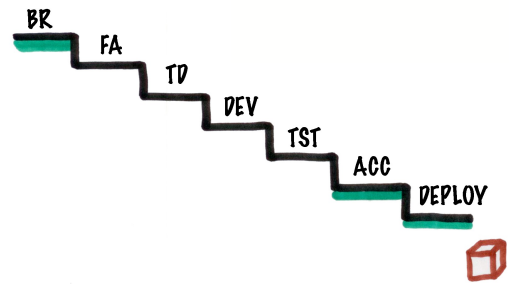
✓ **ACT** Examiner le plan, le résultats et le processus afin de déterminer des améliorations. Application de ces améliorations dans le prochain cycle.

Dans Scrum, chaque Sprint est un cycle complet du PDCA. Les Sprints sont courts et à durée fixe pour s’assurer que les hypothèses émises dans la phase “PLAN” soient confrontées à la réalité le plus tôt et le plus fréquemment possible. Ce qui amène à une amélioration des résultats et de la qualité.

# Un développement itératif et incrémental

## Approche planifiée

Beaucoup d'organisations utilisent une approche traditionnelle, une méthode de gestion qui s'inspire des cycles en V "Waterfall". Ce type de méthode part du principe que les spécifications produites, tout comme les technologies utilisées, sont constantes et prévisibles.

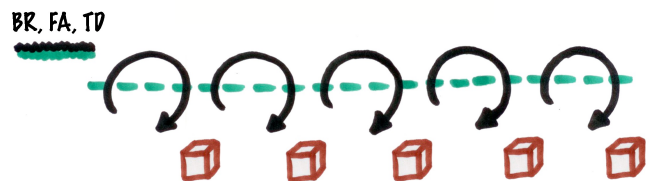


Caractéristiques:

- ✓ Les phases projets sont strictement distinctes. Par exemple : toutes les analyses doivent être finalisées avant que commence le développement, et la réalisation se fait en un seul cycle PDCA.
- ✓ Le client est seulement impliqué très tôt (lors des spécifications) et plus tard (lors des tests de validation) dans le projet
- ✓ Les phases projet à plus haut risque, comme l'intégration et le test sont pris en compte tard dans le projet
- ✓ La valeur métier (le produit final) est délivrée à la fin du projet.

## Adaptatif

Lorsque toutes les dépendances (et liens de cause à effet) ne sont pas prévisibles, il est préférable d'utiliser une approche empirique et adaptative comme Scrum. Chaque Sprint a un cycle de PDCA court qui délivre un incrément de produit, certes limité mais déjà fonctionnel. Cela permet à la Scrum team de vérifier et de s'ajuster fréquemment, en évaluant les incréments de produit et la progression du projet, et en améliorant leur processus à chaque Sprint.



Caractéristiques:

- ✓ Plusieurs cycles PDCA incluent toutes les phases nécessaires (analyse, développement et test) pour délivrer un incrément de produit fonctionnel.
- ✓ Nécessité d'une collaboration continue et quotidienne avec le client
- ✓ Délivrer fréquemment des incréments de produit en apportant rapidement une valeur métier et en augmentant le retour sur investissement.
- ✓ Du fait d'une préparation minimale du plan initial et de ses analyses, la gestion du changement des spécifications produit devient simple et peu coûteuse.

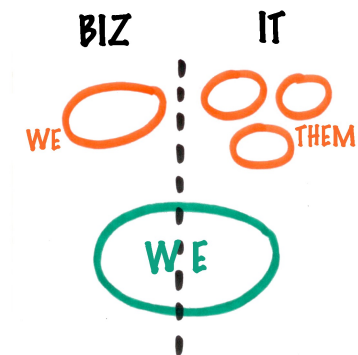
Remarque:



Dans une approche adaptative, il reste important de préparer en amont un plan initial pour le produit. Le facteur clé de succès est de maintenir son plan initial et ses débuts d'analyses, tout en se préparant à des changements futurs.

## L'équipe et la Collaboration

Scrum insiste sur le fait que la Development Team doit avoir toute la connaissance nécessaire et les compétences pour délivrer un incrément de produit 'done' à chaque Sprint. Ce qui signifie que les membres de l'équipe doivent représenter plusieurs départements (aussi bien Métier que Informatique), avec différentes fonctions (aussi bien analystes, qu'ingénieurs, ou testeurs...)



## Auto-Organisation

La Development Team est auto-organisée. Ce qui signifie que les membres de l'équipe doivent être capable de prendre toutes les décisions/actions nécessaires pour réaliser le Sprint, et définir ce qu'il faut délivrer pour l'incrément de produit 'done'. Ni le manager, ni le Scrum Master ne doit faire du management directif, contrôler et commander l'équipe.

L'auto-organisation requiert un espace et des limites claires qui sont définis par le Scrum framework : Le Sprint, la Définition du Done, le Product Backlog, les priorités définies par le Product Owner, les critères d'acceptance...



L'auto-organisation ne signifie pas "la liberté de faire ce que l'on veut", ou "le travail multidisciplinaire bâclé". Elle demande de la rigueur, du respect, de la transparence, de la communication et de la collaboration.

# Les valeurs de Scrum

L'ensemble du travail à réaliser dans le cadre de Scrum nécessite un ensemble de valeurs acceptées par tous et qui vont servir de fondamentaux pour l'équipe, que ce soit dans ses principes ou dans sa manière de travailler. A travers l'esprit d'équipe et l'amélioration continue, Scrum fait non seulement émerger ces valeurs mais repose sur elles. Ces valeurs sont : focus, courage, ouverture, engagement et respect.

## Focus

Parce que nous sommes concentrés et focalisés, nous travaillons uniquement sur peu de choses à la fois, nous travaillons effectivement tous ensemble et nous produisons un résultat de qualité. Par conséquent nous livrons de la valeur plus tôt.

