



Version française

Guide Léger de la Théorie et de la Pratique de Scrum

Version 2.0

Pete Deemer
GoodAgile

www.goodagile.com

Gabrielle Benefield
Evolve

www.evolvebeyond.com

Craig Larman

www.craiglarman.com

Bas Vodde
Odd-e

www.odd-e.com



Note aux lecteurs: Il y a beaucoup de descriptions concises de Scrum disponibles en ligne, et cet ouvrage d'initiation a pour objectif de fournir un niveau additionnel de détail sur les pratiques. Il n'a pas pour objectif de constituer l'étape finale de l'apprentissage de Scrum; il est conseillé aux équipes souhaitant adopter Scrum de se doter de l'ouvrage de Ken Schwaber *Agile Project Management with Scrum* ou *Agile Software Development with Scrum*, et de profiter des excellentes possibilités de formation ou de coaching Scrum qui sont offertes; tous les détails sont disponibles sur scrumalliance.org. Nous adressons nos remerciements à Ken Schwaber, Dr. Jeff Sutherland, et Mike Cohn pour leurs généreuses contributions.

© 2012 Pete Deemer, Gabrielle Benefield, Craig Larman, Bas Vodde

Cette traduction est mise à disposition selon les termes de la Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 3.0 France.
Pour voir une copie de cette licence: <http://creativecommons.org/licenses/by-nc-nd/3.0/fr/>



Au-delà du développement traditionnel

Le développement traditionnel fait de groupes monofonctionnels, de boucles de feedback limités et tardives, de plannings préétablis et prédictifs, et d'un flux séquentiel de l'analyse jusqu'aux tests n'est pas vraiment efficace dans notre monde extrêmement vélocé. Cette approche retarde le feedback, l'apprentissage, ainsi que le retour sur investissement potentiel car elle ne délivre le logiciel opérationnel qu'en fin de partie, occasionnant durant le jeu un manque de transparence, un manque de capacité d'amélioration, une diminution de la flexibilité, et une augmentation des risques métiers et techniques.

Une alternative – des équipes pluridisciplinaires et un cycle de développement itératif – existe depuis des décennies, mais n'a pas été aussi largement mise en œuvre que l'approche traditionnelle.

Scrum combine au sein d'un cadre simple des concepts de développement éprouvés, incluant: de véritables équipes, des équipes pluridisciplinaires, des équipes auto-organisées, des boucles de feedback couvrant un cycle complet, itératives, courtes, et réduisant le coût du changement.

Vue d'ensemble

Scrum est un cadre de développement dans lequel des équipes plurifonctionnelles réalisent des produits de manière itérative et incrémentale. Scrum structure le développement en cycles de travail appelés **Sprints**. Ces itérations ne durent jamais plus de quatre semaines (deux semaines la plupart du temps), et s'enchaînent l'une après l'autre sans interruption. Les Sprints sont d'une durée limitée, ou « *timeboxés* » – ils se terminent à une date spécifique, que le travail soit terminé ou non, et ne sont *jamais prolongés*. Généralement les Equipes Scrum choisissent une durée de Sprint et la maintiennent durant le projet, jusqu'à ce qu'elles puissent encore augmenter leur productivité et utiliser alors un cycle plus court. Au début de chaque Sprint, une **Equipe plurifonctionnelle** (environ sept personnes) sélectionne des **éléments** (exigences du client) dans une liste priorisée. L'Equipe s'accorde collectivement sur une cible constituée de ce qu'elle pense pouvoir livrer à la fin du Sprint, de manière tangible et réellement « terminée ». Aucun nouvel élément n'est ajouté durant le Sprint; Scrum accepte le changement pour le Sprint suivant, mais la durée fixe d'un Sprint en cours est faite pour se focaliser sur un objectif relativement stable, clair et limité. Chaque jour, l'Equipe se réunit brièvement afin de contrôler sa progression et ajuster les prochaines étapes nécessaires à la finalisation du travail restant. A la fin de chaque Sprint, une revue est organisée avec les parties prenantes durant laquelle l'Equipe montre ce qu'elle a réalisé. Le feedback obtenu peut être pris en compte sur le Sprint suivant. Scrum insiste sur la nécessité de livrer un produit opérationnel à la fin de chaque Sprint, et réellement « terminé ». Dans le cas de logiciels, cela signifie un système intégré, entièrement testé, documenté pour ses utilisateurs et potentiellement déployable. Les rôles, les artefacts et les événements clés sont présentés dans la figure 1.

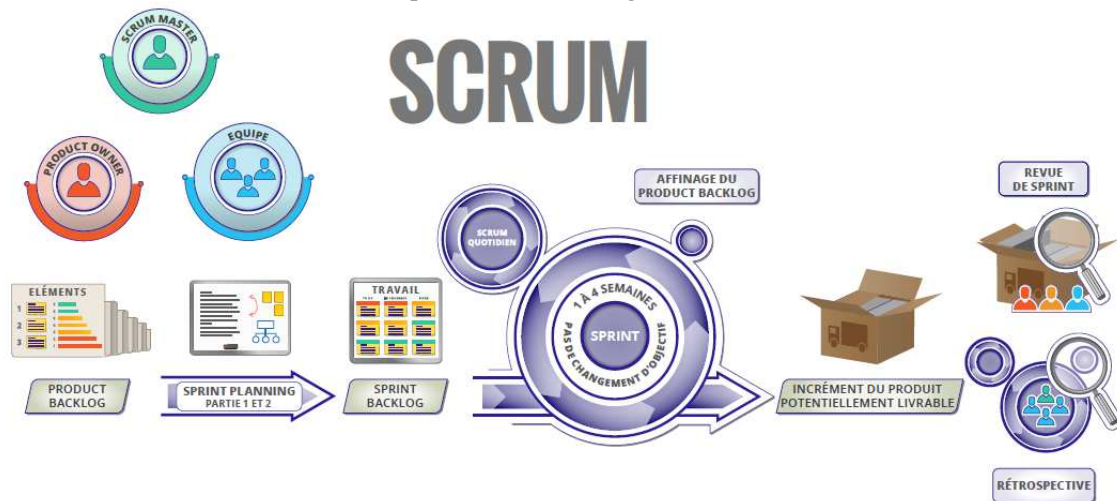


Figure 1. Vue d'ensemble de Scrum

L'un des thèmes majeurs de Scrum c'est « inspecter et adapter ». Etant donné que le développement implique inévitablement de l'apprentissage, de l'innovation, et des surprises, Scrum souligne la nécessité de développer par étapes courtes, d'inspecter à la fois l'efficacité des pratiques mises en œuvre et le produit qui en résulte, et adapter ainsi les objectifs du produit et les activités du processus. *Boucle infinie.*

Les rôles de Scrum

Dans Scrum, il y a trois rôles : le Product Owner, l'Equipe, et le ScrumMaster. Ensemble ils forment l'Equipe Scrum.

Le Product Owner est chargé de maximiser le retour sur investissement (ROI) en identifiant les fonctionnalités du produit, en les transférant dans une liste priorisée, en choisissant celles qui doivent être en tête de liste pour le prochain Sprint, et en re-priorisant et en affinant continuellement cette liste. Lorsqu'il s'agit d'un produit commercial, le Product Owner est responsable des pertes et profits. Dans le cas d'une application interne, le Product Owner n'est pas responsable du ROI au sens commercial du terme (génération de revenus), mais il est tout de même en charge de maximiser le ROI en choisissant – à chaque Sprint – les éléments ayant à la fois la plus forte valeur métier et le moindre coût. En pratique, le mot « valeur » reste un terme relativement vague, et la priorisation peut être influencée par le désir de satisfaire des utilisateurs clés, par la nécessité de s'aligner avec des objectifs stratégiques, de prendre en compte des risques, de s'améliorer, ou par d'autres facteurs. Dans certains cas, le Product Owner et le client sont la même personne ; c'est souvent le cas pour les applications internes. Dans d'autres cas, le client peut représenter des millions de personnes avec une grande variété de besoins, auquel cas le rôle du Product Owner est similaire à celui du Responsable Produit ou du Responsable Marketing chez les éditeurs. Cependant, le Product Owner est quelque peu différent du Responsable Produit traditionnel en ce sens qu'il interagit activement et fréquemment avec l'équipe, priorise en collaborant avec l'ensemble des parties prenantes et passe en revue les résultats de chaque Sprint, au lieu de déléguer les décisions liées au développement à un chef de projet. Il est important de noter que dans Scrum, il y a une et une seule personne possédant la fonction et l'autorité du Product Owner. Cette personne est responsable de la valeur du travail réalisé; même si cette personne ne doit pas travailler seule.

L'Equipe (également appelée Equipe de Développement) construit le produit qui est défini par le Product Owner : une application ou un site web par exemple. Dans Scrum, l'Equipe est « plurifonctionnelle » : elle inclut toute l'expertise nécessaire pour fournir une version du produit potentiellement livrable à chaque Sprint. Elle est également « auto organisée » (autogérée), avec un grand degré d'autonomie et de responsabilité. L'équipe décide des éléments à implémenter dans un Sprint (éléments issus de la liste proposée par le Product Owner), et des moyens les plus adaptés pour réaliser cet objectif.

Chaque personne dans l'Equipe est simplement qualifiée de *membre*. Il faut noter qu'il n'y a pas de titre de spécialiste au sein d'un groupe qui adopte Scrum; pas d'analyste fonctionnel, pas de DBA, pas d'architecte, pas de team leader, pas de concepteur IHM, pas de programmeur. Tous les membres travaillent ensemble durant chaque Sprint, par tous les moyens appropriés et nécessaires à l'atteinte de l'objectif qu'ils ont eux-mêmes déterminé.

