

Scrum

Scrum Master et Product Owner

Table des matières

Cadre de Scrum	3
Qu'est-ce que Scrum ?.....	4
Aperçu du cadre de Scrum	5
Rôles de Scrum.....	6
Rituels de Scrum	7
Artéfacts Scrum.....	9
Scrum et principes Agile	10
Principes de Scrum.....	11
Valeurs de Scrum	15
Histoire de Scrum.....	16
Agile	17
Rôles de Scrum	16
Product Owner (PO)	17
Scrum Master.....	18
Équipe de développement.....	19
Les rituels de Scrum	20
Raffinage du Backlog de produit.....	21
Planification du sprint	22
Scrum quotidien (stand-up meeting).....	24
Revue de Sprint ou Démo	26
Rétrospective du sprint	29
Backlog de produit et User Stories	31
Un Backlog de produit "prêt"	32
User Stories	33
INVESTissez dans vos User Stories.....	34
Les pièges des User Stories	35
Estimation et planification Agile.....	35
Estimations relatives	36
Vélocité	38
Le graphique d'avancement de la version (release burndown graph)	39

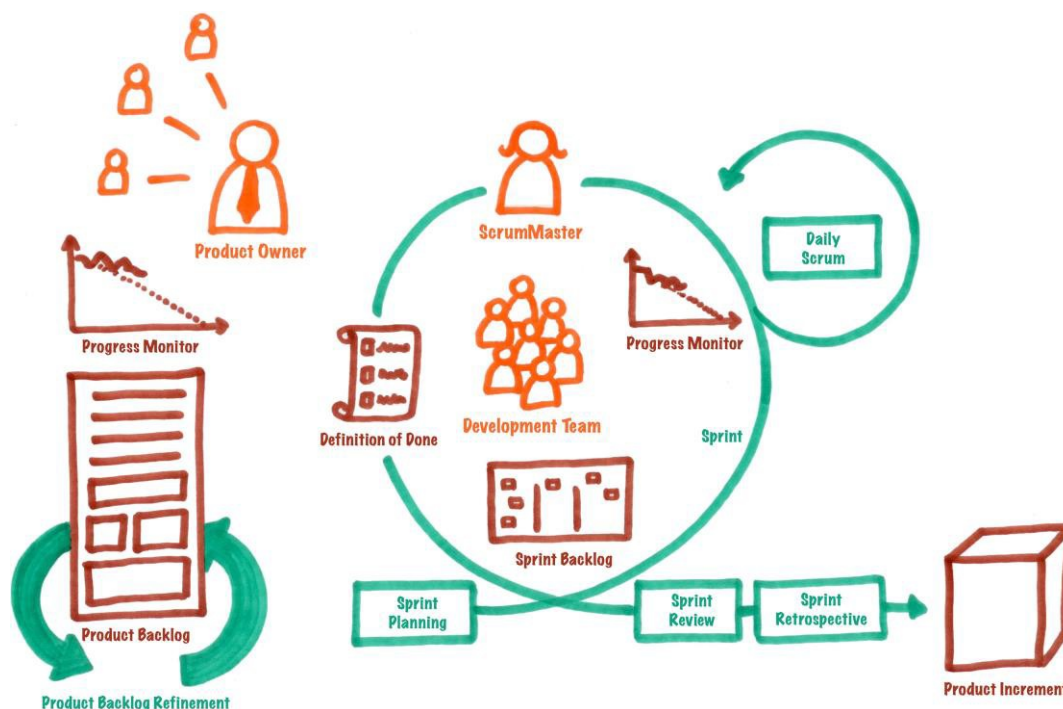
Cadre de Scrum

Qu'est-ce que Scrum ?

Scrum est un **cadre** pour la conception de produits ou de services. Le but de Scrum est de le faire avec une équipe **auto-organisée** en utilisant des itérations courtes et time-boxées (géré dans un bloc de temps limité) qui délivrent chacune un incrément potentiellement livrable du produit. De cette façon, Scrum maximise le contrôle, la flexibilité, la qualité et le retour sur investissement.

Le cadre Scrum définit un ensemble limité et bien équilibré de rôles, de rituels et d'artefacts qui fonctionnent ensemble selon certaines règles. À côté de cela, Scrum exige, et au même moment permet, un ensemble spécifique de valeurs et de principes.

Aperçu du cadre de Scrum



Scrum commence par le besoin des parties prenantes d'une organisation à fournir un certain produit ou service. Le **Product Owner** élabore une vision du produit et rassemble les objectifs et les exigences du produit dans le **Product Backlog**. Il est responsable de la maximisation de la valeur livrée du produit, en tenant compte du temps, du budget et d'autres contraintes. Il le fait en **raffinant** de manière continue le **Backlog de produit** avec l'équipe de développement.

L'**équipe de développement** est une équipe interfonctionnelle et auto-organisée qui fournit des incréments de valeur du produit/service, pouvant être livrés, par itérations courtes et cadencées appelées **Sprints**. Le **ScrumMaster** agit comme un guide et un coach pour l'équipe et l'organisation, les aidant à bien mettre en œuvre Scrum et à éliminer tous les obstacles. Le Product Owner, le ScrumMaster et l'équipe de développement sont appelés ensemble l'**équipe Scrum**.

Un Sprint commence par la **planification du Sprint**. Il s'agit d'un rituel géré par bloc de temps (timebox) dans lequel l'équipe de développement tire le travail du Backlog de produit dans le Sprint, en tenant compte des priorités du Product Owner. Le résultat de la planification du Sprint est le **Sprint Backlog**, qui est un plan pour le Sprint qui contient les prévisions de l'équipe sur l'**incrément de produit qui sera livré** dans ce Sprint. L'équipe Scrum utilise la "Definition of Done" pour planifier le travail et pour évaluer si un élément du Backlog de produit est vraiment fait et fournit une valeur commerciale.

Chaque jour, l'équipe Scrum se réunit pour une courte réunion, appelée **Scrum quotidien**, afin de vérifier l'avancement, d'éliminer les problèmes et de s'assurer que l'objectif du sprint est atteint. L'équipe Scrum utilise des **moniteurs de progression** pour suivre et visualiser l'avancement de son travail.

À la fin du sprint, l'équipe Scrum organise une réunion de **révision du sprint** pour présenter l'incrément de produit au Product Owner et aux parties prenantes. Et pour générer du feedback sur le produit et discuter de l'avancement du produit. Après cette réunion, l'équipe Scrum fait le point sur le sprint et évalue son processus dans la **rétrospective du sprint**, ce qui aboutit à un plan d'amélioration actionnable pour le prochain sprint.

5

Rôles de Scrum

Scrum définit trois rôles différents : le Product Owner, le ScrumMaster et l'équipe de développement. Ensemble, ils sont appelés l'équipe Scrum.



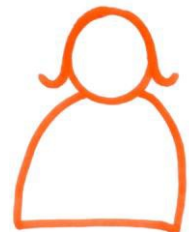
**Product Owner
(PO)**

- ✓ Responsable de la maximisation de la valeur du produit dans le respect des contraintes de temps et de budget.
- ✓ Représenter les clients, les utilisateurs et les parties prenantes.
- ✓ Gérer et ordonner le Backlog de produit.



**Équipe de
développement**

- ✓ Une équipe pluridisciplinaire comprenant toutes les personnes nécessaires (compétences et connaissances) pour livrer les incréments de produit "faits".
- ✓ Une équipe auto-organisée
- ✓ Responsable de la réalisation de l'objectif du sprint et de la livraison d'un incrément de produit "terminé".



Scrum Master

- ✓ Un leader serviteur qui soutient l'équipe Scrum et l'organisation dans la mise en œuvre de Scrum.
- ✓ Chargé de veiller à ce que les obstacles soient signalés et résolus dès que possible.

Rituels de Scrum

Scrum définit quatre cérémonies :

Planification du sprint

Au début d'un Sprint, l'équipe Scrum se réunit pour prévoir et planifier le Sprint. Les membres de l'équipe décident de la nature et de la quantité de travail à transférer du Backlog de produit au Backlog de sprint et se mettent d'accord sur la manière de réaliser ce travail.

Scrum quotidien

Chaque jour, les membres de l'équipe de développement se réunissent pour une courte réunion (15 minutes maximum). Ils discutent de la progression du Sprint, des obstacles éventuels et de ce qui est nécessaire pour atteindre l'objectif du Sprint.

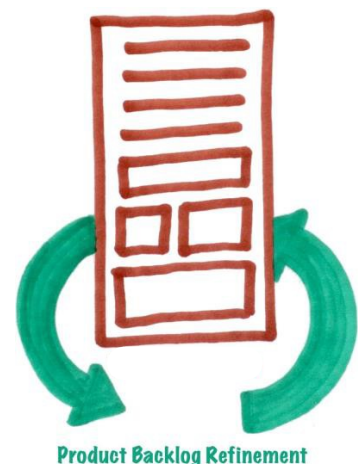
Revue du Sprint

À la fin du sprint, l'équipe Scrum et les parties prenantes intéressées se réunissent pour examiner l'incrément de produit livré. L'équipe de développement fait la démonstration des éléments du Backlog de produit et interagit avec les parties prenantes pour obtenir des commentaires. Le Product Owner valide et accepte le travail "terminé".