## Ouverture

Lors de notre travail d'équipe, nous nous renvoyons du feedback, tant sur la façon dont on se sent que sur ce qui gêne notre travail. Nous apprenons ainsi, par la pratique, qu'il est bon d'exprimer nos préoccupations pour qu'elles puissent être traitées

## Engagement

Vu que nous avons un grand contrôle sur notre destin, nous devenons plus déterminés à réussir.

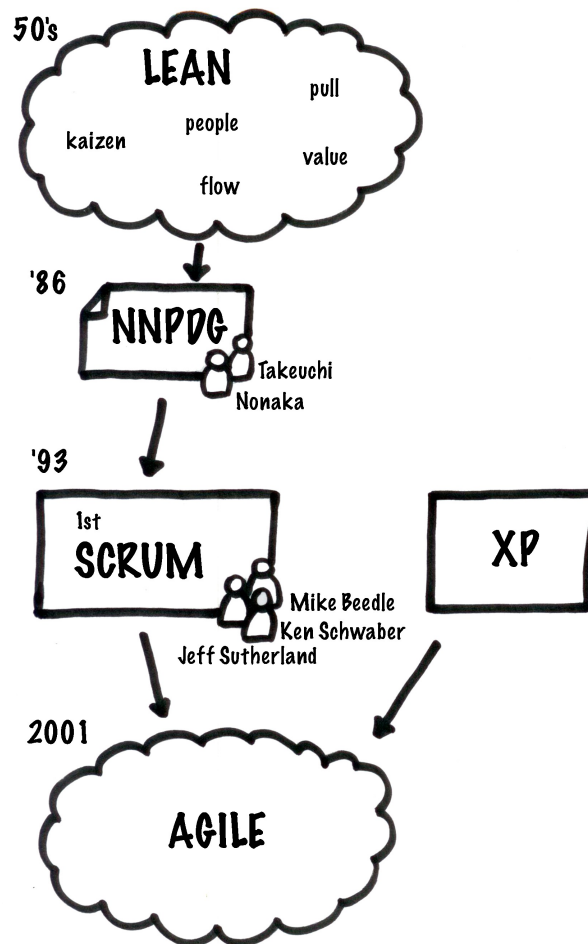
## Respect

En travaillant ensemble, en partageant les succès et les échecs, nous en venons à nous respecter les uns les autres et à devenir nous-mêmes dignes de respect.

## Courage

Nous ne sommes pas seuls et nous sentons que nous avons le support nécessaire, ce qui nous donne plus de ressources à notre disposition. Cela nous donne le courage de relever de plus grands défis.

# Historique de Scrum



L'histoire de Scrum a démarré en 1986 avec un article de la Harvard Business Review intitulé "The New Product Development game" (NNPDG) par Takeuchi and Nonaka. Cet article reposait sur les principes lean développés dans le monde de la construction automobile (Toyota). Cette philosophie (Lean) a été introduite par Toyota avec la Toyota Production System comme une alternative aux systèmes de production de masse en cours à l'époque.

Dans l'article NNPDG une comparaison est établie entre une course relais, le développement de produits et le rugby. L'article décrit comment se passer régulièrement la balle de l'un à l'autre permet à l'équipe d'atteindre son but et peut correspondre mieux aux exigences actuelles du marché.

C'est seulement en 1993 que Jeff Sutherland et son équipe basés à Easel ont appliqués et traduits ces idées au monde de l'informatique créant ainsi le premier Scrum. Ken Schwaber et Mike Beedle ont documentés le process Scrum dans le livre "Agile Software Development with Scrum".

En 2001 un ensemble de leaders reconnus de l'industrie informatique se sont rencontrés lors d'une conférence dans l'Utah pour discuter de méthodes de développement plus légères. Ils ont introduits le terme Agile et ont créé le Manifeste Agile pour le développement Logiciel qui définit leurs valeurs et principes communs.

# Agile

Le Manifeste Agile définit un ensemble de valeurs et principes. Voir [www.agilemanifesto.org](http://www.agilemanifesto.org).

**Les individus et leurs interactions**  
plus que les processus et les outils

**Des logiciels opérationnels**  
plus qu'une documentation exhaustive

**La collaboration avec les clients**  
plus que la négociation contractuelle

**L'adaptation au changement**  
plus que le suivi d'un plan

1. Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.
2. Accueillez positivement le changement des besoins, même tard dans le projet. Les processus Agile exploitent le changement pour donner un avantage compétitif au client.
3. Livrez fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.
4. Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.
5. Réalisez les projets avec des personnes motivées. Fournissez-leur l'environnement et le soutien dont ils ont besoin et faites-leur confiance pour atteindre les objectifs fixés.
6. La méthode la plus simple et la plus efficace pour transmettre de l'information à l'équipe de développement et à l'intérieur de celle-ci est le dialogue en face à face.
7. Un logiciel opérationnel est la principale mesure d'avancement.
8. Les processus Agile encouragent un rythme de développement soutenable. Ensemble, les commanditaires, les développeurs et les utilisateurs doivent être capables de maintenir indéfiniment un rythme constant.
9. Une attention continue à l'excellence technique et à une bonne conception renforce l'Agilité.
10. La simplicité – c'est-à-dire l'art de minimiser la quantité de travail inutile – est essentielle.
11. Les meilleures architectures, spécifications et conceptions émergent d'équipes auto-organisées.
12. A intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence.

# **Le Product Backlog et les user stories**



# Un Product Backlog ‘complet’

Le Product Owner doit s'assurer que le Product Backlog est à jour et complet à tout moment. Cela permet à la Development Team de sélectionner les bons (et à plus forte valeur) éléments du Product Backlog qui seront ajoutés et réalisés dans le prochain Sprint.