Etant donné qu'il n'y a que des *membres*, l'Equipe n'est pas seulement pluridisciplinaire mais fait également preuve de capacité diverse d'apprentissage; chacun possède certainement des atouts particuliers, mais continue également de progresser sur d'autres spécialités. Chaque personne devra posséder des compétences sur différents domaines et à différents niveaux, et devra « aller là où le travail se trouve ». Les individus doivent être à même de prendre en charge des tâches dans des domaines moins familiers lorsqu'il est nécessaire de faciliter la finalisation d'un élément. Par exemple, une personne dont la compétence principale serait la conception d'IHM peut posséder

une compétence secondaire dans l'automatisation des tests; un autre dont la compétence principale serait la conception technique peut aussi aider en analyse et en programmation.

Une Equipe Scrum est composée de 7 personnes (plus ou moins 2 personnes), et pour un produit logiciel, l'équipe peut inclure des personnes avec des compétences en analyse, développement, test, conception d'interfaces, conception de bases de données, architecture, documentation, etc. L'Equipe développe le produit et fournit des idées au Product Owner pour le rendre encore meilleur. Dans Scrum, les Equipes sont plus productives et efficaces si tous les membres sont dédiés à 100% à un seul produit au cours d'un Sprint. Afin d'éviter des changements de contextes et un fléchissement dans la focalisation, inconvénients souvent coûteux, il faut éviter la multiplication des tâches entre plusieurs projets ou produits. Les équipes stables étant plus productives, il faut éviter les changements au sein de l'équipe. Les grands projets composés d'un nombre important d'individus sont organisés en plusieurs Equipes, chacune d'elle se concentrant sur différentes fonctionnalités du produit, avec une coordination étroite de leurs travaux. Etant donné qu'une équipe gère l'ensemble des tâches sur un périmètre client (planning, analyse, programmation et test), chaque équipe devient alors ce que l'on nomme une *équipe fonctionnelle (feature team)*.

Le ScrumMaster aide le groupe à apprendre et à appliquer Scrum afin que la valeur métier se matérialise. Le ScrumMaster fait tout ce qui est en son pouvoir pour aider l'Equipe, le Product Owner et l'organisation à réussir le projet. Le ScrumMaster n'est pas le manager de l'Equipe, ni même un chef de projet, un chef d'équipe ou un représentant de l'Equipe. Son rôle est plutôt de *servir* l'Equipe; il aide à supprimer les obstacles, protège l'Equipe des interférences extérieures, et facilite l'adoption par l'Equipe des pratiques modernes du développement. Il enseigne, il coach et il guide le Product Owner, l'Equipe ainsi que le reste de l'organisation dans une utilisation fine de Scrum. Le ScrumMaster est un formateur et un coach. Il s'assure que chacun (y compris le Product Owner, ainsi que le top management) comprenne les principes et les pratiques de Scrum, et il accompagne l'organisation dans le changement souvent difficile mais nécessaire au succès du développement agile. Etant donné que Scrum met en lumière les obstacles et les menaces qui pèsent sur l'efficacité de l'Equipe et du Product Owner, il est important d'avoir un ScrumMaster engagé et travaillant énergiquement à la résolution des problèmes auxquels seront confrontés aussi bien l'Equipe que le Product Owner. Il y a idéalement un ScrumMaster dédié et à plein temps, bien qu'une petite équipe puisse voir l'un de ses membres jouer ce rôle (on réduit alors sa charge sur les travaux courants). D'excellents ScrumMaster peuvent provenir de n'importe quelle formation ou discipline: ingénierie, conception, test, responsable produit, chef de projet ou responsable qualité.

Le ScrumMaster et le Product Owner ne peuvent pas être la même personne tant leurs objectifs sont différents, et la combinaison des deux rôles conduit souvent à de la confusion et des conflits. Cette combinaison pousse souvent le Product Owner à faire du micro management, ce qui est à l'opposé des équipes auto-organisées telles que Scrum le définit. A la différence d'un manager classique, le ScrumMaster ne dit pas à chacun ce qu'il doit faire et n'assigne pas de tâches – il facilite le processus, en soutenant l'Equipe dans son organisation et sa gestion. Si le ScrumMaster était précédemment en position de chef de projet au sein de l'Equipe, il devra changer significativement d'état d'esprit et de style d'interaction, afin que l'Equipe puisse pleinement réussir avec Scrum.

A noter qu'il n'existe pas de rôle de chef de projet dans Scrum. C'est qu'il n'est pas nécessaire; les responsabilités traditionnellement associées au chef de projet ont été divisées et réassignées aux trois rôles de Scrum, et principalement à l'Equipe et au Product Owner, plutôt qu'au ScrumMaster. Pratiquer Scrum en y ajoutant le rôle de chef de projet indique une méconnaissance fondamentale de Scrum, et conduit typiquement à des conflits dans les responsabilités, un manque de clarté dans l'autorité et de piètres résultats. Parfois un (ex-)chef de projet peut jouer le rôle de ScrumMaster, mais la réussite de cette approche est largement dépendante des individus, et de la façon dont ils comprennent les différences fondamentales entre ces deux positions, autant au niveau des pratiques au quotidien que de l'état d'esprit nécessaire pour réussir. Un bon moyen pour parfaitement comprendre le rôle de ScrumMaster, et de développer les compétences fondamentales requises, est de suivre la formation Certified ScrumMaster de la Scrum Alliance.

En plus de ces trois rôles, d'autres parties prenantes contribuent au succès du projet. Parmi elles les managers, les clients et les utilisateurs. Certaines parties prenantes, telles que les directions fonctionnelles ou techniques, pourront voir évoluer significativement leur rôles dans une organisation adoptant Scrum, mais ils gardent néanmoins toute leur valeur. Par exemple :

- Ils soutiennent l'Equipe en respectant les règles et l'esprit de Scrum
- Ils aident à supprimer les obstacles que l'Equipe et le Product Owner identifient
- Ils fournissent leur expertise et leur expérience

Dans Scrum, ces personnes ne sont plus des « nounous » (assignation des tâches, obtention de l'état d'avancement, et autres formes de micro management) mais deviennent les « gourous » et les « serviteurs » de l'Equipe (conseils, coaching, facilitant la suppression des obstacles et la résolution des problèmes, fournissant des idées créatives, et apportant leurs compétences aux membres de l'Equipe). Dans cette mutation, les managers peuvent avoir besoin de changer leur style de management; par exemple, en utilisant le questionnement socratique pour aider l'Equipe à découvrir la solution à un problème, plutôt que de simplement décider d'une solution et la faire appliquer par l'Equipe.

Le Product Backlog

Quand un groupe décide de passer à Scrum, avant même de démarrer le premier Sprint, il a besoin d'un Product Backlog, c'est à dire une liste priorisée (ordonnée 1, 2, 3, ...) de caractéristiques orientées client.

Le Product Backlog existe (et évolue) tout au long de la vie du produit; c'est la feuille de route du produit (Figure 2 et Figure 3). Tout au long du projet, le Product Backlog centralise la liste de « tout ce qui peut être fait par l'Equipe, par ordre de priorité ». Il n'existe qu'un seul Product Backlog pour un produit; le Product Owner est donc nécessaire pour décider des priorisations sur l'ensemble du périmètre, en maintenant les intérêts des parties prenantes (Equipe incluse).

Le Product Backlog contient une variété **d'éléments**, principalement des nouvelles fonctionnalités métier (« possibilité pour chaque utilisateur d'ajouter un livre dans le panier»), mais aussi des objectifs *majeurs* d'ingénierie (« réécrire le système du C++ en Java»), d'amélioration technique (« refonte du module de traitement des transactions afin de le rendre extensible»), des travaux exploratoires (« étudier des solutions pour accélérer la validation des cartes de crédit»), et potentiellement des anomalies connues (« diagnostiquer et corriger l'erreur dans le script de traitement des commandes»). Les anomalies sont insérées dans le cas où elles sont peu nombreuses (un produit avec beaucoup d'anomalies possède généralement un système de gestion et de suivi dédié).

Les éléments du Product Backlog sont élaborés de différentes manières, à partir du moment où ils restent clairs et viables. Méprise assez répandue, le Product Backlog ne contient pas des « user stories»; il contient simplement des *éléments*. Ces éléments peuvent prendre la forme de User Stories, de Use Cases, ou tout autre artefact de gestion des exigences que le groupe souhaite utiliser. Peu importe l'approche choisie du moment que la plupart des éléments apporte de la valeur au client.