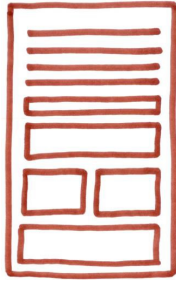
Rétrospective du sprint

Juste après la revue du Sprint, l'équipe Scrum se réunit pour discuter de ce qui s'est bien et mal passé pendant le Sprint. Ils se mettent d'accord sur les améliorations les plus importantes et définissent les actions et accords concrets pour les exécuter dans le(s) prochain(s) Sprint(s).

Le raffinement du Backlog de Produit n'est pas une cérémonie, mais une activité continue appartenant au Product Owner, mais exécutée par la Scrum Team au complet. Il sert à préparer le Backlog de produit pour le(s) sprint(s) à venir. Cela comprend l'ajout, le découpage, le détail, l'estimation, l'ordre, etc. des éléments du Backlog de produit pour s'assurer que le Backlog de produit est prêt pour la planification du prochain Sprint.

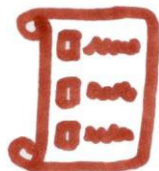


Artéfacts Scrum



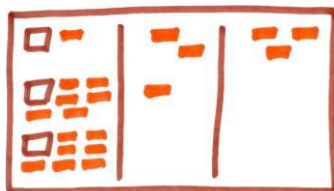
Backlog de produit

Une liste ordonnée d'éléments du Backlog de produit qui représentent toutes les exigences, les besoins et les idées pour le produit. Chaque élément doit représenter un élément de valeur (commerciale) du produit. Le Backlog de produit est la seule source de travail pour l'équipe de développement.



Définition of Done

Un accord entre le Product Owner, l'équipe de développement et les parties prenantes sur ce que signifie le fait qu'un élément du Product Backlog soit considéré comme "terminé".
La Definition of Done ne définit pas les critères d'acceptation spécifique (functional/technique) pour les éléments individuels du Product Backlog, mais constitue une liste de contrôle générale, applicable à tous les éléments du Product Backlog.



Backlog du sprint (tableau de bord)

La liste des éléments affinés du Product Backlog qui sont sélectionnés pour être livrés dans le Sprint actuel, ainsi que le plan de l'équipe pour accomplir le travail. Ce plan comprend une ventilation par tâche des éléments du Backlog de produit.



Incrément de produit

Un incrément du produit potentiellement livrable. Chaque Sprint doit aboutir à un nouvel incrément de produit qui répond à la Definition de Done.



Moniteurs de progrès

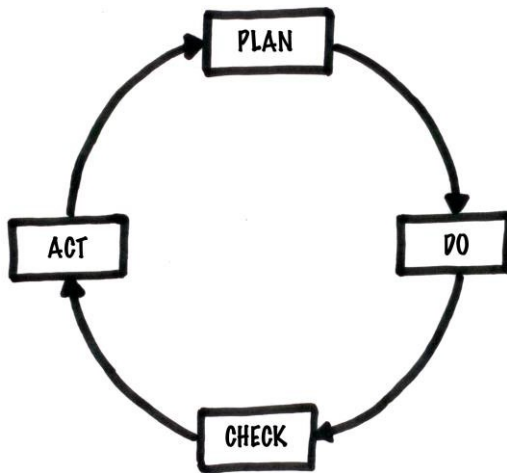
Tout artefact créé par l'équipe Scrum pour assurer la transparence sur l'état et la progression du produit. Les moniteurs d'avancement typiques comprennent les tableaux de tâches et les graphiques d'avancement au niveau du sprint et les graphiques d'avancement au niveau du produit.

Scrum et principes Agile

Principes de Scrum

Inspecter et adapter (empirisme)

Scrum est un processus empirique, ce qui signifie qu'il tire ses connaissances de l'observation et de l'expérience plutôt que de la planification et des assumptions théoriques. Pour mettre en œuvre ce principe d'**inspection et d'adaptation**, Scrum se fonde sur le cycle PDCA, également connu sous le nom de roue de Deming. Il s'agit d'une méthode itérative pour assurer le contrôle et l'amélioration continue lors de la gestion de processus ou de produits.



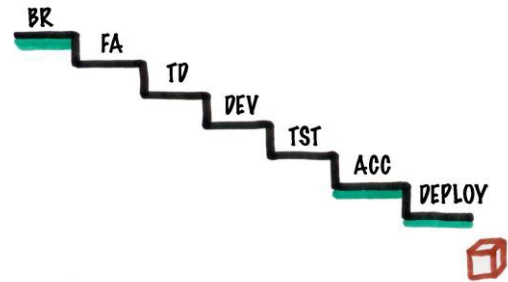
- ✓ **PLAN** Définir l'objectif et les résultats attendus et décide de la manière (processus) dont ils seront délivrés.
- ✓ **DO** Exécuter le travail selon le plan.
- ✓ **CHECK** Mesurer et analyser les résultats obtenus et comparez-les au plan initial.
- ✓ **ACT** Investir dans le plan, les résultats et le processus pour déterminer les améliorations. Mettez en œuvre ces améliorations au cours du cycle suivant.

Dans Scrum, chaque Sprint est un passage par le cycle PDCA. Les Sprints sont courts et timeboxé (bloc de temps limité) pour assurer que les assumptions faites dans la planification sont vérifiées avec la réalité aussi tôt et fréquemment que possible, menant à une qualité et des résultats améliorés.

Développement itératif et incrémental

Axé sur les plans

De nombreuses organisations utilisent des approches plus traditionnelles, axées sur la planification, pour la création de produits, généralement appelées approches "en cascade". L'hypothèse sous-jacente du travail piloté par le plan est que les exigences et la technologie sont stables et prévisibles dès le départ.



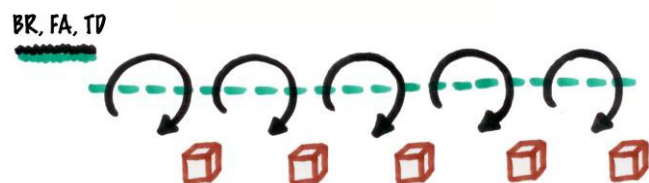
Caractéristiques :

- ✓ Phases strictement séparées, par exemple, toute l'analyse doit être faite avant que le développement puisse commencer. Un seul passage par le cycle PDCA.
- ✓ Le client n'est impliqué qu'au début et à la fin du projet (exigences et tests d'acceptation).
- ✓ Certaines des parties les plus risquées, comme l'intégration et le testing, ne sont couvertes que tardivement. La valeur commerciale (produit libérable) n'est livrée qu'à la fin du projet.

Adaptive

Lorsque toutes les relations de cause à effet ne peuvent pas être connues à l'avance, il est plus judicieux d'utiliser une approche empirique et adaptative telle que Scrum. Chaque Sprint est un cycle PDCA court qui délivre un incrément de produit petit, mais fonctionnel.

Cela permet à l'équipe Scrum d'inspecter et s'adapter très fréquemment, en évaluant les incréments de produit et l'avancement du projet et en améliorant le processus à chaque Sprint.



Caractéristiques :

- ✓ Cycles PDCA fréquents comprenant tous les travaux nécessaires (analyse, développement, test) pour livrer un incrément de produit fonctionnel.
- ✓ Une collaboration continue (quotidienne) avec le client est nécessaire.
- ✓ Livraisons fréquentes d'incrément de produits fonctionnels, apportant une valeur commerciale précoce et augmentant le retour sur investissement.
- ✓ En raison de la planification initiale minimale, les modifications des exigences du produit sont faciles et peu coûteuses.



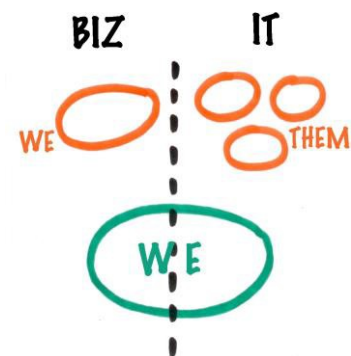
Remarques :

Lorsque vous utilisez une approche adaptative, il est tout à fait judicieux d'élaborer un plan initial pour votre produit. La clé du succès consiste à limiter autant que possible la planification et l'analyse initiales, et à se préparer aux changements futurs.

Équipe et collaboration

Scrum insiste sur le fait que l'équipe de développement doit posséder toutes les connaissances et les compétences nécessaires pour livrer des incréments de produit "faits" à chaque sprint. Cela signifie que les membres de l'équipe comprennent des personnes provenant de différents départements (comprenant généralement les affaires et l'informatique) et ayant des fonctions différentes (analystes, ingénieurs, testeurs ...).

Bien que chaque personne puisse avoir ses propres compétences et connaissances spécialisées, l'équipe a un objectif commun : collaborer pour livrer des incréments de produit/service aussi performant et efficient que possible.