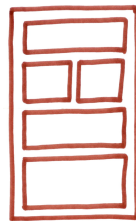
Un bon Product Backlog est DEEP:

## D ÉTAILLÉ CORRECTEMENT



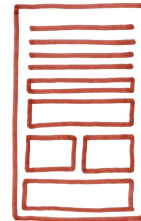
**Product Backlog**

Des éléments trop petits et trop détaillés ne donnent pas de vue d'ensemble et compromettent une bonne analyse fonctionnelle



**Product Backlog**

Les éléments de base sont bien divisés mais ne peuvent être inclus dans un Sprint, ils sont trop petits



**Product Backlog**

Bon niveau de détail : les 1ers éléments sont petits et facilement inclus dans un Sprint, les autres sont moins prioritaires et donc plus grands, ils seront analysés plus tard

## E VOLUTIF

Le Product Backlog est un document évolutif qui est mis à jour continuellement et amélioré pour définir un produit à forte valeur et pour fournir une information correcte, au bon moment.

## E STIMÉ

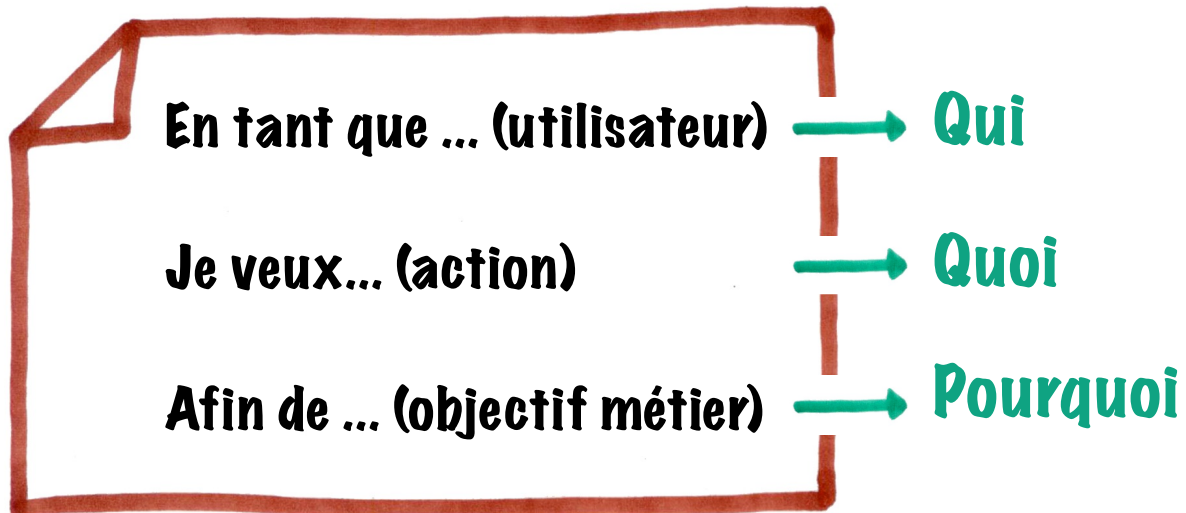
La taille et le coût de tous les éléments du Product Backlog doivent être estimés pour permettre au Product Owner de prendre de bonnes décisions en fonction des priorités et du planning des projets/produits. Cela permet aussi de s'assurer que les 1ers éléments sélectionnés sont suffisamment petits pour être inclus dans un Sprint.

## P RIORISÉ

A tout moment, le Product Owner doit s'assurer que le Product Backlog est en ordre pour construire un produit qui aura la plus forte valeur possible

# User stories

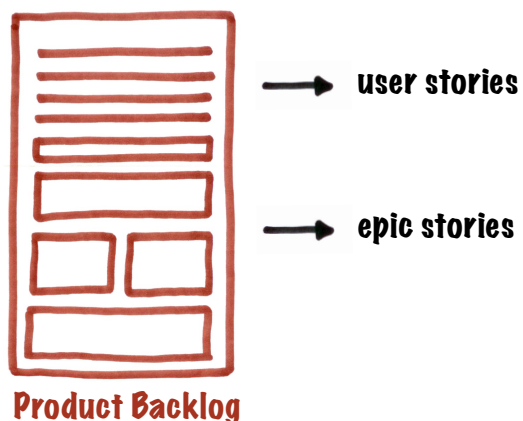
La Scrum Team utilise les éléments du Product Backlog, comme des User stories. Ce sont les spécifications des utilisateurs ou les caractéristiques du produit.



Objectifs des User stories :

- ✓ Décrire les caractéristiques produit ou les spécifications de l'utilisateur
- ✓ Partager la même compréhension et se mettre d'accord entre le business/ client et la Development Team.
- ✓ Un point de départ pour commencer les discussions sur les analyses et le design
- ✓ Un élément de planification.

## Epic versus user stories



Le format de la User Story peut être appliqué à tout niveau de détails pour décrire les spécifications, qu'elles soient globales ou détaillées. Les User Stories large sont appelées des Epic Stories, qui demandent ensuite à être divisées en des User Stories petites et détaillées, afin qu'elles soient mises dans un Sprint. Un bon Product Backlog contient des petites User Stories (qu'on peut intégrer dans un Sprint) mais aussi des Epic Stories (qu'on ne peut pas encore intégrer dans un Sprint).

# INVESTir dans les User Stories

When writing user stories, it is good to keep the INVEST acronym in mind to come to good, well-formulated user stories. Lors de la réflexion et l'écriture, il est bon de retenir l'acronyme INVEST pour bien formuler les User Stories.

**I****NDEPENDANT** Les User Stories doivent être un maximum indépendantes les unes des autres pour permettre une meilleure flexibilité dans le planning et l'implémentation

**N****EGOCIABLE** Une bonne User Story n'est pas un contrat détaillé, mais favorise la co-création et la négociation dans le détail. Il doit comprendre l'essentiel qui permet l'estimation et le planning.