Priorité	Element	Détails (URL wiki)	Effort initial estimé	Nouvel effort estimé					
				Reste à faire sur le Sprint					
				1	2	3	4	5	6
1	En tant qu'acheteur, je veux placer un livre dans un panier de courses (cf croquis de l'IHM sur le wiki)	...	5						
2	En tant qu'acheteur, je veux supprimer un livre de mon panier	...	2						
3	Améliorer les performances des transactions (cf performances cibles sur le wiki)	...	13						
4	Investiguer une solution pour accélérer la validation des cartes de crédit (cf métriques de performance sur le wiki)	...	20						
5	Mise à jour des serveurs avec Apache 2.2.3	...	13						
6	Diagnostiquer et réparer l'erreur de calcul de commande (ID bugzilla 14923)	...	3						
7	En tant qu'acheteur, je veux créer et sauvegarder une liste d'envies	...	40						
8	En tant qu'acheteur, je veux ajouter ou supprimer des articles dans ma liste d'envies	...	20						

Figure 2. Le Product Backlog

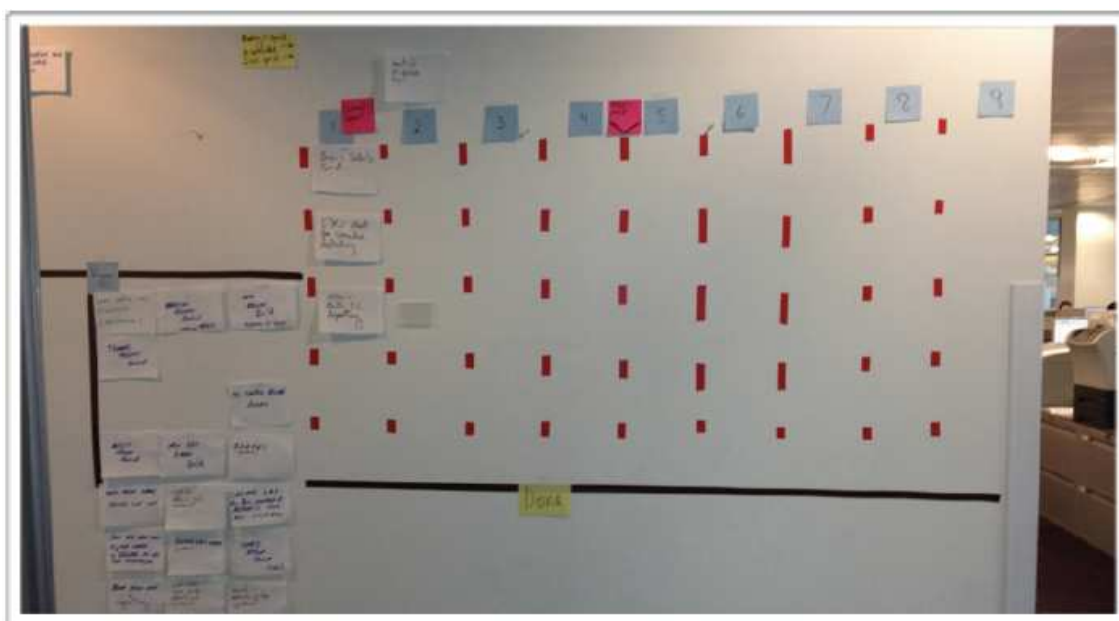


Figure 3. Management visuel : Les éléments du Product Backlog affichés au mur

Un bon Product Backlog est DEEP...

Détaillé de manière appropriée. Les éléments les plus prioritaires sont de granularité plus fine et sont plus détaillés que les éléments moins prioritaires (les premiers devant être traités avant les autres). Par exemple, les 10% du haut du backlog doivent être composés d'éléments de petite taille et parfaitement analysés, alors que les 90% restant le sont moins.

Estimé. Les éléments de la release en court doivent être estimés, et peuvent en outre faire l'objet de ré-estimations à chaque Sprint en fonction des nouvelles informations apparues et de la connaissance acquise. L'Equipe fournit au Product Owner une estimation de l'effort requis pour chaque élément du Product Backlog, et des *risques techniques* éventuels. Le Product Owner et les autres contributeurs métier répondent aux demandes formulées sur le produit, par exemple avec des informations sur les revenus générés, la diminution des coûts, les risques métiers, l'importance pour les différentes parties prenantes, et ainsi de suite.

Emergent. En réponse à la variabilité et à l'apprentissage permanent, le Product Backlog est affiné régulièrement. A chaque Sprint, des éléments peuvent être ajoutés, supprimés, modifiés,

fractionnés, et re-priorisés. Ainsi, le Product Backlog est continuellement mis à jour par le Product Owner afin de refléter les évolutions du besoin client, les nouvelles idées ou enseignements, les mouvements de la concurrence, les obstacles techniques qui apparaissent, et ainsi de suite.

Priorisé. Les éléments en haut du Product Backlog sont priorisés ou *ordonnés* de 1 à N. En général, les éléments plus prioritaires doivent donner *plus pour votre argent* : une plus grande valeur métier pour un moindre coût. L'augmentation de la priorité d'un élément peut également être motivée par la nécessité de s'attaquer aux risques importants, avant qu'ils ne vous attaquent.

Délivrer en fonction de ce qui apporte le plus de valeur n'est généralement pas un concept associé au développement traditionnel, mais ça l'est pour Scrum. Dès lors, le Product Owner doit apprendre à évaluer cette « valeur métier », et le ScrumMaster doit accompagner le Product Owner dans cet apprentissage. Que signifie « valeur métier » ? Certains groupes utilisent une simple estimation relative en point-valeur pour chaque élément. Cette estimation synthétise approximativement un ensemble de facteurs tels que la génération de revenus, la réduction des coûts, les préférences des utilisateurs, les caractéristiques différenciatrices, et ainsi de suite. Certains évaluent les éléments avec les contributions que peuvent apporter les clients pour leurs développements. Ils utilisent alors ce revenu spécifique (généralisé à court terme) en guise de valeur. Pour d'autres groupes, établir une valeur pour chaque élément reste trop vague et trop granuleux ; ils appliquent une approche de plus haut niveau, pour un lot de fonctionnalités, et basée sur un objectif de revenu effectif lors du déploiement de ce lot (« augmenter de 10% au 1^{er} septembre le taux de souscription »). Dans ce cas, le Product Owner doit définir le contenu du prochain incrément associé au Produit Viable Minimum (MVP – Minimum Viable Product).

Afin d'estimer l'effort, une technique couramment employée consiste à estimer en taille relative (factorisation de l'effort, de la complexité et de l'incertitude).

Cela reste des suggestions; Scrum ne définit pas de technique pour exprimer ou pour prioriser les éléments du Product Backlog, pas plus qu'il ne définit de technique ou d'unité d'estimation.

Une technique communément employée dans Scrum consiste à suivre la quantité de travail produite à chaque Sprint, par exemple une moyenne de 26 points par Sprint. A l'aide de cette information, l'Equipe peut projeter dans le temps une date de release qui inclut toutes les fonctionnalités, ou bien déduire le nombre de fonctionnalités qu'elle peut réaliser pour une date donnée, dans le cas où la moyenne se stabilise et qu'aucun changement n'apparaisse. Cette moyenne s'appelle la « vitesse ». La vitesse s'exprime dans la même unité que celle utilisée pour estimer les éléments du Product Backlog.

Les éléments du Product Backlog peuvent varier de manière significative en taille ou en effort. Les plus grands sont divisés en éléments plus petits durant l'atelier d'affinage (Product Backlog Refinement Workshop) ou durant la réunion de Planification de Sprint. Les plus petits d'entre eux peuvent être également assemblés. Les éléments du Product Backlog destinés aux prochains Sprints doivent être suffisamment réduits et détaillés de sorte qu'ils soient facilement assimilables par l'Equipe. Les engagements pris durant la réunion de Planification de Sprint ont ainsi du sens; la taille de chaque élément est gérable.

Étant donné qu'elles peuvent constituer des options d'investissement, les améliorations techniques majeures et consommatrices de temps et d'argent doivent se retrouver dans le Product Backlog, le choix ultime revenant au Product Owner. Il faut noter que dans Scrum, l'Equipe possède l'indépendance et l'autorité dans le choix du nombre d'éléments du Product Backlog à réaliser dans un Sprint. L'Equipe est ainsi libre de prendre en charge un certain nombre d'améliorations techniques mineures car cela fait partie du métier et que l'objectif est généralement d'améliorer le travail des développeurs. Ceci étant dit, la *majeure* partie du temps de l'Equipe durant un Sprint doit être consacrée aux objectifs du Product Owner, et non aux tâches d'ingénierie.

L'un des mythes de Scrum est qu'il permet de se soustraire à l'écriture de spécifications détaillées. En réalité, il revient au Product Owner de décider du niveau de détails requis, et cela peut varier

d'un Backlog à un autre, en fonction de la compréhension de l'Equipe, et d'autres facteurs. Il faut décrire tout ce qui est important dans un minimum de place – en d'autres termes, ne pas décrire tous les détails possibles d'un élément, mais juste clarifier ce qui est nécessaire pour la compréhension, au travers du dialogue continu entre l'Equipe, le Product Owner et les parties prenantes. Les éléments de faible priorité, dont l'implémentation est éloignée dans le temps et dont la granularité est généralement assez forte, possèdent moins de spécifications détaillées. Les éléments avec une priorité élevée devant être implémentés rapidement tendent à avoir plus de détails.

La définition de Terminé

Chaque Sprint doit produire ce qui est officiellement appelé un Incrément du Produit Potentiellement Livrable. Avant de démarrer le premier Sprint, le Product Owner, l'Equipe et le ScrumMaster doivent passer en revue les pré-requis permettant d'affirmer qu'un élément du Product Backlog est *potentiellement livrable*. Toutes les activités nécessaires au déploiement du produit doivent être ajoutées à la définition de Potentiellement Livrable et doivent ainsi être réalisées au cours du Sprint.

Malheureusement, lorsqu'une Equipe débute avec Scrum, elle n'est souvent pas en mesure de pouvoir déployer un Incrément du Produit Potentiellement Livrable à chaque Sprint. Cela est souvent dû à un manque d'automatisation ou bien à une composition de l'Equipe pas suffisamment pluridisciplinaire (par exemple certains experts ne sont pas encore intégrés dans l'Equipe). Au fil du temps, l'Equipe doit progresser afin de pouvoir déployer un Incrément du Produit Potentiellement Livrable à chaque Sprint, mais pour commencer, elle doit d'abord adapter sa Définition de Terminé à ses capacités existantes.

Avant le premier Sprint, le Product Owner et l'Equipe doivent s'accorder sur une Définition de Terminé, qui doit être un sous-ensemble des activités nécessaires pour déployer un Produit Potentiellement Livrable (pour une Equipe experte, ce sera la même chose). L'Equipe planifiera les tâches du Sprint selon cette Définition de Terminé.