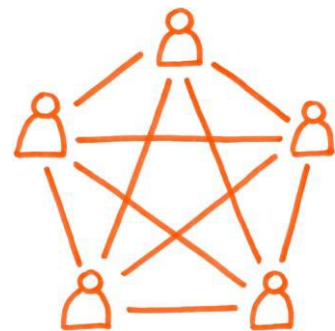


Auto-organisation

L'équipe de développement est une équipe auto-organisée. Cela signifie que les membres de l'équipe prennent toutes les décisions et actions nécessaires sur la manière d'exécuter le sprint et de livrer le prochain incrément de produit "terminé". Aucun manager, ni le ScrumMaster, ne font de la microgestion ni ne commandent et ne contrôlent l'équipe.

L'auto-organisation a besoin de frontières claires et celles-ci sont fournies par le cadre Scrum : le Sprint, la Definition of Done, le Product Backlog, les priorités et les critères d'acceptation du Product Owner, etc.

L'auto-organisation ne consiste pas à "faire ce que l'on veut" ou à permettre un travail de type cow-boy. Elle exige discipline, respect, transparence, communication et collaboration.



Valeurs de Scrum

Les valeurs ci-dessous sont essentielles à Scrum. Une organisation ou une équipe qui souhaite mettre en œuvre Scrum doit comprendre et adopter ces valeurs. Scrum requiert ces valeurs, et en même temps, Scrum facilite la vie de ces valeurs de manière plus forte et plus profonde.

Focus

La concentration est essentielle pour combiner qualité et rapidité. En se concentrant uniquement sur les éléments les plus importants, l'équipe Scrum collabore mieux, produit des produits de meilleure qualité et fournit de la valeur plus rapidement.

Ouverture

Une équipe Scrum offre une vision transparente et honnête de l'état d'avancement et de la progression. Les préoccupations et les empêchements sont ouvertement indiqués afin qu'ils puissent être traités. Cette ouverture pousse les équipes à une meilleure collaboration et à une efficacité accrue.

Respecter

Un excellent travail d'équipe et un environnement transparent exigent le respect entre les membres de l'équipe Scrum. Les membres de l'équipe se respecteront mutuellement et s'aideront à être dignes de respect.

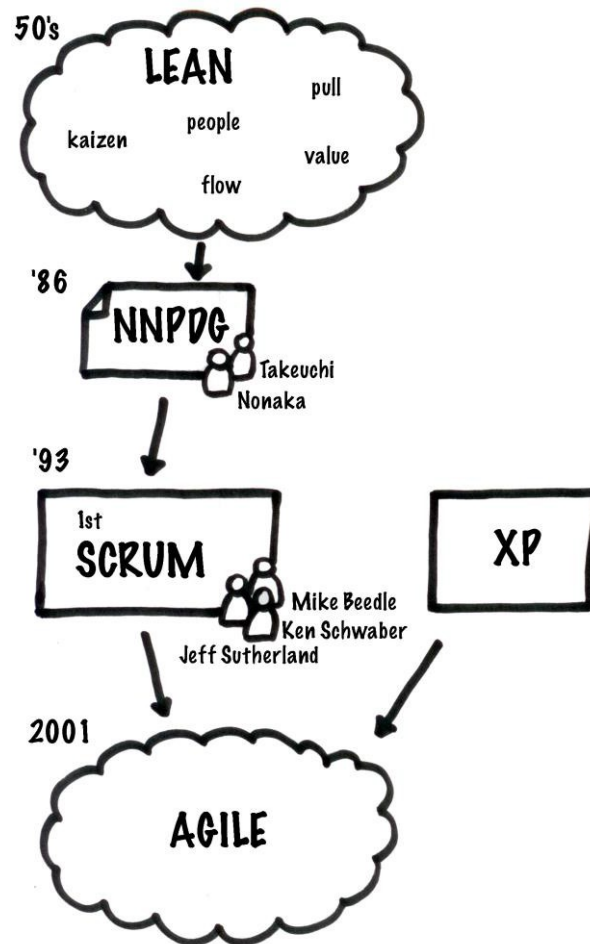
Courage

L'amélioration continue, la transparence et l'excellence du travail d'équipe exigent du courage et fournissent également le courage nécessaire pour relever de plus grands défis.

Engagement

Une équipe Scrum apprend à prendre des engagements dignes de confiance en tant qu'équipe. Cela stimule à la fois le travail d'équipe et les résultats. En ayant le contrôle de son propre développement, une équipe Scrum s'engage davantage à réussir.

Histoire de Scrum



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

Dans l'article du NNPDG, le contraste est fait entre une approche "course de relais" du développement de produits et l'approche holistique ou "rugby". L'article décrit comment le fait de se renvoyer la balle en équipe pour aller jusqu'au bout peut mieux répondre aux exigences concurrentielles d'aujourd'hui.

Ce n'est qu'en 1993 que Jeff Sutherland et son équipe chez Easel ont traduit ces idées dans la pratique du développement logiciel et ont mis en œuvre le premier Scrum. Ken Schwaber et Mike Beedle ont documenté le processus Scrum dans "Agile Software Development With Scrum".

En 2001, quelques leaders érudits de l'industrie du logiciel se sont rencontrés lors d'une conférence dans l'Utah pour discuter des méthodes de développement dites "légères". Ils ont introduit le terme Agile et ont rédigé le Manifeste pour le développement agile de logiciels qui définit leurs valeurs et principes communs.

Agile

Le Manifeste Agile définit un ensemble de valeurs et de principes. Voir www.agilemanifesto.org.

Individus et interactions vs les processus et les outils

- ✓ Créez des projets autour de personnes motivées. Donnez-leur l'environnement et le soutien dont ils ont besoin, et faites-leur confiance pour accomplir leur travail.

- ✓ La méthode la plus performante et la plus efficace pour transmettre des informations à une équipe de développement et au sein de celle-ci est la conversation face à face.

- ✓ Les meilleures architectures, exigences et conceptions émergent d'équipes auto-organisées.

Logiciel fonctionnel vs documentation exhaustive

- ✓ Livrer les logiciels de travail fréquemment, de quelques semaines à quelques mois, avec une préférence pour les échelles plus courtes.

- ✓ Le logiciel fonctionnel est la principale mesure du progrès.

- ✓ La simplicité - ici l'art de réduire le plus possible la quantité de travail non nécessaire - est essentielle.

- ✓ Les hommes d'affaires et les développeurs doivent collaborer quotidiennement tout au long du projet.

Collaboration avec les clients vs la négociation des contrats

- ✓ Notre plus grande priorité est de satisfaire le client par une livraison rapide et continue de logiciels de valeur.

- ✓ Accueillir l'évolution des besoins, même à un stade avancé du développement. Les processus agiles exploitent le changement pour l'avantage concurrentiel du client.

Réagir au changement vs le fait de suivre un plan

- ✓ Une attention particulière à l'excellence technique et à une bonne conception renforce l'agilité.

- ✓ À intervalles réguliers, l'équipe réfléchit sur la manière de devenir plus efficace, puis accorde et ajuste son comportement en conséquence.

- ✓ Les processus agiles favorisent le développement durable. Les commanditaires, les développeurs et les utilisateurs doivent être en mesure de maintenir indéfiniment un rythme constant.

Rôles de Scrum

Product Owner (PO)



Product Owner

Responsabilités

- ✓ Définit, surveille et communique la vision du produit et la planification à long terme.
- ✓ S'assurer que la valeur livrée est maximisée dans le respect des contraintes de temps et de budget.
- ✓ Représente tous les clients et les parties prenantes (commerciales).
- ✓ S'approprie, ordonne et raffine le Backlog de produit.
- ✓ Visualise et communique l'état et la progression du produit.
- ✓ Soutien l'équipe de développement dans la livraison de chaque incrément de produit.

Caractéristiques

- ✓ Idéalement le Product Owner est une seule personne, assistée par d'autres (parties prenantes, équipe de développement).
- ✓ Est transparent et parle d'une seule voix envers l'équipe Scrum, les parties prenantes, la direction ..
- ✓ A le pouvoir (mandat) de décider des variables du produit : portée, temps et budget.
- ✓ Il a une bonne connaissance des affaires et du domaine, et est familier avec la gestion des produits, le marketing, les ventes, etc.
- ✓ Est accessible à l'équipe de développement pour la soutenir pendant les sprints.

Scrum Master

Responsabilités

- ✓ Veille à ce que Scrum (et les pratiques Agile et Lean connexes) soit correctement mises en œuvre au sein de l'équipe Scrum et de l'organisation.
- ✓ Il est un leader serviteur et un coach pour l'équipe Scrum, qu'il aide à mettre en œuvre le processus Scrum.
- ✓ Veille à ce que les obstacles soient éliminés le plus rapidement possible...
- ✓ Favorise l'auto-organisation de l'équipe de développement.
- ✓ Facilite et encourage l'amélioration continue.