**V****ALEUR** Une User Story doit fournir une valeur business, ce qui signifie qu'elle doit apporter de la valeur pour les utilisateurs ou pour les clients du produit.

**E****STIMABLE** La Development Team doit être capable d'estimer toute User Story. C'est nécessaire à la planification mais aussi pour vérifier la compréhension de tous dans son niveau de détails

**S****MALL** Small (Petit) ou au bon niveau de détail. Une User Story qui s'apprête à être inclus dans un Sprint doit être petite pour que plusieurs d'entre elles soient ajoutées. Les Epic Stories peuvent être plus grandes, leur taille dépendent de leur importance et leur priorité (urgence et risque)

**T****ESTABLE** Une User Story est bonne lorsque toutes les personnes impliquées comprennent comment elle sera testée et acceptée. C'est le critère d'acceptance qu'on ajoute à la User Story qui est le critère de succès.

# Les pièges des User Stories

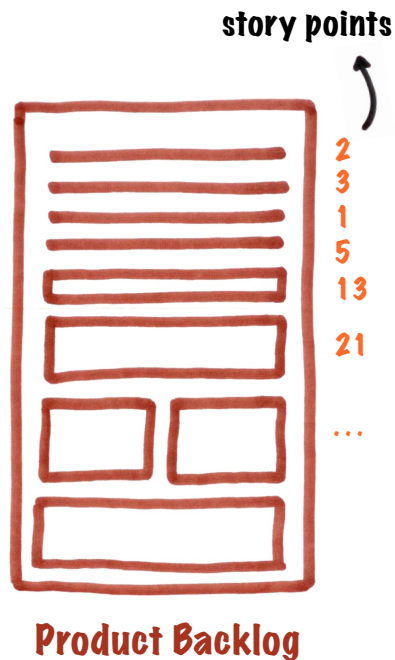
Parallèlement aux critères INVEST que l'on a évoqué, voici une série de pièges classiques lors de l'écriture des User Stories :

- ✓ **TROP DE DETAILS OU ORIENTEE-SOLUTION** La User Story contient trop de détails techniques ou fonctionnels, la solution est décrite plutôt que le problème métier (le besoin ou l'objectif)
- ✓ **UTILISATEUR MAL DEFINI** L'utilisateur est défini de façon trop général, ne représente pas un persona, ou bien la User Story est définie d'un point de vue technique ou applicatif
- ✓ **OBJECTIF METIER VAGUE** Le besoin ou l'objectif métier est manquant ou décrit vaguement. Il se rapproche plutôt de 'QUOI' que 'POURQUOI', souvent caché derrière la formulation 'Je veux X pour que X'
- ✓ **PAS DE CRITERES D'ACCEPTANCE** Aucun critère d'acceptance (conditions du niveau de satisfaction) n'est ajouté dans la User Story alors qu'on se rapproche de la fin du Sprint Planning.

# **Agile: estimation et planning**

# Relative estimations

Bien qu'elles ne soient pas définies dans le Core Scrum, la plupart des Scrum Teams utilisent les estimations relatives pour évaluer le coût des éléments du Product backlog. Ce qui signifie :



- ✓ Les estimations sont seulement réalisées par la Development Team, en tenant compte des connaissances de tous les membres, ils s'engagent dans ces estimations.
- ✓ Les estimations sont formulées en (user) Story points qui sont des nombres relatifs et comparables

0 1 2 3 5 8 13 21 ...

- ✓ La suite de Fibonacci (ou similaire) est utilisée comme seules valeurs possibles pour les estimations; ce qui rend l'exercice incertain lors d'estimation de plus grands éléments.
- ✓ La Development Team se met d'accord sur un élément (petit et clair) du Product Backlog à utiliser comme référence (le '1') qui sera le point de comparaison.

## Estimations relatives vs. absolues

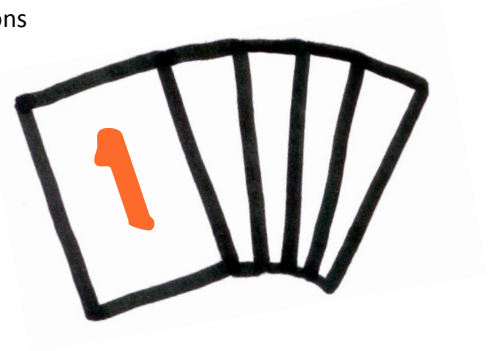
Pourquoi ne pas estimer en jours/homme ou en temps ? Car plusieurs études démontrent que des estimations relatives sont :

- ✓ **plus rapide**
- ✓ **plus précise**

Par ailleurs, les estimations relatives sont indépendantes de la personne qui se chargera plus tard de l'exécution du travail. Ce qui évite le blâme et permet à la Development Team de s'auto-organiser dans le Sprint

## Planning poker

Le Planning Poker est une technique efficace pour les estimations relatives, car il évite d'influencer les membres et conduit à de meilleures estimations. À côté de cela, il est un moyen très efficace pour mettre en œuvre le principe Lean de « décider au dernier moment opportun » car il permet à l'équipe de discuter des éléments du Product Backlog avec un bon niveau de détail pour obtenir des estimations. Cela permet à l'équipe de poser les bonnes questions adéquates pour prendre les décisions fonctionnelles et techniques nécessaires à ce moment (et pas plus loin) du projet.