Un bon Product Owner voudra toujours que la Définition de Terminé soit au plus près de Potentiellement Livrable car cela augmentera la transparence dans les développements et limitera *les retards et les risques*. Dans le cas où la Définition de Terminé diverge de Potentiellement Livrable, une partie du travail est alors repoussée juste avant la release, ce qui augmente les risques et les retards potentiels. Ce décalage dans le travail est parfois nommé *travail non fait*.

Une Equipe Scrum doit continuellement progresser, et cela se reflète au travers de l'extension de sa Définition de Terminé.

La Planification de Sprint

Résumé : réunion de préparation d'un Sprint, typiquement divisée en deux parties (la première partie c'est le « quoi » et la seconde partie c'est le « comment »).

Participants : Partie une : Product Owner, Equipe, ScrumMaster. Partie deux : Equipe, ScrumMaster, Product Owner (optionnel, mais doit être disponible pour toutes questions).

Durée : chaque partie est limitée à une heure par semaine de Sprint.

Au démarrage de chaque Sprint, la **Réunion de Planification de Sprint** a lieu. Cette réunion se divise en deux réunions distinctes, la première se nomme le **Sprint Planning Partie Une**.

Dans le **Sprint Planning Partie Une**, le Product Owner et l'Equipe passent en revue les éléments de forte priorité du Product Backlog et souhaités par le Product Owner pour ce Sprint. En général, ces éléments auront été parfaitement analysés lors d'un Sprint précédent (au cours d'une séance d'affinage du Product Backlog), de sorte que cette réunion n'aborde que des questions mineures de clarification. Lors de cette réunion, le Product Owner et l'Equipe discutent des objectifs et du

contexte de ces éléments, dans le but d'éclairer l'Equipe sur la pensée du Product Owner. Cette première partie se focalise sur la compréhension de ce que souhaite le Product Owner et pourquoi il le souhaite. A la fin de cette partie, le Product Owner (toujours très occupé) peut quitter la réunion, mais il doit rester disponible pour la partie suivante (par exemple par téléphone).

Durant la partie une, l'Equipe et le Product Owner peuvent également échafauder le **but du Sprint**. Il s'agit d'une phrase résumant l'objectif du Sprint, qui doit idéalement porter un thème fédérateur. Le but du Sprint apporte également à l'Equipe une certaine flexibilité sur le périmètre qu'elle est à même de livrer. En effet, étant donné qu'elle peut être amenée à retirer certains éléments demandés (le Sprint est limité dans le temps), elle doit néanmoins rester engagée dans la livraison d'un produit tangible et « terminé » et s'inscrivant dans l'esprit du but de ce Sprint.

Quelle est la taille idéale des éléments à sélectionner pour un Sprint ? Chaque élément doit être suffisamment petit afin de pouvoir être réalisé en beaucoup moins de temps que la durée complète du Sprint. Il est couramment recommandé de vérifier que la taille d'un élément est suffisamment petite pour être terminé au maximum en un quart de Sprint par l'ensemble de l'Equipe.

Le **Sprint Planning Partie Deux** s'intéresse plus particulièrement à *comment* implémenter les éléments que l'Equipe a décidés de prendre en charge. L'Equipe prévoit la quantité d'éléments qu'elle peut réaliser pour la fin du Sprint, en dépilant le haut du Product Backlog (en d'autres termes, en commençant avec les éléments de plus forte priorité pour le Product Owner), et en parcourant la liste dans l'ordre descendant. *C'est une pratique clé dans Scrum : l'Equipe décide de la quantité de travail à réaliser, plutôt que de se voir assigner ce travail par le Product Owner.* Cela contribue à rendre l'engagement plus fiable car celui-ci est pris par l'Equipe sur la base de son analyse et de sa planification. Bien que le Product Owner n'ait pas le contrôle sur la quantité de travail pris en charge par l'Equipe, il est assuré que ce travail a été sélectionné à partir du haut du Product Backlog – c'est à dire avec les éléments qu'il a lui-même identifiés comme étant les plus importants. L'Equipe peut cependant faire pression pour insérer des éléments positionnés plus bas dans la liste; c'est généralement le cas lorsque l'Equipe et le Product Owner s'aperçoivent qu'il est à la fois simple et opportun d'ajouter des éléments de priorité inférieure aux éléments de plus forte priorité.

La réunion de Planification de Sprint dure généralement quelques heures, mais jamais plus de quatre heures pour un Sprint de deux semaines. L'Equipe s'engage, et pour mener au succès, cet engagement nécessite une soigneuse réflexion. Les parties une et deux ont une durée équivalente; pour un Sprint de deux semaines chacune durera deux heures maximum.

Scrum ne définit pas de manière particulière pour mener cette seconde partie du Sprint Planning. Certaines équipes utilisent la vélocité enregistrée lors des derniers Sprints pour déterminer la quantité de travail qu'il est possible de produire. D'autres équipes auront une approche plus fine pour le calcul de leur capacité.

Lorsque l'Equipe utilise le calcul de capacité, elle débutera la seconde partie du Sprint Planning par l'estimation du temps que chaque membre peut effectivement dédier aux tâches du Sprint. La plupart des équipes prennent l'hypothèse que chaque membre possède un temps de travail productif compris entre 4 à 6 heures par jour – le reste du temps étant consacré aux emails, aux pauses déjeuner, à facebook, aux réunions, et à boire du café. Ainsi la capacité réelle de l'équipe pour le Sprint à venir est déterminée. Une fois que la capacité est déterminée, l'Equipe peut en déduire le nombre d'éléments du Product Backlog qu'elle peut réaliser dans la période, et comment elle pourra mener à bien ce travail. Cela débute souvent par une discussion de conception autour d'un tableau blanc. Une fois que la conception générale est comprise, l'Equipe décompose les éléments du Product Backlog en tâches. Avant de s'intéresser aux éléments du Product Backlog, l'Equipe peut analyser et créer les tâches nécessaires à un axe d'amélioration identifié lors de la dernière rétrospective de Sprint. Ensuite, l'Equipe commence par le premier élément dans le Product Backlog – c'est-à-dire l'élément le plus prioritaire pour le Product Owner – et poursuit vers le bas de la liste jusqu'à atteindre sa capacité maximale. Pour chaque élément, une liste de tâches est créée, constituée de tous les travaux nécessaires pour réaliser l'élément, ou bien de l'élément lui-

même si celui-ci ne nécessite qu'une à deux heures de travail. Cette liste de tâches est appelée le **Sprint Backlog** (Figure 4 et Figure 5).

				Nouvelles estimations de l'effort						
				Nombre de jours restants						
Élément du Product Backlog	Tâche du Sprint	Volontaire	Effort initial estimé	1	2	3	4	5	6	7
En tant qu'acheteur, je veux placer un livre dans mon panier	Modifier la base de données		5							
	Créer la page web (UI)		8							
	Créer la page web (logique Javascript)		13							
	Ecrire les tests automatisés de recette		13							
	Mettre à jour la page d'aide pour l'acheteur		3							
...										
Améliorer la performance du traitement des transactions	Fusionner le code DCP et terminer les tests		5							
	Terminer la commande machine pour pRank		8							
	Changer DCP et lecteur pour utiliser API HTTP pRank		13							

Figure 4. Un exemple de Sprint Backlog

A la fin du Sprint Planning Meeting, l'Équipe détermine de manière réaliste une cible constituée de ce qu'elle pense pouvoir produire pour la fin du Sprint. C'est ce que l'on désigne traditionnellement un engagement de Sprint – l'Équipe s'engage à faire son maximum pour atteindre cette cible. Malheureusement, cet engagement a pu être parfois interprété comme une promesse gravée dans le marbre plutôt qu'un engagement à faire son maximum. Afin d'éviter cette confusion, l'objectif de Sprint est désormais appelé une « prévision » communiquée au Product Owner.

Scrum favorise la polyvalence des ressources, plutôt que le « travail en fonction d'un titre », tel qu'un testeur ne faisant que des tests. En d'autres termes, les membres de l'Équipe « vont là où le travail se trouve », et s'entraident autant que possible. S'il y a beaucoup de tâches de test, alors tous les membres de l'Équipe peuvent y prendre part. Pour autant, cela ne veut pas dire que chacun doit être un généraliste. Il ne fait pas de doute que certains sont spécialisés dans les tests (et autre), mais le travail de l'Équipe est collectif et chacun peut apprendre des autres. En conséquence, lors du découpage et de l'estimation des tâches durant la Planification du Sprint, il n'est pas nécessaire – ni même approprié – que les personnes se portent volontaires pour les tâches « qu'elles savent faire le mieux ». Il est plutôt recommandé que chacun sélectionne ses tâches au fil de l'eau et une par une, en considérant que cette sélection va délibérément impliquer un apprentissage (éventuellement par un travail en binôme avec un spécialiste). C'est la raison pour laquelle il n'y a pas d'assignation de tâches durant la planification du Sprint, mais que ceci se fait dès que nécessaire tout au long du Sprint.

Cela dit, il existe de *rare*s cas où *Jean* doit lui-même faire une tâche particulière car pour d'autres, cela prendrait trop de temps ou serait tout simplement impossible. Peut-être que Jean est la seule ressource avec un talent de graphiste. Les autres membres de l'Équipe ne pourraient même pas dessiner « un bonhomme en bâton » même si leur vie en dépendait. Dans ce cas rare – et si ce n'est pas le cas, c'est qu'il y a un problème – il peut être nécessaire de se demander si la totalité des tâches de graphisme planifiées qui *doivent* être réalisées par Jean sont faisables au cours d'un seul Sprint.