Scrum Master

Caractéristiques

- ✓ Une connaissance et une compréhension approfondies de Scrum, Lean et Agile et les compétences nécessaires pour les expliquer et les enseigner.
- ✓ Un leader serviteur ! Montre par l'exemple et est au service de l'équipe.
- ✓ Bonnes aptitudes à la communication, à l'encadrement et à la facilitation.
- ✓ A des connaissances sur la dynamique d'équipe, la gestion du changement culturel, l'amélioration des processus, etc.
- ✓ Est enthousiaste, dynamique et a la persévérance pour continuer à s'attaquer aux obstacles et aux améliorations.
- ✓ Connaissance de la structure, des processus, des enjeux politiques et de la culture organisationnels.
- ✓ N'a pas une vision égoïste, mais est motivé par les résultats, les améliorations, et la croissance des équipes et de l'organisation.

Équipe de développement



**Équipe de
développement**

Responsabilités

- ✓ Dans le cadre de la planification du sprint, décide de la quantité de travail qui peut être transférée dans le sprint.
u Planifie et organise le Sprint en décomposant le travail. Livrer un
- ✓ incrément de produit "terminé" à la fin de chaque sprint.
- ✓ S'auto-organise pour maximiser le travail et la qualité fournis dans un Sprint et atteindre l'objectif du Sprint.
- ✓ Soutenir et aider le Product Owner à raffiner le Product Backlog. La contribution minimale consiste à estimer le coût/efforts des éléments du Product Backlog, mais idéalement, l'équipe de développement est davantage impliquée.

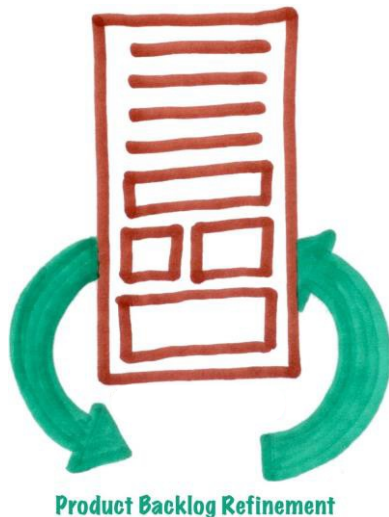
Caractéristiques

- ✓ L'auto-organisation !
- ✓ Interfonctionnelle ! Cela signifie que toutes les compétences, les connaissances et l'expérience (toutes les personnes) nécessaires pour livrer le produit "fini" font partie de l'équipe.
- ✓ La taille de l'équipe de développement est de 5 à 9 personnes.

Les rituels de Scrum

Raffinage du Backlog de produit

Le but du raffinage du Backlog de Produit est de préparer et améliorer constamment le Backlog de Produit pour le(s) Sprint(s) à venir. Le raffinage du Backlog de Produit n'est pas une activité gérée par bloc de temps limité (timebox), mais une activité continue appartenant au Product Owner et exécutée par l'ensemble de la Scrum Team. C'est une collaboration entre le Product Owner et l'équipe de développement. Il comprend :



- ✓ Ajout et suppression d'éléments dans le Backlog de produit.
- ✓ Diviser les grands éléments (Epic) en éléments plus petits (Epic ou Story sprintables).
- ✓ Ajout de détails et de critères d'acceptation aux éléments du Backlog de produit.
- ✓ Ordonner le Backlog de produit par priorité.
- ✓ Estimation de l'effort/coût des éléments du Backlog de produit.

Le Product Owner doit s'assurer de faire le bon raffinage du Product Backlog afin que celui-ci soit prêt et optimisé pour la planification du prochain Sprint.

Ordonner le Backlog de produit

Lorsqu'il ordonne le Backlog de produit, le Product Owner prend en compte les critères suivants :

- ✓ **LA VALEUR CLIENT ou COMMERCIALE** - Bien qu'elle soit difficile à mesurer ou à quantifier, le Product Owner doit définir des moyens d'estimer la valeur commerciale de chaque élément du Backlog Produit. Cela devrait guider l'ordonnancement du Backlog de produit pour atteindre la vision à long terme.
- ✓ **LE RISQUE** - Le Product Owner peut placer les éléments à risque en haut du Product Backlog pour atténuer les risques importants (techniques ou commerciaux) ou vérifier les hypothèses critiques.
- ✓ **LE COÛT** - Outre la valeur, le coût des articles est également important à prendre en compte lors de la commande pour assurer un retour sur investissement maximal.
- ✓ **LES DÉPENDANCES** - Le Product Owner devra également tenir compte des dépendances (techniques et commerciales) lors de la commande.

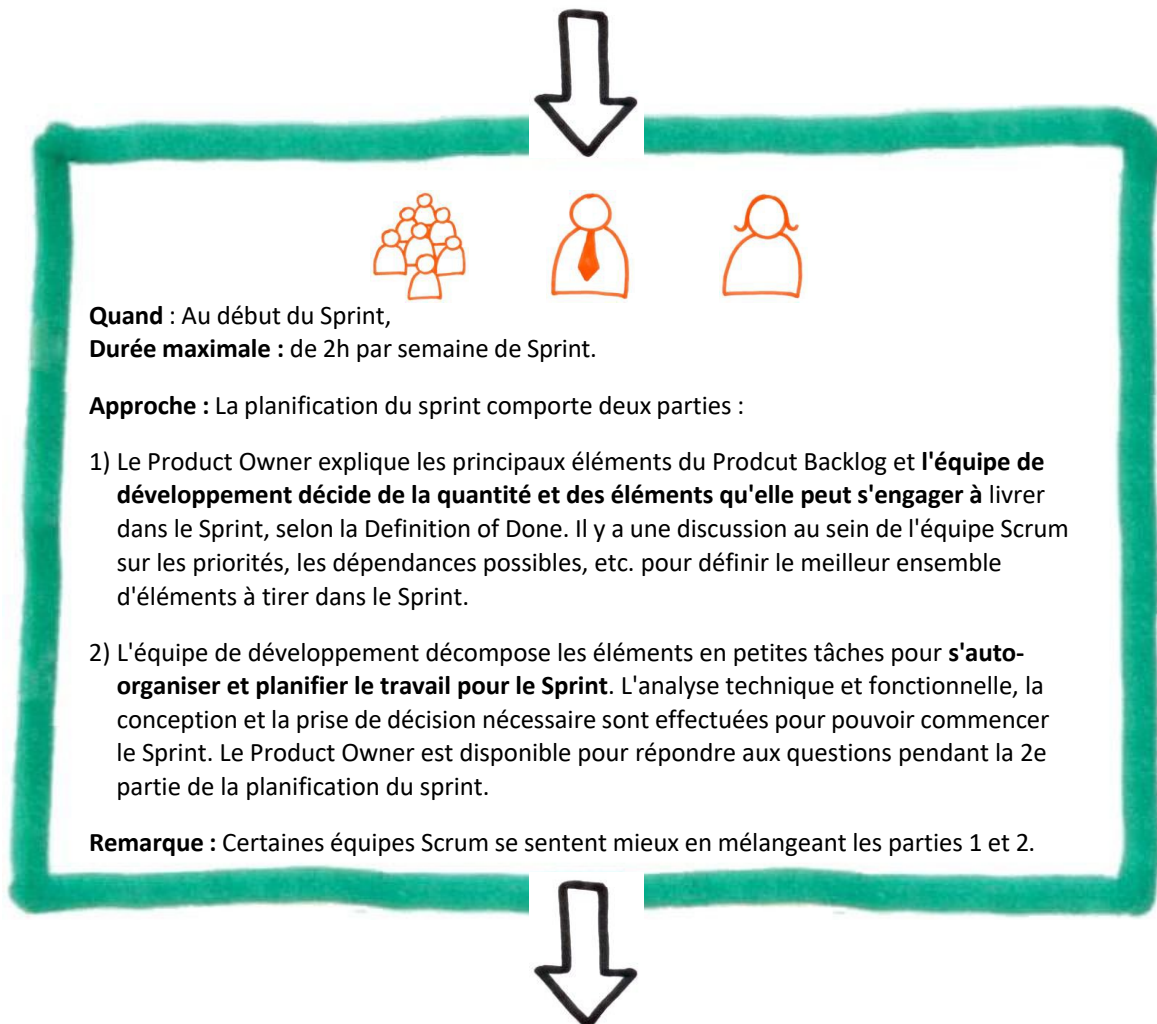
Planification du sprint

L'objectif de la planification du sprint est de permettre à l'équipe Scrum de prévoir et de planifier le sprint. Les membres de l'équipe de développement décident de la quantité de travail à transférer du Backlog de produit au Backlog de sprint, en tenant compte des priorités du Product Owner. Et ils se mettent d'accord sur la manière de réaliser ce travail.

INPUT

Un Backlog de produit "prêt", ce qui signifie :

- ✓ Il est priorisé par le Product Owner en fonction de la valeur, du risque, du coût et des dépendances.
- ✓ Les éléments les plus importants sont réalisables et "sprintables" (INVEST).
- ✓ Les éléments les plus importants sont estimés par l'équipe de développement.