### Comment cela fonctionne t-il ?

- ✓ Le Product Owner explique brièvement l'élément du Product Backlog
- ✓ Chaque membre d'équipe compare cet élément avec l'élément de référence (le '1') et choisit sa carte numérotée, sans la montrer aux autres.
- ✓ Lorsque tout le monde a choisi, les cartes sont retournées au même moment.
- ✓ Si nécessaire, les personnes ayant l'estimation la plus haute et la plus basse expliquent leur raisonnement
- ✓ L'équipe se met d'accord sur l'estimation, éventuellement avec un autre tour de table

# Vélocité

$$\text{Vélocité} = \text{story points DONE} / \text{Sprint}$$

La vélocité est le point de mesure pour évaluer la charge de travail réalisée dans un Sprint par a Scrum Team. La vélocité est exprimée en story points. Le total des story points égale tous les éléments du Product Backlog que l'équipe a délivré selon la définition du Done pour un Sprint.

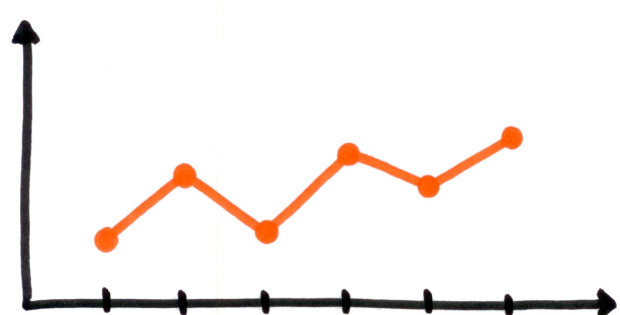
La Vélocité est utilisée :

- ✓ **Par la Development Team:** Lors du Sprint Planning en tant qu'un des critères pour décider de la charge de travail qui sera mise dans un Sprint. Lors du Sprint Retrospective pour entamer les discussions sur y-une meilleure efficacité.
- ✓ **Par le Product Owner:** La vélocité de l'équipe (c'est à dire la moyenne des derniers x Sprints) peut être utilisée pour les prévisions des mises en production futures (release planning)

La vélocité est toujours relative à la durée d'un Sprint et à la composition de l'équipe. La vélocité n'est pas un outil de management pour comparer les équipes entre elles ou pour mesurer la productivité.

## Graphique Vélocité

Les Scrums Teams visualisent leur vélocité dans ce graphique. Cela permet de repérer les tendances et de discuter des choses telles que la stabilité de la vélocité (prévisibilité) et la manière d'accroître la vélocité.

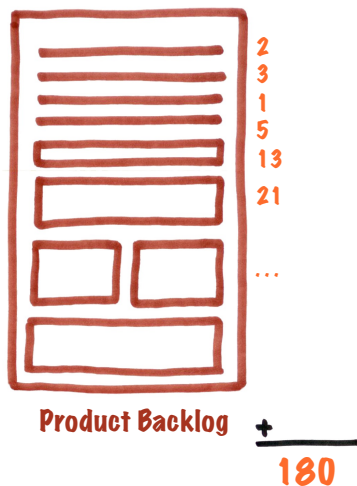




# Représentation graphique : Burndown chart

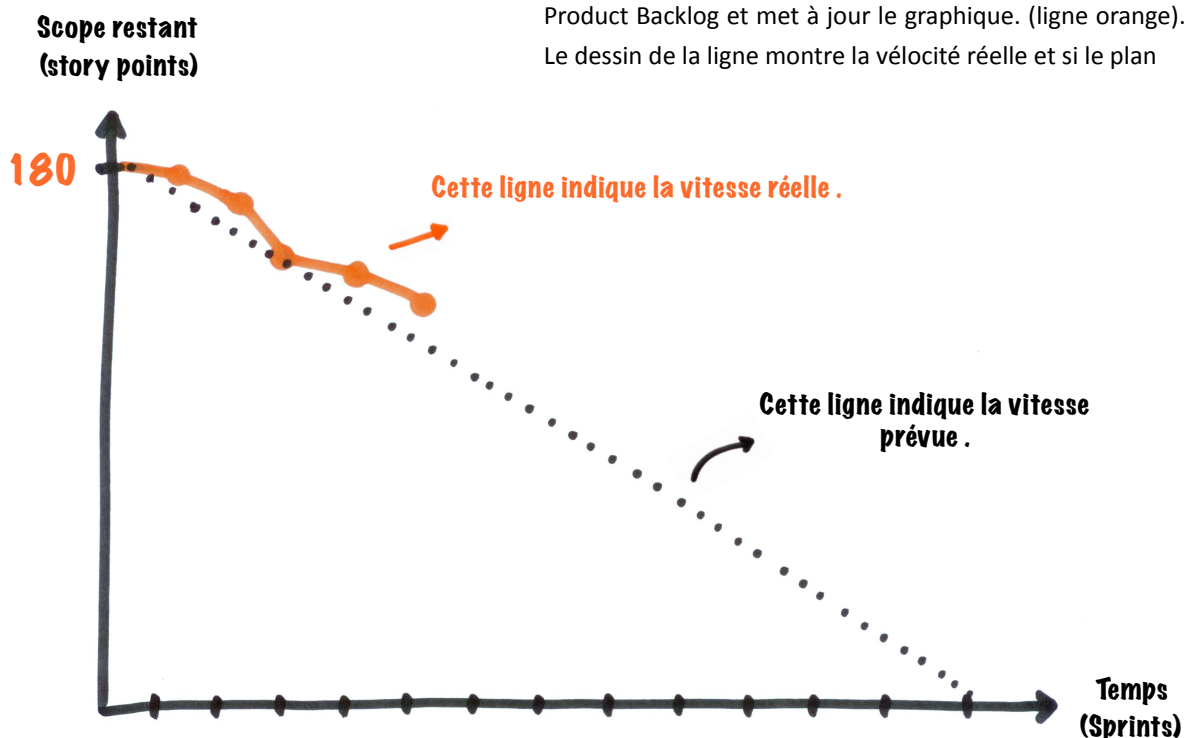
Le Product Owner est responsable pour la mise en production (release planning) : Définir et Prévoir les mises en production et le scope associé, le temps et le budget, ainsi que le suivi des mises en production durant le développement du produit. Un des outils le plus important du Product Owner est le « release burndown graph », traduisible comme une représentation graphique du travail restant à faire dans une période, et actualisé aussi souvent que possible pour montrer une tendance de l'avancement

Ce graphique montre le scope qui reste (en story points) versus le temps. Ce qui permet de visualiser si l'équipe est en bonne voie (ou non) de délivrer en fonction du plan initial.