Pour leur Sprint Backlog, beaucoup d'Équipes font usage d'un outil visuel de suivi des tâches, sous la forme d'un tableau mural (souvent appelé le tableau Scrum), où les tâches (écrites sur des Post-It) avancent de colonnes en colonnes. Ces colonnes sont étiquetées « Pas Démarré », « En Cours », et « Terminé ». Voir Figure 5.



Figure 5. Management visuel – les tâches du Sprint Backlog sur le mur

L'un des aspects fondamentaux de Scrum, c'est qu'à partir du moment où l'Équipe s'engage, tout ajout ou tout changement doit être repoussé au prochain Sprint. Cela signifie que si le Product Owner décide à la moitié du Sprint qu'il y a de nouveaux éléments sur lesquels l'Équipe doit travailler, ces changements ne seront pris en compte qu'au démarrage du prochain Sprint. Si un événement extérieur implique un changement majeur de priorité, signifiant alors que l'Équipe perdrait son temps à continuer son travail sur le Sprint en cours, le Product Owner ou l'Équipe peut stopper le Sprint. L'Équipe s'arrête, et une nouvelle réunion de Planification de Sprint instancie un nouveau Sprint. Les bouleversements alors engendrés étant potentiellement importants, le Product Owner et l'Équipe sont généralement dissuadés d'avoir recours à cette décision.

Le fait d'être protégé des changements de direction durant un Sprint influe de manière puissante et positive sur l'Équipe. Tout d'abord, l'Équipe part au travail avec l'absolue certitude que ses objectifs ne seront pas modifiés. Ceci renforce sa capacité à se concentrer sur l'atteinte de ces mêmes objectifs. De plus, cela impose au Product Owner de bien réfléchir à la priorisation des éléments du Product Backlog qui seront proposés à l'Équipe pour le Sprint.

En suivant ces règles, le Product Owner bénéficie de deux choses. Premièrement, il est assuré que l'Équipe s'engage à réaliser un lot de travaux clairs et réalistes qu'elle a elle-même choisi (l'Équipe augmente peu à peu son niveau d'expertise dans le choix et la réalisation de ses objectifs). Deuxièmement, le Product Owner a la possibilité d'effectuer tous les changements qu'il souhaite dans le Product Backlog avant le démarrage du prochain Sprint. A ce stade, les ajouts, les suppressions, les modifications, et les re-priorisations sont possibles et autorisés. Bien qu'il ne puisse pas modifier les éléments sélectionnés et développés dans le Sprint en cours, il n'est qu'à une longueur de Sprint ou moins de faire réaliser les changements qu'il souhaite. Fini la stigmatisation autour du changement – changement de direction, changement d'exigences, ou simplement changement d'avis – et c'est peut-être pour cette raison que les Product Owners sont la plupart du temps ceux qui manifestent le plus d'enthousiasme à propos de Scrum.

Le Daily Scrum

Résumé : suivi et coordination entre les membres de l'Equipe.

Participants : l'ensemble de l'Equipe est requise ; le Product Owner est optionnel ; le ScrumMaster est généralement présent mais s'assure que l'Equipe est autonome.

Durée : 15 minutes maximum.

Dès que le Sprint débute, l'Equipe se lance dans une autre pratique fondamentale dans Scrum : le **Daily Scrum**. Il s'agit d'une réunion courte (15 minutes ou moins) qui se déroule chaque jour de travail à une heure fixe. Tout le monde au sein de l'Equipe y participe. Afin de garantir la brièveté de ce rendez-vous, il est recommandé que chacun se tienne debout. C'est l'opportunité pour l'Equipe de synchroniser ses travaux et d'échanger ensemble sur les obstacles rencontrés ou à venir. Lors du Daily Scrum, chaque membre, un par un, expose successivement *aux autres membres de l'Equipe* trois points (et seulement trois): (1) Qu'a-t-il fait depuis la dernière réunion ; (2) que prévoit-il de faire jusqu'à la réunion suivante ; et (3) quels sont les points blocages ou les obstacles sur sa route. Il faut noter que le Daily Scrum n'est pas une réunion de suivi pour le reporting à un manager. C'est le moment privilégié pour une Equipe auto-organisée d'échanger sur la bonne marche de l'activité, et de se coordonner. Une personne note les problèmes rencontrés, et le ScrumMaster doit aider les membres de l'Equipe à les résoudre. Il y a peu ou pas de discussion approfondie durant le Daily Scrum, l'objectif n'étant que de répondre à ces trois questions. Si des discussions sont nécessaires, elles se tiennent lors de réunions consécutives, immédiatement après le Daily Scrum, bien que dans Scrum personne ne soit obligé d'y participer. Ces réunions ad'hoc sont des événements courants où l'Equipe prend en compte les informations remontées lors du Daily Scrum ; en d'autres termes, un moyen supplémentaire d'inspection et d'adaptation. Pour des Equipes peu expérimentées avec Scrum, la présence lors du Daily Scrum de managers ou de toute autre personne en possession d'une autorité n'est généralement *pas* recommandée. L'Equipe pourrait alors se sentir « contrôlée » – sous la pression de devoir annoncer des progrès majeurs chaque jour (espoir irréaliste), inhibée d'exposer ouvertement ses problèmes – et freinée dans sa capacité d'auto-organisation car subissant des actes de micro management. Au lieu de cela, il est plus utile pour les parties prenantes de s'enquérir de l'état des lieux auprès de l'Equipe après la réunion, et de proposer leur aide sur les points de blocage ralentissant la progression.

Suivre la progression durant le Sprint

Dans Scrum, l'Equipe est auto gérée, et pour réussir, elle doit savoir s'organiser. Chaque jour, les membres de l'Equipe actualisent leurs estimations du reste à faire pour les tâches en cours du **Sprint Backlog** (Figure 6). Il est courant qu'à la suite une personne additionne les efforts restants pour l'ensemble de l'Equipe, et le trace sur le **Sprint Burndown Chart** (Figure 7). Ce graphique montre, chaque jour, une nouvelle estimation du reste à faire estimé pour terminer les tâches de l'Equipe. Ce graphique se présente idéalement sous la forme d'une courbe inclinée vers le bas, suivant une trajectoire visant à atteindre « zéro effort restant » le dernier jour du Sprint. C'est pour cela que ce graphique se nomme un *burndown* chart [graphe de *combustion*]. Et si parfois la trajectoire est bonne, parfois elle l'est moins; c'est la réalité du développement. L'important est que cela montre à l'Equipe sa progression vers son but, non pas en fonction de combien de temps a été *consommé* dans le passé (ce qui a peu d'intérêt en termes de *progression*), mais en fonction de combien de travail il *reste à faire* dans le futur – c'est à dire ce qui sépare l'Equipe de son objectif. Si la courbe ne suit pas l'inclinaison nécessaire pour l'achèvement des travaux à l'approche de la fin du Sprint, alors l'Equipe doit s'adapter, par exemple en réduisant le périmètre du travail à faire ou en trouvant un moyen de travailler plus efficacement tout en gardant un rythme soutenable.

Bien que le Sprint Burndown Chart peut être créé et diffusé sous la forme d'une feuille de calcul, beaucoup d'Equipes estiment qu'il est plus efficace de l'afficher au format papier sur un mur de l'espace de travail. Les mises à jour sont alors réalisées au feutre; cette solution « faiblement technologique / hautement humaine » est rapide, simple, et souvent plus visible qu'un graphique au format électronique.

Elément du Product Backlog	Tâche du Sprint	Volontaire	Effort initial estimé	Nouvelles estimations de l'effort						
				Nombre de jours restants						
				1	2	3	4	5	6	7
En tant qu'acheteur, je veux placer un livre dans mon panier	Modifier le base de données	Youssef	5	4	3	0	0	0		
	Créer la page web (UI)	Philippe	3	3	3	2	0	0		
	Créer la page web (logique Javascript)	Pierre & Sophie	2	2	2	2	1	0		
	Ecrire les tests automatisés de recette	Sarah	5	5	5	5	5	0		
	Mettre à jour la page d'aide pour l'acheteur	Youssef & Philippe	3	3	3	3	3	0		
...										
Améliorer la performance du traitement des transactions	Fusionner le code DCP et terminer les tests		5	5	5	5	5	5		
	Terminer la commande machine pour pRank		3	3	8	8	8	8		
	Changer DCP et lecteur pour utiliser API HTTP pRank		5	5	5	5	5	5		
Total				50	49	48	44	43	34	