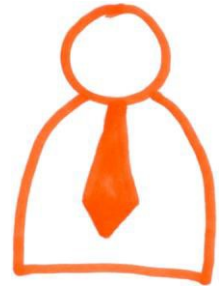
SORTIE

Le **Sprint** Backlog, contenant :

- ✓ La prévision de l'équipe concernant les éléments du Backlog de produit qui seront livrés dans ce Sprint.
- ✓ Une ventilation de ces éléments et le plan d'exécution des travaux.

Le Product Owner dans la planification du sprint

La chose la plus importante pour le Product Owner concernant la planification du sprint est d'être préparé. Il doit avoir effectué la bonne quantité de raffinage du Backlog de produit avec l'équipe de développement afin que le Backlog de produit soit prêt pour la planification du sprint.



Dans la planification du sprint, le Product Owner :

- ✓ Explique les éléments du Backlog de produit et ses décisions sur l'ordre/la priorité.
- ✓ Répond aux questions fonctionnelles et commerciales de l'équipe sur les articles.
- ✓ Convient avec l'équipe des changements possibles dans l'ordre des éléments du Backlog de produit, pour tenir compte des dépendances, de la composition et des compétences de l'équipe, etc. pour permettre à l'équipe de développement de maximiser la valeur livrée dans le Sprint.
- ✓ Raffine les critères d'acceptation fonctionnels des éléments du Backlog de produit avec l'équipe de développement si nécessaire.

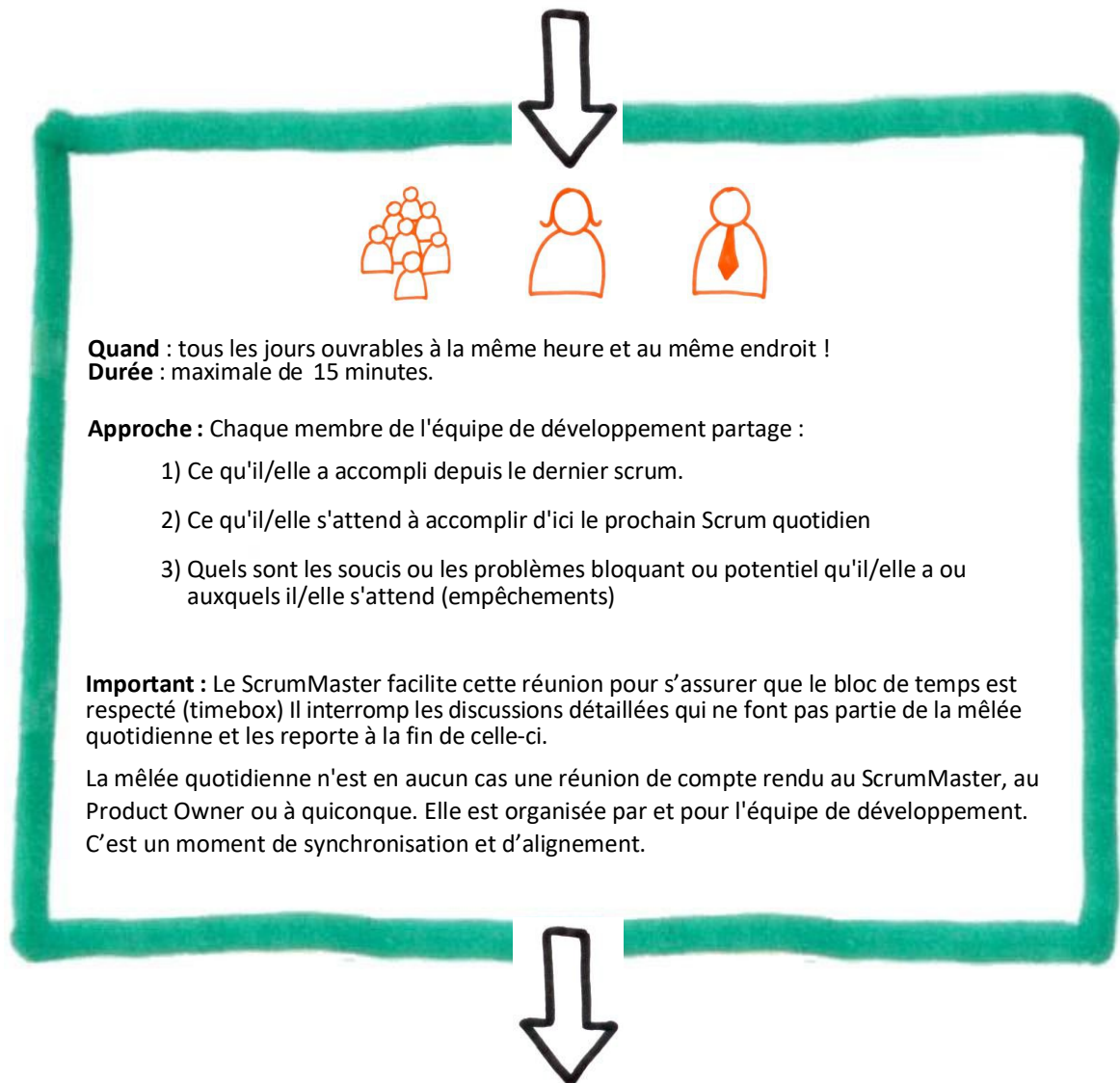
Souvent, le Product Owner prend la tête de la première partie de la planification du sprint, en expliquant les principaux éléments du Product Backlog et en aidant l'équipe de développement à décider de la quantité de travail à intégrer dans le sprint. Lorsque l'équipe de développement commence à décomposer et à planifier le travail, le Product Owner prend généralement un peu de recul. Mais il reste proche et disponible pour l'équipe pour répondre à toutes les questions qu'elle pourrait avoir.

Scrum quotidien (stand-up meeting)

Le but du Scrum quotidien est que l'équipe de développement discute de manière transparente de l'avancement du Sprint, des obstacles éventuels et de ce qui est nécessaire pour atteindre l'objectif du Sprint.

INPUT

- ✓ Les expériences et les progrès des dernières 24 heures (de travail). En général, aucun apport physique (documents, etc.) n'est nécessaire, mais les membres de l'équipe doivent venir préparés à la mêlée quotidienne.



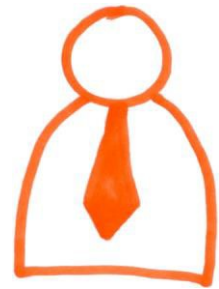
SORTIE

- ✓ Transparence sur qui fait quoi et sur les progrès (et les obstacles) vers l'objectif du sprint. Le Sprint Backlog / tableau de bord est mis à jour.
- ✓ Accord au sein de l'équipe de développement sur la manière d'aborder le jour ouvrable suivant.

Le Product Owner dans le Scrum quotidien

La présence du Product Owner n'est pas obligatoire dans le Daily Scrum. Mais il est préférable que le ou les Product Owner participent régulièrement au Scrum quotidien pour.. :

- ✓ Faire le point avec l'équipe pour savoir comment elle se porte.
- ✓ Chercher à savoir si l'équipe de développement a des questions auxquelles il doit répondre ou s'il peut aider à lever les obstacles ou en prendre connaissance
- ✓ Brièvement, explique à l'équipe certaines précisions sur les tâches en cours (en lien avec le produit) et convient des prochaines sessions de raffinement du Backlog de produit.