Pour réaliser ce graphique, le Product Owner:

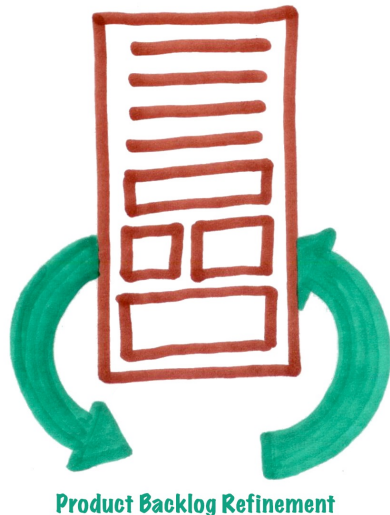
- ✓ Demande à la Development Team d'estimer tous les éléments du Product Backlog qui feront partie de la mise en production.
- ✓ Ajoute les estimations pour obtenir le total des story points dans la mise en production (180 dans notre exemple)
- ✓ Etablit la vélocité prévue sur base de la vélocité mesurée (par ex . la moyenne des derniers 3-4 vélocités). En l'absence de données disponibles, il demande à la Development Team la vélocité initiale prévue. Il prend en compte également la durée Sprint et l'équipe
- ✓ Dessine la progression attendue, sur base du scope, budget et temps (les pointillés sur notre exemple)
- ✓ Après chaque Sprint, il mesure le scope qui reste dans le Product Backlog et met à jour le graphique. (ligne orange). Le dessin de la ligne montre la vélocité réelle et si le plan



# Scrum: les Rituels

# Redéfinition du Product Backlog

Le but de la redéfinition du Product Backlog est de permettre à celui-ci d'être prêt pour les Sprint(s) à venir. La redéfinition du Product Backlog est une action récurrente dont le Product Owner est responsable et qui doit être réalisée par toute la Scrum team. C'est une collaboration entre le Product Owner et la Development Team qui inclus :



- ✓ Ajout et retrait des éléments du Product Backlog.
- ✓ Division des éléments trop grands (epic stories) en éléments plus petits et réalisables (epic stories ou sprintable user stories).
- ✓ Détail des produits du Product Backlog et critères d'acceptance.
- ✓ Définir les priorités du Product Backlog
- ✓ Estimation des coûts/ charge de travail pour les éléments du Product Backlog

Le Product Owner s'assure de la redéfinition du Product Backlog afin que celui-ci soit prêt pour le prochain Sprint Planning.

## Ordonner le Product Backlog

Lors de la définition des priorités du Product Backlog, le Product Owner tient compte des critères suivants :

- ✓ **LE CLIENT ou LA VALEUR METIER** Bien qu'il soit difficile à mesurer ou à quantifier, le Product Owner doit trouver des critères pour estimer la valeur métier de chaque élément du Product Backlog. Ce sont ces estimations qui doivent permettre de définir les priorités du Product Backlog pour atteindre la vision à long terme.
- ✓ **RISQUE** Le Product Owner peut inclure des éléments à risque en priorité dans le Product Backlog, ceci afin de limiter l'impact d'autres risques possibles (technique ou métier) ou pour évaluer un risque potentiel.
- ✓ **COUTS** En plus de la valeur métier, le coût des éléments est à considérer lorsqu'on définit les priorités, afin de garantir un retour sur investissement maximal.
- ✓ **DEPENDENCES** Le Product Owner doit considérer les dépendances techniques ou métier lors de la définition des priorités.

# Sprint Planning

Le but du Sprint Planning pour la Scrum Team est de prévoir et planifier le Sprint. La Development Team décide des éléments à prendre du Product Backlog tout en tenant compte des priorités du Product Owner. Ils s'accordent sur ce qui doit être fait.

Un 'Product Backlog' disponible signifie :

## INPUT

- ✓ que les priorités ont été définies par le Product Owner, sur base de la valeur métier, du risque, du coût et des dépendances.
- ✓ que les 1er éléments de la liste sont petits, réalisables dans un Sprint (INVEST).
- ✓ que les 1er éléments de la liste sont estimés par la Development



**Durée:** Dès le début du Sprint total, durée maximum de 2h/semaine de Sprint

**Approche:**

- 1) Le Product Owner explique les éléments du Product Backlog et la **Development Team décide sur quels éléments ils peuvent s'engager à délivrer sur ce Sprint, en respectant le concept 'Definition of Done'**. Les priorités sont discutées par la Scrum Team, ainsi que les dépendances, et le meilleur choix sur les éléments à mettre dans le Sprint.
- 2) La Development Team découpe les **éléments en tâches réalisables pour s'auto-organiser et planifier le travail lors du Sprint**. Les analyses techniques, fonctionnelles, le design et les prises de décision doivent être faites pour commencer le Sprint. Le Product Owner reste à disposition de la Development team en cas de question lors de la 2ème partie du Sprint Planning.