Figure 6. Mises à jour quotidiennes du reste à faire dans le Sprint Backlog

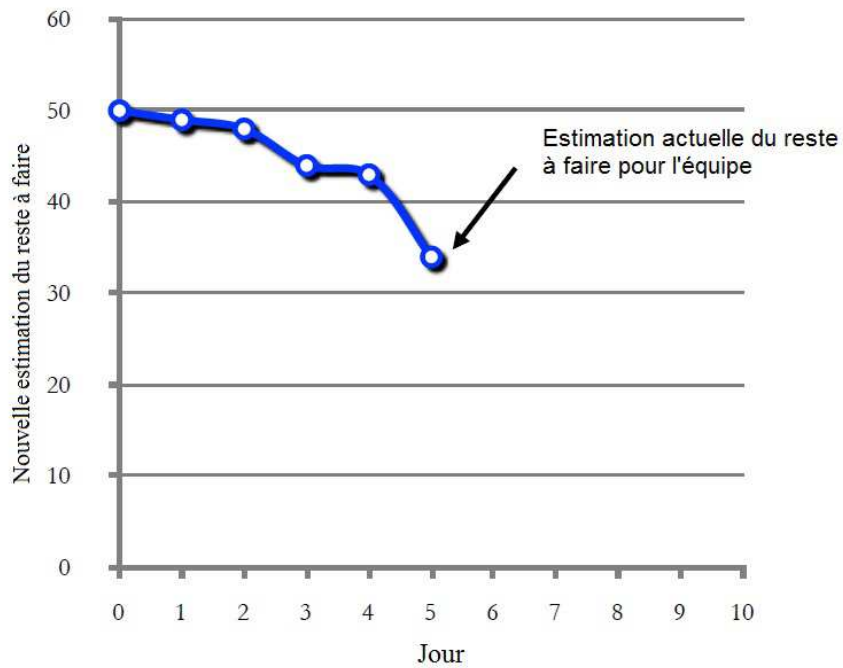


Figure 7. Sprint Burndown Chart

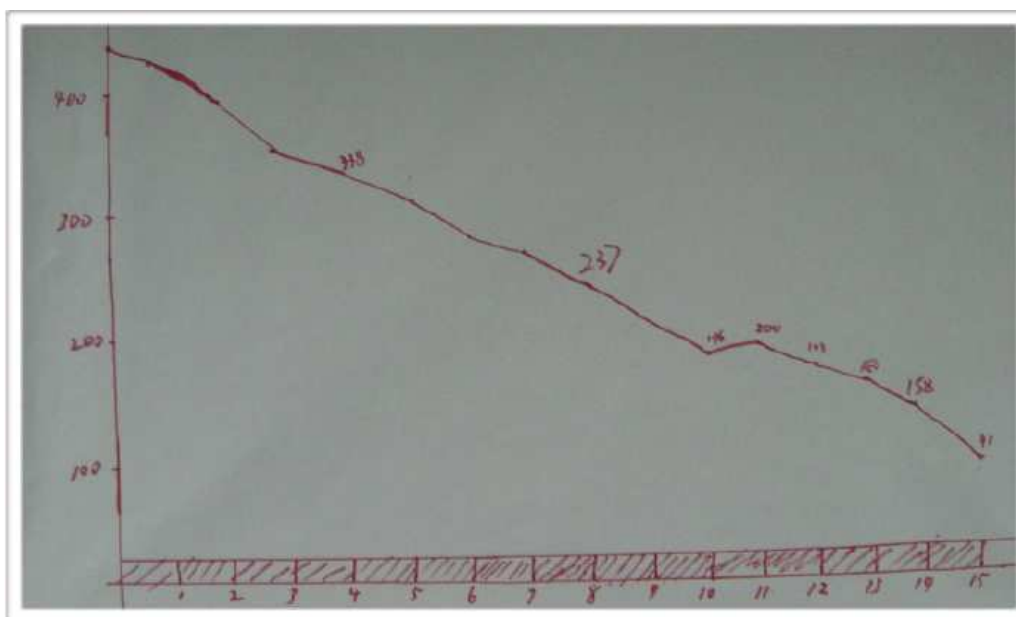


Figure 8. Management visuel : un Sprint Burndown Chart manuel

L'affinage du Product Backlog

Résumé : fractionner les gros éléments, analyser les éléments, ré-estimer, et re-prioriser, pour les Sprints à venir.

Participants : l'Equipe; le Product Owner participera entièrement à l'activité s'il possède lui-même l'expertise requise pour l'analyse détaillée, sinon il pourra simplement indiquer les directions à suivre où les re-priorisation à effectuer.

Durée : habituellement pas plus de 10% de la capacité de production de l'Equipe sur le Sprint, bien que cela puisse prendre plus de temps pour les éléments nécessitant une analyse poussée. Par exemple, sur une durée du Sprint de deux semaines, un jour pourra être consacré à l'affinage.

L'une des recommandations de Scrum les moins connues, mais néanmoins importantes, est qu'une partie de chaque Sprint doit être dédiée à l'affinage (ou « toilettage ») du Product Backlog, et ce afin de préparer les Sprints à venir. Cela inclut une analyse détaillée des exigences, le fractionnement d'éléments de forte granularité en éléments plus petits, l'estimation de nouveaux éléments, et la ré-estimation d'éléments existants. Scrum ne dit pas comment faire ce travail, mais une technique fréquemment utilisée consiste à organiser une réunion dédiée en milieu ou fin du Sprint, de sorte que l'Equipe et le Product Owner puissent se focaliser sur ce travail sans être perturbé.

Cette activité de clarification ne concerne *pas* les éléments sélectionnés pour le Sprint en cours, mais les éléments futurs, qui seront généralement traités dans un ou deux Sprints. Grâce à cette pratique, la Planification des Sprints devient relativement simple car le Product Owner et l'Equipe travaillent sur un lot d'éléments clairs, bien analysés et soigneusement estimés. Lorsque cet atelier d'affinage n'a pas été effectué (ou bien mal effectué), il y a signe qui ne trompe pas: la réunion de Planification de Sprint engendre de nombreuses questions, des découvertes, de la confusion, et se termine au final sur un sentiment d'inachevé. L'activité de planification déborde alors sur le Sprint lui-même, ce qui n'est ni souhaité ni souhaitable.

La Revue de Sprint

Résumé : inspection et adaptation de l'incrément de fonctionnalité du produit.

Participants : l'Equipe; le Product Owner, le ScrumMaster. Toutes parties prenantes en fonction du besoin, conviées par le Product Owner.

Durée : timebox d'une heure par semaine de Sprint.

Une fois le Sprint terminé, il y a la **Revue de Sprint**, réunion durant laquelle l'Equipe et le Product Owner passent le Sprint en revue. Sont présents à cette réunion le Product Owner, les membres de l'Equipe, le ScrumMaster, ainsi que les clients, utilisateurs, parties prenantes, experts, décideurs, et toute autre personne intéressée. Pour un Sprint de deux semaines, la durée maximum est de deux heures. Chaque personne présente est libre de poser des questions ou apporter des contributions.

Cette réunion est souvent nommée à tort la « démo », et cette dénomination ne permet pas d'en saisir l'objectif réel. Une idée clé dans Scrum, c'est « l'inspection et l'adaptation ». L'objectif est de regarder et d'étudier ce qu'il se passe et ainsi permettre de s'améliorer sur la base d'évaluations, en cycles itératifs. La Revue de Sprint est une activité d'inspection et d'adaptation pour le produit. C'est un moyen pour le Product Owner de connaître la situation actuelle du produit et de l'Equipe (un examen du Sprint, en quelque sorte) ; et pour l'Equipe de connaître la situation actuelle du Product Owner et du métier. Par conséquent, l'important dans cette Revue est qu'une conversation en profondeur s'établisse entre l'Equipe et le Product Owner, afin d'échanger sur la situation, obtenir des conseils, et ainsi de suite. La revue inclut certainement une manipulation de la version de l'application réalisée par l'Equipe durant le Sprint, mais si on se focalise uniquement sur le visionnage du produit plutôt que sur l'échange autour du produit, il y a alors un déséquilibre.

La côté « démo » de la Revue de Sprint ne doit pas être comprise comme une « présentation » donnée par l'Equipe – il n'y a pas de diaporama. Il s'agit d'une inspection pratique du logiciel en fonctionnement, par exemple déployé sur un environnement de développement « bac à sable ». La

salle pour la revue sera idéalement équipée d'un ou plusieurs ordinateurs, sur lesquels plusieurs personnes peuvent manipuler l'application. Il faut privilégier des sessions interactives durant lesquelles les utilisateurs et le Product Owner interagissent réellement avec le logiciel, plutôt qu'une session passive de démonstration par l'Equipe.

Pas plus de 30 minutes doivent être consacrées à la préparation de la Revue, sans quoi cela indiquerait un problème.

La Rétrospective de Sprint

Résumé : inspection et adaptation du processus et de l'environnement.

Participants : l'Equipe, le ScrumMaster, le Product Owner (optionnel). Toute autre partie prenante peut être invitée par l'Equipe, sans quoi elle n'est pas autorisée à y participer.

Durée : timebox de 45 minutes par semaine de Sprint.

La Revue de Sprint permet d'inspecter et d'adapter le produit. La Rétrospective de Sprint, qui suit la Revue de Sprint, permet d'inspecter et d'adapter le processus et l'environnement. C'est l'opportunité pour l'Equipe d'échanger sur ce qui fonctionne bien et sur ce qui ne fonctionne pas, et de s'accorder sur des changements à expérimenter. Lors de cette rétrospective, le ScrumMaster peut prendre efficacement le rôle de facilitateur, mais il est souvent préférable de trouver une personne extérieure et neutre pour mener cette réunion. Une bonne approche consiste à s'échanger le rôle de facilitateur entre ScrumMaster pour les rétrospectives, ce qui favorise la pollinisation croisée parmi les Equipes.

Il y a plusieurs techniques permettant de mener des Rétrospectives de Sprint, et le livre Agile Restrospective (Derby, Larsen 2006) en propose un certain nombre.

Beaucoup d'Equipes se concentrent sur les problèmes, et c'est dommageable. Cela peut conduire à associer les rétrospectives à des événements négatifs ou déprimants. Au lieu de cela, il faut s'assurer que les rétrospectives abordent également les forces et les aspects positifs ; il y a de nombreux ouvrages sur l'Investigation Appréciative [appreciative inquiry] qui offrent de nombreux conseils.

Démarrer le prochain Sprint

A la suite de la revue de Sprint, le Product Owner peut appliquer toute nouvelle mise à jour sur le Product Backlog – ajouter de nouveaux éléments, retirer des éléments obsolètes, ou en réécrire certains autres. Le Product Owner doit s'assurer que tout changement se répercute dans le Product Backlog. Voir la figure 9 pour un exemple de Product Backlog mis à jour.