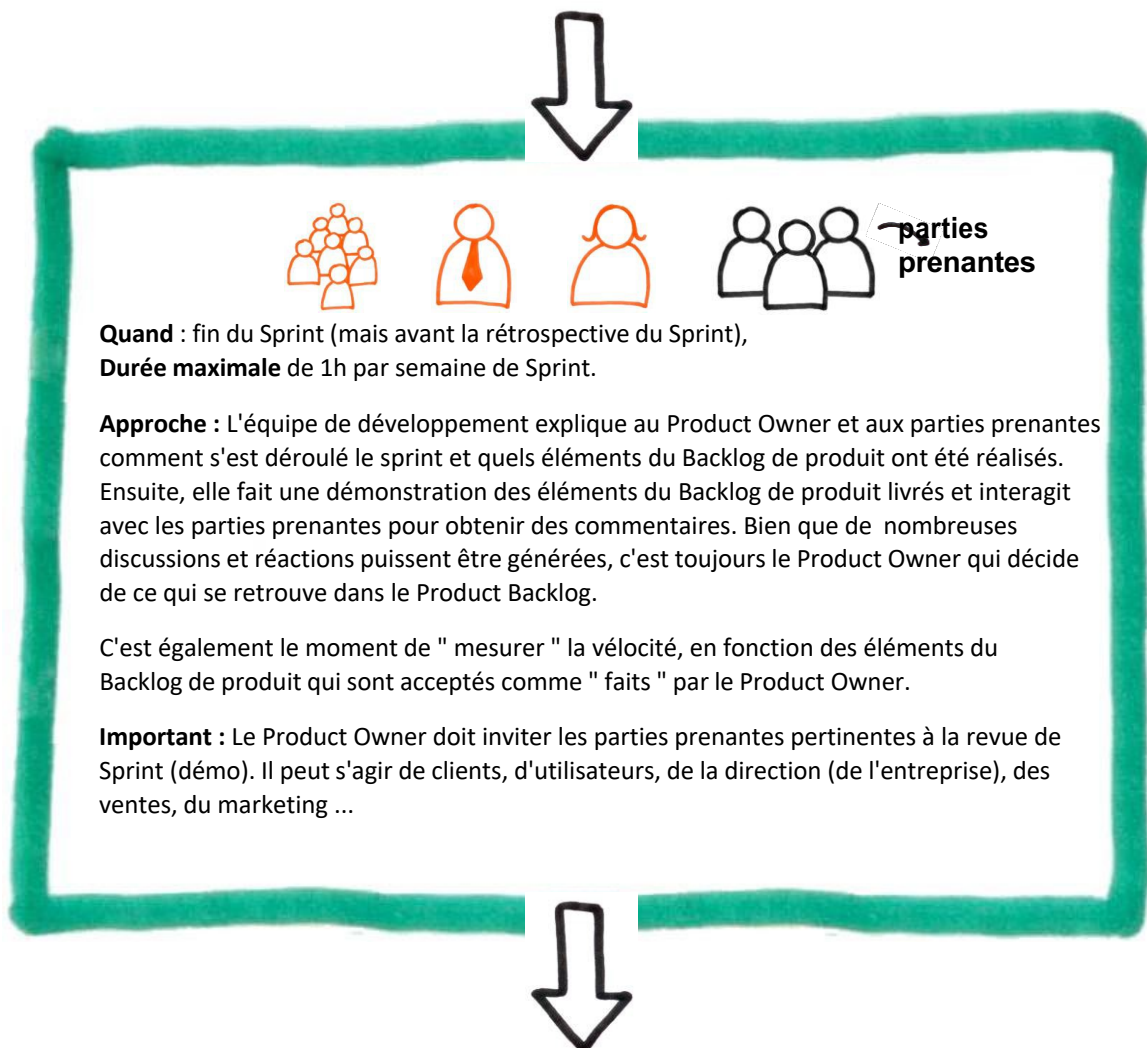


Revue de Sprint ou Démo

L'objectif de la revue de sprint est de permettre à l'équipe Scrum et aux parties prenantes intéressées de se rencontrer et d'examiner l'incrément de produit réalisé durant le sprint. Cette réunion a pour but de valider et de connaître le produit livré et d'obtenir un aperçu de la façon dont il fonctionne à ce stade.

INPUT

- ✓ L'incrément de produit/service fonctionnel qui a été réalisé pendant ce sprint.
- ✓ Clarté sur le fait que tous les éléments prévus du Backlog de produit sont vraiment faits et fonctionnels (selon la Definition of Done) à la fin du Sprint.



SORTIE

- ✓ Le plus important : l'incrément de produit/service réalisé !
- ✓ Obtenir des informations et des commentaires sur le produit de la part des parties prenantes. Récolter de nouvelles idées ou toute information utile pour le Product Owner pour les futurs éléments du Product Backlog.
- ✓ Un aperçu de la façon dont les clients utilisent le produit, ce qui permet à l'équipe de développement de prendre des décisions plus judicieuses en matière de mise en œuvre
- ✓ La vélocité de ce Sprint.

Le Product Owner dans la revue de sprint

Dans la revue de sprint, le Product Owner :

- ✓ Invite et amène les parties prenantes (commerciales) concernées : clients, utilisateurs, sponsors, marketing, ventes ...
- ✓ Accepte / rejette formellement l'incrément de produit sur la base de ses critères d'acceptation et de la Definition of Done.
- ✓ Explique sa vision du produit et ses décisions aux parties prenantes si nécessaire.
- ✓ Recueille les commentaires des parties prenantes et de l'équipe de développement sur le produit afin de mettre à jour le Backlog de produit si nécessaire.



Souvent, le Product Owner prend la tête de cette réunion, facilitant le contenu des discussions. Et donne le crédit à l'équipe pour la livraison de l'incrément de produit réalisé.

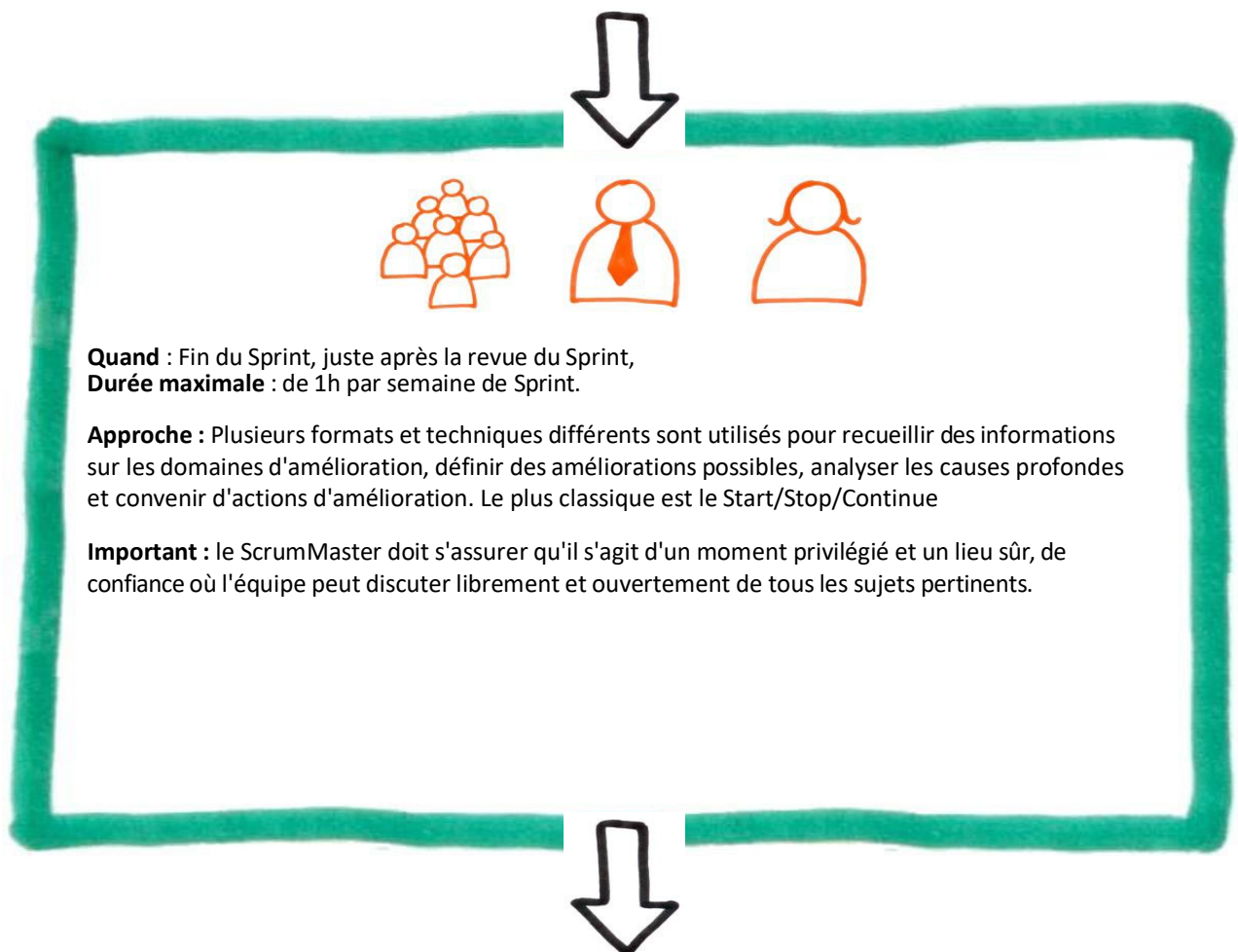
Un bon Product Owner collabore avec l'équipe de développement pendant le Sprint afin de fournir une valeur maximale lors de la revue de Sprint. Le Product Owner est préparé pour la revue de Sprint et connaît les résultats des tests de l'incrément de produit qui va être présenté et donc les éléments à accepter et à rejeter.

Rétrospective du sprint

Le but de la Rétrospective est de réfléchir sur le Sprint passé, en considérant le processus, la collaboration, les outils, la productivité et le niveau de satisfaction/motivation/enthousiasme/bien-être/harmonie, l'état d'esprit. Et sur ce constat, de construire un plan d'action pour améliorer les choses dans le(s) prochain(s) sprint(s).

INPUT

- ✓ Expérience et connaissances acquises lors du ou des Sprint(s) précédent(s)
- ✓ Commentaires recueillis lors de la revue de sprint
- ✓ Vitesse prévue et réalisée des Sprints précédents
- ✓ Les plans d'amélioration précédents et leur état actuel d'avancement



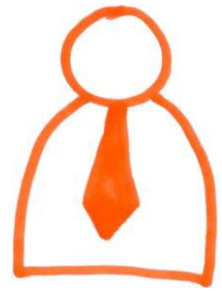
SORTIE

- ✓ Plan d'amélioration concret, réaliste, actionnable et convenu et réparti dans l'équipe. Contient des actions et des accords qui sont exécutables dans le(s) sprint(s) à venir.