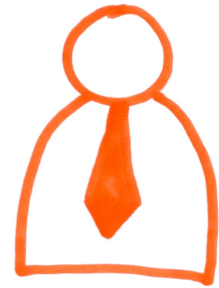
Le Sprint Backlog contient :

## OUTPUT

- ✓ Les prévisions de l'équipe pour les éléments du Product Backlog qui seront délivrés lors de ce Sprint.
- ✓ Une découpe des tâches pour chacun de ces éléments et le plan pour réaliser le travail

# Rôle du Product Owner lors du Sprint Planning

La chose la plus importante pour un Product Owner concernant le Sprint Planning est de venir préparé. Ensemble avec la Development Team, il doit avoir revu et détaillé le Product Backlog pour que celui-ci soit prêt lors du Sprint Planning.



## Rôle du Product Owner lors du Sprint Planning

- ✓ Explains the Product Backlog items and his decisions on the order/priority.  
Expliquer les éléments du Product Backlog et ses décisions sur l'ordre de priorité.
- ✓ Answers the team's functional and business questions about the items. Répondre aux questions fonctionnelles et métier concernant les éléments.
- ✓ Se mettre d'accord avec l'équipe sur les changements possibles en respectant l'ordre des éléments du Product Backlog. Ce qui permet de considérer les dépendances, la composition de l'équipe, les compétences etc... afin de que la Development Team puisse maximiser la valeur produite dans le Sprint.
- ✓ Revoir les critères d'acceptance des éléments du Product Backlog avec la Development Team si besoin

De manière usuelle, c'est le Product Owner qui prend le lead au démarrage du Sprint Planning, en expliquant les 1ers éléments du Product Backlog et décidant conjointement avec la Development Team de ce qui sera inclus dans le Sprint.

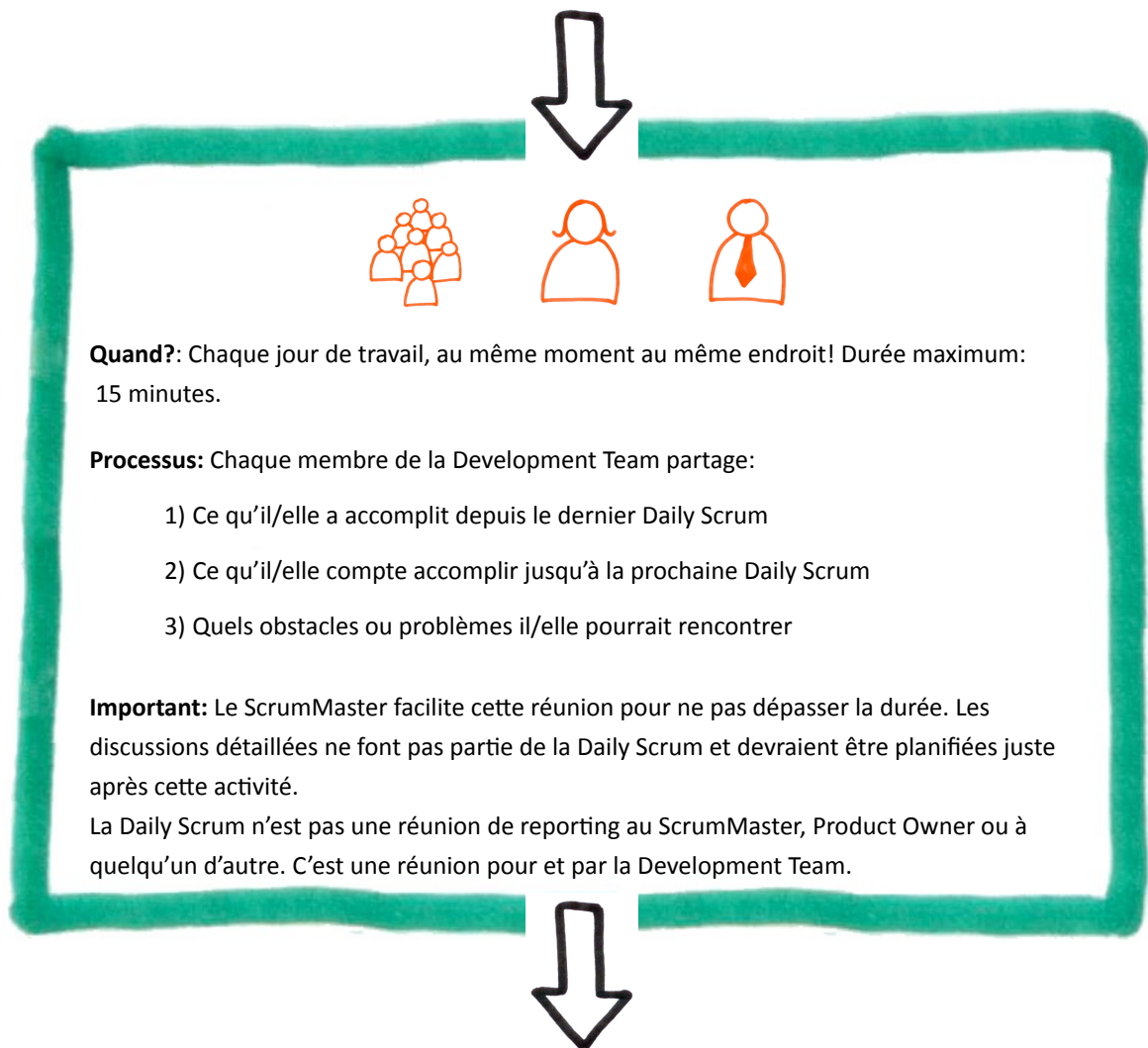
Lorsque la Development Team commence à décomposer et planifier, le Product Owner reste à l'écoute et disponible pour l'équipe afin de répondre aux questions.

# Daily Scrum

La Daily Scrum est l'occasion, pour la Development Team, de discuter en toute transparence de l'avancement du Sprint, des éventuels obstacles et de ce qui est nécessaire pour atteindre possible l'objectif (goal) du Sprint.

## INPUT

- ✓ Les expériences et le progrès de ces 24 dernières heures de travail. Souvent cette activité ne nécessite pas d'input physique (documents etc.), bien que les équipiers auront préparé cette activité avant de s'y rendre.