Il n'y a pas de temps d'arrêt entre les Sprints – les Equipes passent d'une Rétrospective de Sprint un après-midi à une Planification de Sprint le matin suivant (ou après le weekend).

L'un des principes du développement agile est l'application d'un « rythme soutenable », et c'est uniquement en travaillant aux heures normales et à une cadence raisonnable que les Equipes peuvent continuer ce cycle indéfiniment. La productivité de l'Equipe augmente avec le temps grâce à l'évolution des pratiques et la suppression des obstacles qui ralentissent la cadence, et non pas en augmentant la charge de travail ou en faisant des compromis sur la qualité.

Element	Détails (URL wiki)	Priorité	Valeur estimée	Effort initial estimé	Nouvel effort estimé					
					Reste à faire sur le Sprint					
					1	2	3	4	5	6
En tant qu'acheteur, je veux placer un livre dans un panier de courses (cf croquis de l'IHM sur le wiki)	...	1	7	5	0	0	0			
En tant qu'acheteur, je veux supprimer un livre de mon panier	...	2	6	2	0	0	0			
Améliorer les performances des transactions (cf performances cibles sur le wiki)	...	3	6	13	13	0	0			
Investiguer une solution pour accélérer la validation des cartes de crédit (cf métriques de performance sur le wiki)	...	4	6	20	20	20	0			
Mise à jour des serveurs avec Apache 2.2.3	...	5	5	13	13	13	13			
Diagnostiquer et réparer l'erreur de calcul de commande (ID bugzilla 14923)	...	6	2	3	3	3	3			
An tant qu'acheteur, je veux créer et sauvegarder une liste d'envies	...	7	7	40	40	40	40			
En tant qu'acheteur, je veux ajouter ou supprimer des articles dans ma liste d'envies	...	8	4	20	20	20	20			
...										
				Total	537	580	570	500		

Figure 8. Product Backlog mis à jour.

Les Sprints continuent jusqu'à ce que le Product Owner décide que le produit est prêt à être relâché. Le summum de Scrum, c'est un produit potentiellement livrable à la fin de chaque Sprint, c'est à dire sans aucun travaux à terminer, tels que des tests ou de la documentation. Cela implique que tout soit complètement *terminé* à chaque Sprint, de telle sorte qu'il soit réellement possible de livrer le produit ou de le déployer immédiatement après la Revue. Cependant, beaucoup d'organisations possèdent des pratiques de développement, des outils et une infrastructure inadaptes, et sont dans l'incapacité d'aboutir à cette vision de la perfection. Dès lors, un Sprint de Release peut être nécessaire pour effectuer ces travaux restants. Lorsqu'un Sprint de Release est nécessaire, cela est considéré comme un mal nécessaire, mais l'organisation doit néanmoins tenter d'améliorer ses pratiques pour que cette nécessité disparaisse.

La gestion des Releases

Le problème de la planification de releases à long terme se pose parfois dans un modèle itératif. Deux cas sont possibles : (1) un nouveau produit dans sa première release, et (2) un produit existant dans une release future.

Dans le cas d'un nouveau produit, ou d'un *produit existant adoptant Scrum*, il est nécessaire de procéder à l'affinage du Product Backlog avant le premier Sprint, lorsque le Product Owner et l'Equipe initialisent un véritable Product Backlog Scrum. Cela peut prendre quelques jours, voire une semaine, et nécessiter un atelier de travail (parfois appelé Création du Product Backlog Initial ou Planification de Release), quelques analyses détaillées des besoins, et une estimation de tous les éléments identifiés pour la première release.

D'une manière plus surprenante, dans le cas d'un produit et d'un Product Backlog existants, il n'est pas nécessaire de planifier spécifiquement la prochaine release. Pourquoi ? Parce que le Product Owner et l'Equipe doivent mener l'affinage du Product Backlog à chaque Sprint (cinq à dix pourcents de chaque Sprint), de manière à continuellement préparer le futur. Ce mode de *développement continu du produit* rend inutile le besoin d'étapes jalonnées du type préparation-réalisation-transition, que l'on voit dans les cycles de développement traditionnels.

Lors de l'atelier préliminaire d'affinage du Product Backlog et durant l'affinage continu du Backlog à chaque Sprint, les connaissances de l'Equipe et du Product Owner augmentent au même rythme que la planification de la release, ainsi que de l'affinage des estimations, des priorités, et du contenu.

Certaines releases sont pilotées par les délais ; par exemple : « Nous produirons la version 2.0 de notre solution pour le salon du 10 novembre ». Dans cette situation, l'Equipe terminera autant de Sprints que possible (et développera autant de fonctionnalités que possible) dans le temps imparti. D'autres produits impliquent qu'un certain nombre de fonctionnalités soient disponibles pour pouvoir affirmer qu'ils sont terminés (le produit ne pouvant pas être lancé sans que ces besoins soient satisfaits, quelque soit le temps que cela peut prendre). Etant donné que Scrum met l'accent sur la production de code potentiellement livrable à chaque Sprint, le Product Owner peut alors choisir de faire des releases intermédiaires, afin de permettre au client de tirer plus rapidement profit du travail terminé.

Etant donné qu'il n'est sans doute pas possible de connaître tout dès le début, l'attention est portée sur la création et l'affinage d'un plan, permettant de donner à la release une orientation générale et de clarifier la manière dont les arbitrages seront rendus (périmètre versus calendrier, par exemple). Il faut imaginer cela telle une feuille de route permettant d'aller vers une destination finale ; les chemins précis à suivre et les décisions à prendre durant le voyage peuvent être déterminés en cours de route.

La destination est plus importante que le voyage.

La plupart des Product Owner mettent en place une démarche spécifique pour la gestion des releases. Par exemple, ils peuvent décider d'une date de release, et travailleront alors avec l'Equipe afin d'estimer les éléments du Product Backlog pouvant être terminés à cette date. Les éléments prévus pour la release en cours sont parfois appelés *éléments de release*. Dans une situation où un engagement de type « prix fixe / date fixe / contenu fixe » est requis, – par exemple, un projet au forfait – un ou plusieurs de ces paramètres doivent inclure une contingence permettant de gérer l'incertitude et le changement. A ce titre, Scrum ne diffère pas des autres approches.

Focalisation Produit ou Application

Pour les applications comme pour les produits – destinés à la commercialisation ou à un usage interne au sein d'une organisation – Scrum fait bouger les lignes en passant de l'ancien modèle centré sur le *projet* à un nouveau modèle centré sur le *développement continu*. Il n'y a plus de projet avec un début, un milieu et une fin. Et donc plus de chef de projet au sens traditionnel du terme. Au lieu de cela, il y a simplement un Product Owner stable et une Equipe durablement auto-organisée collaborant dans une série « sans fin » de Sprints de durée fixe, jusqu'à ce que l'application ou le produit soit retiré. Tout le travail nécessaire de gestion de « projet » est pris en charge par l'Equipe et le Product Owner – qui est un représentant interne du métier ou un responsable d'application. Il n'est pas piloté par un manager IT ou par une personne du Project Management Office.

Scrum peut également être utilisé pour de vrais projets de type « initiative ponctuelle » (versus créations ou évolutions d'applications sur du long terme). Mais même dans ce cas, ce sont l'Equipe et le Product Owner qui assurent la gestion de projet.

Que ce passe-t-il lorsqu'il n'y a pas suffisamment de nouveaux travaux issus d'une ou de plusieurs applications existantes et permettant de garantir le maintien d'une Equipe dédiée sur du long terme ? Dans ce cas, une Equipe unique et stabilisée peut sélectionner des éléments d'une application lors d'un Sprint, puis des éléments d'une autre application le Sprint suivant. Cette situation implique souvent des Sprints relativement courts, par exemple une semaine.

Mais parfois il n'y a pas suffisamment de nouveaux travaux même pour mettre en œuvre la solution précédente, et l'Equipe peut être amenée à choisir des éléments de *plusieurs* applications pour un même Sprint. Cependant, il faut se méfier de ce dispositif car il peut engendrer une gestion multitâche entre plusieurs applications générant une perte de productivité. Un pilier majeur de productivité dans Scrum, c'est une Equipe focalisée sur un produit ou une application au sein d'un même Sprint.

Les problèmes courants

Scrum ne se résume pas à un ensemble concret de pratiques – il faut plutôt retenir qu’il s’agit d’un cadre qui assure la transparence, ainsi qu’un mécanisme qui permet « l’inspection et l’adaptation ». Le fonctionnement de Scrum rend visible les dysfonctionnements et les obstacles qui impactent le Product Owner et l’efficacité de l’Equipe, afin qu’ils puissent être gérés. Par exemple, il est possible que Product Owner manque de connaissance à propos du marché, des fonctionnalités, ou sur la façon d’estimer la valeur métier relative de celles-ci. L’Equipe peut également se tromper dans ses estimations ou dans son travail de développement.

L’application du cadre de Scrum permettra de rapidement mettre en lumière ces faiblesses. Scrum ne résout pas les problèmes de développement, il les rend pleinement visible, et propose un cadre permettant aux personnes d’explorer différents moyens de résoudre ces problèmes, en cycles courts et au travers de petites expériences d’amélioration.

Si lors du premier Sprint, l’Equipe ne parvient pas à livrer ce qu’elle avait planifié à cause d’un manque de maturité dans l’analyse et dans l’estimation, cela ressemble à un échec. Mais en réalité, cette expérience est une première étape nécessaire qui permet à l’Equipe d’être plus réaliste et plus affûtée dans ses engagements. Ce modèle – où Scrum participe à rendre visible des dysfonctionnements, et permet à l’équipe d’y apporter des réponses – constitue le mécanisme de base apportant les bénéfices les plus significatifs pour les Equipes vivant l’expérience Scrum.