Le Product Owner (PO) dans la rétrospective du Sprint

Dans la Rétrospective, le Product Owner participe de la même manière que les membres de l'équipe de développement. Tous les participants sont égaux.

Si la relation avec le Product Owner est mauvaise, cela est parfois relaté par l'équipe de développement ou le ScrumMaster lui-même qui dispensera le Product Owner de participer à la Rétrospective. Il est en effet plus important d'avoir un environnement paisible et de confiance dans la Rétrospective que d'avoir le Product Owner participant ayant une influence négative sur le processus. Toutefois, le ScrumMaster se doit dans ce cas considérer la mauvaise relation comme un obstacle important et y travailler, se faisant aider par le management ou un coach agile si nécessaire.



Si l'équipe veut consacrer une rétrospective à un problème technique pour lequel le Product Owner ne peut pas apporter d'aide, elle peut bien sûr convenir avec lui de ne pas le convier à la réunion.

Backlog de produit et User Stories

Un Backlog de produit "prêt".

Le Product Owner doit s'assurer que le Product Backlog est optimisé et "prêt" à tous moments. Cela permet à l'équipe de développement d'intégrer les bons éléments (les plus précieux, bien priorisés) du Product Backlog dans le (prochain) Sprint et de les réaliser au cours du Sprint.

Un bon Backlog de produit est **D.E.E.P.** :

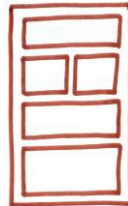
D

étaillé de manière appropriée



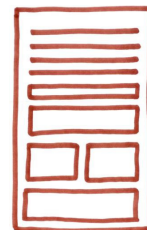
Backlog Produit

Trop d'éléments du Backlog de produit sont trop petits et trop détaillés. Aucune vue d'ensemble et une analyse initiale trop poussée et inutile.



Backlog Produit

Bonne décomposition du produit initial en éléments de haut niveau (Epic). Mais pas de petits éléments sprintables (Story).



Backlog Produit

Un niveau de détail correct : les éléments les plus importants sont petits et peuvent faire l'objet d'un sprint, les éléments moins importants sont plus grands et nécessitent une enquête ultérieure (raffinage).

É

mergent

Le Backlog de produit est un document vivant, qui est **continuellement** mis à jour et amélioré pour définir le produit ayant le plus de valeur et fournir la bonne information au bon moment. S'il est trop détaillé, il n'y a plus de place pour l'émergence d'idées nouvelles. Émergent signifie ici de garder un potentiel d'évolution.

E

stimé

La taille ou l'effort de tous les éléments du Backlog de produit doit être estimé, pour permettre au Product Owner de prendre les bonnes décisions sur les a priori et la planification du projet/produit. Et pour s'assurer que les éléments les plus importants sont suffisamment petits pour être intégrés dans un Sprint. Cela permet ensuite le choix et la planification du sprint.

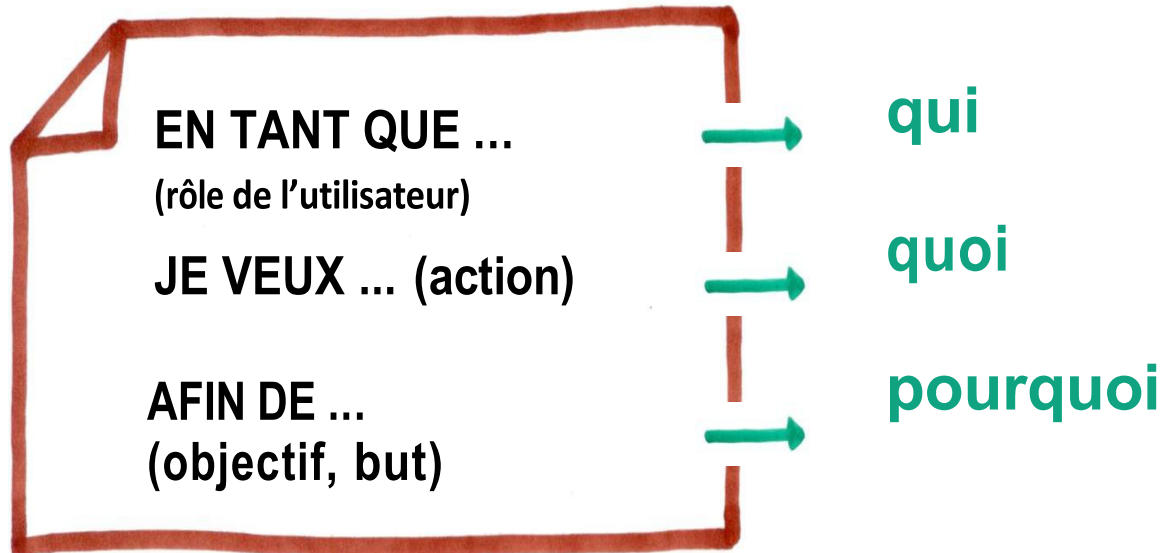
P

riorisé

À tout moment, le Product Owner doit s'assurer que le Product Backlog est ordonné de manière à réaliser le produit/service ayant le plus de valeur possible pour l'utilisateur et l'entreprise en respectant les priorités

User Stories

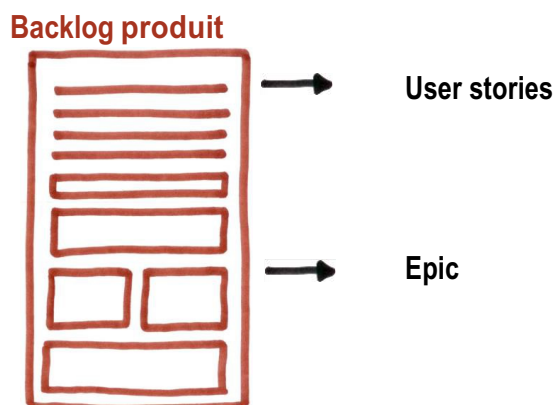
Les équipes Scrum utilisent très souvent les **user stories** comme éléments du Product Backlog, représentant les exigences des utilisateurs ou les caractéristiques du produit. Les user stories sont écrites dans un format spécifique :



Les User Stories servent des objectifs différents :

- ✓ Décrire une caractéristique du produit ou un besoin de l'utilisateur
- ✓ Une compréhension commune entre les parties prenantes/clients et l'équipe de développement.
- ✓ Un prétexte à la conversation et une occasion d'affiner l'analyse et la conception au plus juste.
- ✓ Un élément du planning pour la visibilité dans le tableau de bord (Post-it)

Epic versus user stories



Le format de user story peut être appliqué à n'importe quel niveau de détail, décrivant des exigences importantes et de haut niveau, mais aussi des exigences petites et détaillées. Les grandes user stories trop larges sont souvent appelées EPIC. Elles doivent ensuite être divisées en USER STORIES plus petites et détaillées qui peuvent être tirées dans un Sprint. Un bon Backlog de produit contient à la fois des petites Stories sprintables et des épics plus grandes qui ne sont pas encore sprintables et nécessitent un raffinement supplémentaire.

INVESTissez dans vos User Stories

Lors de la rédaction des user stories, il est bon de garder à l'esprit l'acronyme **I.N.V.E.S.T.** pour aboutir à des user stories de qualité et bien formulées.

I pour INDÉPENDANT

Les user stories doivent être aussi indépendantes les unes des autres que possible, ce qui permet une planification et une implémentation plus flexibles

N pour NÉGOCIABLE

Une bonne User Story n'est pas un contrat détaillé, mais permet la co-crédation et la négociation des détails. Elle doit capturer l'essentiel de ce qui est nécessaire pour l'estimation et la planification.

V pour Valorisable

Une User Story doit apporter une valeur « business », ce qui signifie qu'elle doit avoir de la valeur pour les utilisateurs ou les clients du produit.

E pour ESTIMABLE facilement

L'équipe de développement doit être en mesure d'estimer toute user story. Cela est nécessaire pour la planification, mais c'est aussi un bon moyen de vérifier la compréhension de la story et si elle a le bon niveau de détail.

S pour SECTIONNÉE (SMALL) Ou de taille appropriée.

Une user story qui sera bientôt intégrée dans un Sprint doit être petite, afin que plusieurs user stories puissent être intégrées dans un seul Sprint. Les epics peuvent et doivent être plus grandes, leur taille dépendant de leur importance, de leur urgence et de leur risque.

T pour TESTABLE

Une user story n'est bonne que si toutes les personnes impliquées comprennent comment la tester et l'acceptent. Généralement, des critères d'acceptation sont ajoutés à la story pour définir et se mettre d'accord sur les critères de réussite.

Les pièges des User Stories

Outre le fait de ne pas correspondre aux critères INVEST, il s'agit là de quelques erreurs typiques lors de la rédaction de User Stories :

✓ TROP DÉTAILLÉ OU FOCALISÉ SUR LA SOLUTION

Il est tentant de vouloir donner trop de détails techniques ou fonctionnels et/ou décrire une solution au lieu d'un problème, d'un besoin ou d'un objectif commercial.