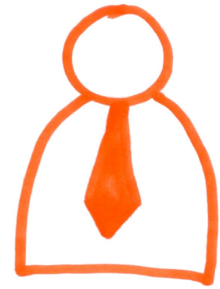
## OUTPUT

- ✓ De la transparence sur qui fait quoi et au sujet du progrès (et des obstacles) concernant l'objectif du Sprint. Le Sprint Backlog est mis à jour.
- ✓ Engagement au sein de la Development Team sur comment aborder le prochain jour de travail.

## Rôle du Product Owner dans la Daily Scrum

La présence du Product Owner n'est pas obligatoire mais utile pour :

- ✓ Connaître l'état d'esprit de l'équipe
- ✓ Connaître les questions que se posent la Development Team et aider en cas d'impondérables
- ✓ Partager brièvement avec l'équipe l'avancement de ses activités et se mettre d'accord sur les prochaines sessions/ actions du Product Backlog à détailler.



# Sprint Review

Le but du Sprint Review est de rassembler la Scrum Team et les stakeholders intéressés pour revoir les incréments de produit. Le meeting consiste à valider et prendre connaissance des incréments de produit, et mieux comprendre la façon de faire (au niveau produit)

## INPUT

- ✓ L'incrément de produit qui est délivré durant le Sprint.
- ✓ Vérifier si les prévisions faites des éléments du Product Backlog ont été réalistes (en respectant le Definition of Done) à la fin du Sprint



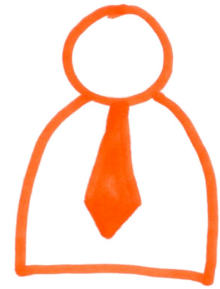
## OUTPUT

- ✓ Le plus important : L'incrément de Produit doit être terminé 'Done' !
- ✓ Considérer le feedback par les stakeholders qui apporte de nouvelles idées au Product Owner et de futurs éléments à ajouter dans le Product Backlog
- ✓ Mieux comprendre comment les clients s'approprient le produit, afin d'améliorer les prises de décisions par la Development Team
- ✓ La vélocité du Sprint



## Rôle du Product Owner dans le Sprint Review

- ✓ Inviter les acteurs principaux (métier) du projet : clients, utilisateurs, sponsor, marketing, commercial...
- ✓ Si nécessaire, rappeler sa vision et ses décisions aux acteurs principaux du projet
- ✓ Acter (Accepter/ Rejeter) de façon formelle les incréments du Produit, sur base des critères d'acceptance et du Definition of Done
- ✓ Rassembler le feedback reçu des acteurs principaux du projet et de la Development Team pour mettre à jour le Product Backlog (si applicable)



De manière usuelle, le Product Owner prend le lead sur cette réunion, jouant un rôle de facilitateur dans les discussions et en donnant du soutien/ support à la Development Team sur les incréments du produit délivrés.

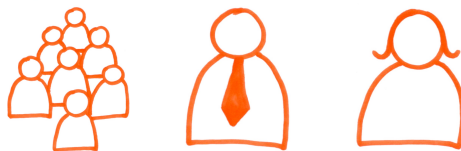
Un bon Product Owner collabore avec la Development Team durant le Sprint pour obtenir un maximum de valeur lors de la Sprint Review. Il est préparé pour le Sprint Review et connaît d'avance les résultats des tests des incréments du Produit, permettant ainsi l'acceptation ou le rejet des éléments.

# Sprint Retrospective

Le but du Sprint Retrospective est de tenir compte du Sprint qui vient de s'achever, ses process, la collaboration, les outils, la productivité et l'humeur de l'équipe. Ensuite, il faut mettre en place un plan pour améliorer les choses dans les prochains Sprint(s)

## INPUT

- ✓ Considérer l'expérience et la connaissance du Sprint qui vient de s'achever
- ✓ Tenir compte du feedback du Sprint Review
- ✓ Les prévisions et la vélocité du Sprint achevé
- ✓ Les plans d'amélioration précédents et leur statut actuel



**Durée:** En fin de Sprint, juste après la Sprint Review, durée maximale d'1h/semaine et par Sprint

**Approche:** Différents formats et techniques sont utilisés pour rassembler l'information, définir les zones d'amélioration, les priorités, analyser les causes et se mettre d'accord sur les actions à prendre.

**Important:** Le ScrumMaster doit s'assurer que le moment est bien choisi et au bon endroit pour discuter ouvertement de tout sujet



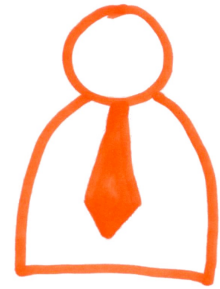
## OUTPUT

- ✓ Le plan d'amélioration doit être réaliste, et validée, il contient des actions et l'équipe marque son accord pour les prochains Sprint(s)

## Rôle du Product Owner dans la Sprint Retrospective

Le Product Owner participe de la même façon que la Development Team et au même niveau.

Si la relation de travail avec le Product Owner est mauvaise, il peut être décidé par la Development Team et le ScrumMaster de ne pas convier le Product Owner. En effet, il est plus important de maintenir un environnement qui va permettre l'échange sans tabou. Cependant, il appartient au ScrumMaster de considérer le problème et de maintenir la bonne relation avec le Product Owner.



If the team wants to dedicate a Sprint Retrospective to eg. a technical issue in which the Product Owner cannot help, they might of course agree with the Product Owner not to join the meeting.

Dans le cas où l'équipe souhaite adresser des sujets en Sprint Retrospective qui ne concernent pas le Product Owner (par ex: problème technique...), il est décidé en commun accord avec le Product Owner que celui-ci ne participe pas