✓ L'UTILISATEUR N'EST PAS BIEN DÉFINI

L'utilisateur est défini trop générique, pas d'utilisation de « persona », ou la User Story est définie d'un point de vue "système" ou technique, fonctionnelle.

✓ L'OBJECTIF DE L'ENTREPRISE AUQUEL ELLE RÉPOND N'EST PAS CLAIR

Le besoin/objectif de l'entreprise est absent ou décrit de manière trop vague. Ou bien il est plus proche du "Quoi" que du "Pourquoi", souvent caché sous la forme "Je veux X pour que Y".

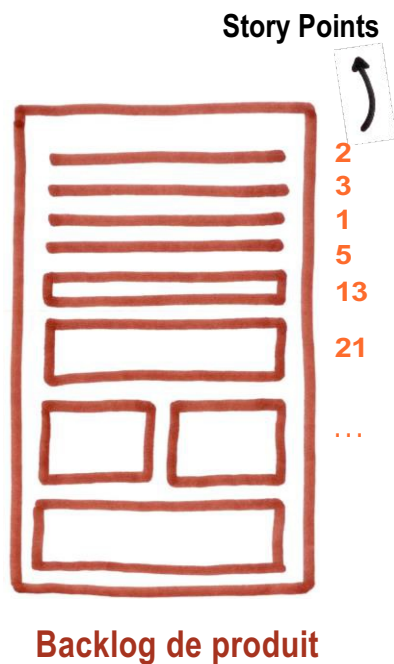
✓ PAS DE CRITÈRES D'ACCEPTATION

Aucun critère d'acceptation (conditions de validation) n'est ajouté à la Story à mesure qu'elle se rapproche de la planification du sprint.

Estimation et planification Agile

Estimations relatives

Bien que cela ne fasse pas partie de la définition de base de Scrum, beaucoup d'équipes Scrum utilisent des estimations relatives pour estimer le coût/efforts des éléments du Backlog de produit. Cela signifie :



- ✓ Les estimations sont réalisées (uniquement) par l'équipe de développement au complet, en tenant compte des connaissances de chacun et en renforçant l'engagement de l'équipe.
 - ✓ Les estimations sont exprimées en points (nombres sans signification), relatifs (et donc comparables) appelés story points.
- 0 1 2 3 5 8 13 21 ...
- ✓ La séquence de Fibonacci (ou similaire) est utilisée comme seules valeurs possibles pour les estimations. Cela permet de résoudre l'incertitude inhérente à l'estimation d'éléments plus importants.
 - ✓ L'équipe de développement s'accorde initialement sur le plus petit élément du Backlog de produit, clair à réaliser et à l'utiliser comme première valeur de référence (le "1") pour pouvoir le comparer.

Estimations relatives ou absolues

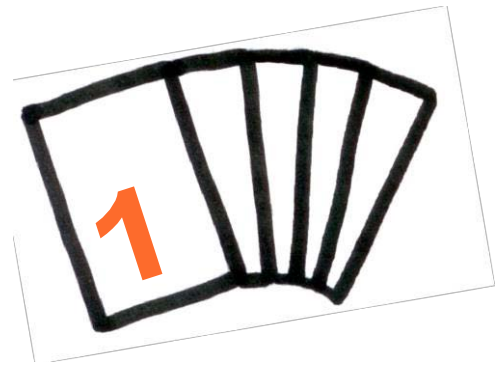
Pourquoi ne pas estimer en jours ou heures absolus ?
Parce que les études montrent que les estimations relatives sont :

- ✓ **plus rapides**
- ✓ **plus précises**

De plus, les estimations relatives sont indépendantes de la personne qui exécute (plus tard) le travail. Cela évite les reproches et les débats sans fin et permet l'auto-organisation de l'équipe de développement au sein du sprint.

Planification Poker

La technique du poker pour la planification est une excellente technique pour faire des estimations relatives. Elle évite l'ancrage et l'influence, ce qui conduit à de meilleures estimations. En outre, c'est un moyen très efficace de mettre en œuvre le principe Lean de **"décider au dernier moment"**, car il aide l'équipe à discuter des éléments du Backlog de produit au bon niveau de détail nécessaire pour obtenir des estimations décentes. Cela permet à l'équipe d'avoir les bonnes questions sur la table et de prendre les décisions fonctionnelles et techniques qui sont nécessaires à ce moment-là (et pas plus).



Comment cela fonctionne-t-il ?

- ✓ Le PO explique brièvement les éléments du Backlog de produit.
- ✓ Chaque membre de l'équipe compare avec l'élément de référence (et les estimations précédentes) et choisit la carte appropriée, la posant devant lui face cachée sur la table.
- ✓ Lorsque tout le monde a choisi, chacun montre sa carte en même temps.
- ✓ Si nécessaire, les personnes ayant les estimations les plus élevées et les plus basses expliquent les raisons de leur vote, afin que l'équipe apprenne à connaître l'élément et au besoin s'aligner sur la compréhension.
- ✓ L'équipe parvient à un consensus sur l'estimation, éventuellement en faisant un autre tour d'estimation.

Vélocité

Vélocité = story points Achevés par Sprint

La vélocité est la mesure de la quantité de travail effectuée par une équipe Scrum au cours d'un Sprint. La vélocité est exprimée en story points, et correspond à la quantité totale de story points pour tous les éléments du Product Backlog que l'équipe a livré selon la Definition of Done pour un Sprint donné.

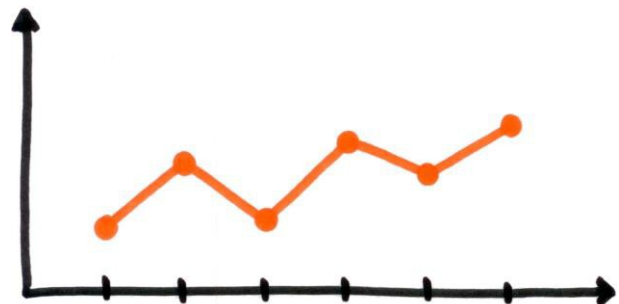
La vélocité est utilisée :

- ✓ **Par l'équipe de développement** : Dans la planification du Sprint comme un des critères pour décider de la quantité de travail à tirer dans le Sprint. Et dans la Rétrospective pour déclencher des discussions sur l'augmentation de l'efficacité.
- ✓ **Par le Product Owner** : La vélocité de l'équipe (ex. moyenne des x derniers Sprints) peut être utilisée pour faire des prévisions sur les futures versions (release planning).

La vélocité doit toujours être liée à la durée du sprint et à la composition de l'équipe. La vélocité n'est en aucun cas un indicateur de mesure permettant à la direction de comparer les équipes ou de mesurer la productivité d'une équipe.

Graphique de vélocité

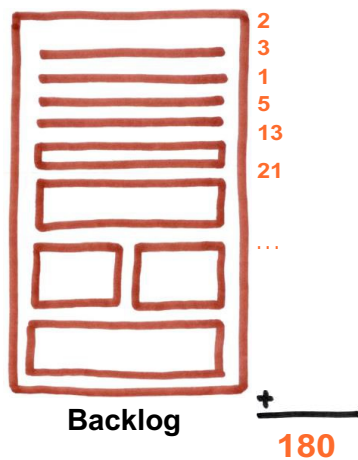
Les équipes Scrum suivent visuellement leur vélocité dans ce qu'on appelle un graphique de vélocité. Cela permet de repérer les tendances et de discuter d'éléments tels que la stabilité de la vélocité (prévisibilité) et la façon de l'augmenter.



Le graphique d'avancement de la version (release burndown graph)

Le Product Owner est responsable de la planification des versions (release) : il définit et planifie les lancements de produit et le périmètre, délai et budget du produit/service. Il assure le suivi de la progression des releases/versions pendant le développement du produit/service. L'un des outils les plus importants du Product Owner est le graphique de progression des versions du produit/service (release burndown graph).

Ce graphique montre la portée restante (en story points) par rapport au temps et visualise ainsi si l'équipe est en bonne voie ou non pour livrer selon le planning (initial).



Pour construire ce graphique, le Product Owner :

- ✓ Demande à l'équipe de développement d'estimer tous les éléments du Product Backlog qui font partie de la version.
- ✓ Additionne toutes les estimations pour obtenir le nombre total de points (story points) dans la version. (180 dans notre exemple.)
- ✓ Prévoit la vélocité sur la base de la vélocité mesurée (par exemple, la moyenne des 3-4 dernières vélocités) ou demande la prévision initiale de vélocité de l'équipe de développement (lorsqu'aucune donnée n'est disponible). Et prend en compte la durée du sprint et la composition de l'équipe.
- ✓ Dessine les progrès attendus, en fonction des contraintes de portée, de budget et de temps. (Ligne en pointillés dans notre exemple)
- ✓ Après chaque sprint, il mesure la portée restante dans le Backlog de produit et met à jour ce graphique. (Ligne orange dans notre exemple.) Le tracé de la ligne de tendance de la vélocité réelle montre si le plan initial est réalisable ou non.