Lorsque la mise en pratique de Scrum est difficile, une erreur couramment rencontrée est de changer Scrum. Par exemple, les Equipes ayant du mal à respecter leurs engagements sur les Sprints peuvent décider de rendre extensible la durée des Sprints, de sorte qu’il ne manque jamais de temps – et par la même occasion, qu’il ne soit jamais possible d’apprendre à améliorer les estimations et la gestion du temps. Dans ce cas de figure, sans le coaching et le support d’un ScrumMaster expérimenté, les organisations peuvent faire de Scrum le simple reflet de leurs propres faiblesses et dysfonctionnements, se privant ainsi du réel bénéfice que Scrum offre : mettre en lumière le bon comme le mauvais, et donner l’opportunité à l’organisation de s’élever à un meilleur niveau.

Une autre erreur courante est de supposer qu’une pratique n’est pas encouragée ou qu’elle est prohibée uniquement parce que Scrum ne la requière pas spécifiquement. Par exemple, Scrum n’oblige pas le Product Owner à établir une stratégie à long terme pour son produit ; pas plus qu’il n’oblige des ingénieurs à prendre conseils auprès d’ingénieurs plus expérimentés à propos de problèmes techniques complexes. Scrum laisse la possibilité aux individus de prendre les bonnes décisions ; et dans la plupart des cas, les pratiques évoquées ci-dessus (parmi beaucoup d’autres) sont bien utiles.

Un autre point dont il faut se méfier, c’est quand des managers imposent Scrum à leurs Equipes. Scrum, c’est donner à une Equipe du champ libre et des outils pour qu’elle puisse se gérer elle-même, et se voir imposer cela d’en haut ne constituera jamais la recette du succès. Une meilleure approche consiste à initier une Equipe à Scrum par ses pairs ou par un manager, puis la former complètement grâce à une formation professionnelle. Elle ferait alors le choix en tant qu’Equipe de suivre fidèlement les pratiques pour une période définie. A la fin de la période, l’Equipe ferait le bilan de son expérimentation et déciderait de continuer ou non.

Bien que le premier Sprint représente généralement un vrai défi pour l’Equipe, la bonne nouvelle est que les avantages de Scrum tendent à émerger dès la fin de celui-ci, ce qui conduit bon nombres d’Equipe Scrum à proclamer : « Scrum est difficile, mais c’est assurément beaucoup mieux que ce que l’on faisait avant ! ».

Appendix A: lectures additionnelles

Il y a beaucoup de documentations publiées sur Scrum. Dans cette section de référence, nous souhaitons mentionner quelques ressources en ligne ainsi que quelques livres.

Ressources en ligne:

- The Lean Primer – Une introduction à la Pensée Lean, influence importante de Scrum
<http://www.leanprimer.com>
- The Distributed Scrum Primer – conseils additionnels pour les équipes distribuées
<http://www.goodagile.com/distributedscrumprimer/>
- The ScrumMaster Checklist – Une liste de questions que les bons ScrumMasters utilisent
<http://www.scrummasterchecklist.org/>
- Feature Team Primer – Adapter Scrum avec des équipes fonctionnelles
<http://www.featureteams.org>
- The Agile Atlas - Core Scrum. Description de Scrum par la ScrumAlliance
<http://agileatlas.org/atlas/scrum>
- Scrum Guide - Description de Scrum par Scrum.org
<http://www.scrum.org/Scrum-Guides>
- Agile Contracts Primer – Comment rédiger des contrats incluant Scrum
<http://www.agilecontracts.org/>

Livres:

- Leading Teams - Richard Hackman
- Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum - Craig Larman, Bas Vodde
- Practices for Scaling Lean & Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum - Craig Larman, Bas Vodde
- Agile Project Management with Scrum - Ken Schwaber
- Succeeding with Agile: Software Development using Scrum - Mike Cohn

Appendix B: terminologie

Burn Down

Tendance du reste à faire sur un Sprint, une Release ou un Produit. Les données sources proviennent du Sprint Backlog et du Product Backlog, le reste à faire est indiqué sur l'axe vertical et les périodes de temps (jours d'un Sprint, ou Sprints) sur l'axe horizontal.

Daily Scrum

Brève réunion d'Equipe qui se tient chaque jour, durant laquelle les membres inspectent leurs travaux, synchronisent leurs tâches et leurs avancements, et font état des obstacles au ScrumMaster afin de les supprimer. A la suite des Daily Scrum, des réunions peuvent avoir lieu pour adapter le travail à venir dans le but d'optimiser le Sprint.

Equipe de développement

Autre nom de l'Equipe.

Terminé

Finalisé conformément à ce qui est mutuellement convenu par toutes les parties et qui se conforme aux standards, aux conventions et aux directives. Quand quelque chose est identifié comme « terminé » lors de la Revue de Sprint, cela doit être conforme à cette définition.

Reste à faire estimé (éléments du Sprint Backlog)

Estimation faite par l'Equipe du nombre d'heures restant à travailler sur une tâche donnée. Ces estimations sont revues chaque jour pour chaque tâche en cours du Sprint Backlog. L'estimation correspond au total des heures restantes, indépendamment du nombre de personnes qui exécutent le travail.

Incrément

Fonctionnalité du produit qui est développée par l'Equipe durant chaque Sprint et qui est potentiellement livrable ou utilisable par les parties prenantes.

Incrément de fonctionnalités du produit potentiellement livrable

Partie complète du produit ou du système pouvant être utilisé par le Product Owner ou par les parties prenantes s'ils désirent l'exécuter.

Sprint

Itération, ou cycle répétitif de travaux similaires, produisant des incréments du produit ou du système. Ne dure pas plus d'un mois et généralement plus d'une semaine. La durée est fixée pour la totalité du projet et toutes les équipes travaillant sur le même produit ou système utilisent la même durée.

Product Backlog

Liste priorisée d'exigences avec une estimation du temps nécessaire pour les transformer en fonctionnalités finies du produit. Plus un élément possède une priorité élevée dans le Product Backlog, plus son estimation est précise. La liste émerge et évolue en fonction des changements du métier ou des technologies.

Élément du Product Backlog

Exigences fonctionnelles, exigences non fonctionnelles, et problèmes, priorisés par ordre d'importance pour le métier, avec leurs dépendances et leurs estimations. La précision de l'estimation dépend de la priorité et de la granularité de l'élément du Product Backlog ; les éléments de plus forte priorité sélectionnables pour le prochain Sprint étant les plus fins et les plus précis.

Product Owner

Personne en charge de gérer le Product Backlog, afin de maximiser la valeur pour le projet. Le Product Owner est le représentant de toute personne ayant un intérêt dans le projet et dans son produit final.

Scrum

N'est pas un acronyme, mais un mécanisme en rugby pour remettre en jeu une balle sortie du jeu.

ScrumMaster

Personne responsable du processus Scrum, de la conformité de son implémentation, et de l'optimisation de ses bénéfices.

Sprint Backlog

Liste définissant le travail d'une Equipe durant un Sprint. Elle se décompose souvent en un lot de tâches plus détaillées. La liste est créée durant le Sprint Planning Meeting et peut être mise à jour durant le Sprint par l'Equipe, avec la suppression ou l'ajout de tâches en fonction des besoins. Chaque tâche du Sprint Backlog sera suivie durant le Sprint et montrera l'estimation du reste à faire.

Tâche du Sprint Backlog

Tâche que l'Equipe ou l'un de ses membres identifie comme nécessaire à la transformation d'un élément du Product Backlog en fonctionnalité du système et sur laquelle l'Equipe s'est engagée.

Réunion de Planification de Sprint

Réunion d'une durée limitée à quatre heures (pour un Sprint de deux semaines) qui marque le début de chaque Sprint. La réunion est divisée en deux parties de deux heures, également d'une durée fixe. Durant la première partie le Product Owner présente à l'Equipe les éléments du Product Backlog les plus prioritaires. L'Equipe et le Product Owner collaborent afin que l'Equipe puisse déterminer la quantité d'éléments qu'elle peut transformer en fonctionnalité durant le Sprint. Durant la seconde partie, l'Equipe détermine comment elle va implémenter ces éléments, en concevant et en détaillant son travail pour atteindre le But du Sprint.

Réunion de Rétrospective Sprint

Réunion animée par le ScrumMaster durant laquelle l'Equipe au complet discute du Sprint qui vient de s'achever et détermine ce qui peut être modifié et qui rendrait le Sprint à venir plus agréable et plus productif.

Réunion de Revue de Sprint

Réunion limitée à deux heures (pour un Sprint de deux semaines) se tenant à la fin de chaque Sprint. L'Equipe collabore avec le Product Owner et les parties prenantes et inspectent le résultat du Sprint. Cela commence généralement par une revue des éléments terminés, une discussion sur les opportunités, les contraintes et les découvertes, et une discussion sur ce qu'il est préférable de démarrer (ce qui peut donner lieu à des changements dans le Product Backlog). Seules les fonctionnalités terminées peuvent être présentées.

Partie prenante

Personne ayant un intérêt dans le résultat d'un projet, parce qu'il le finance, qu'il l'utilise, ou qu'il est concerné par celui-ci.

Equipe

Groupe plurifonctionnel de personnes responsables de s'autogérer dans le but de réaliser un incrément du produit à chaque Sprint.

Boite de temps (Timebox)

Période d'une durée fixe et ne pouvant être étendue, durant laquelle se déroule un événement ou une réunion. Par exemple, une réunion Daily Scrum est limitée à quinze minutes et se termine à la fin des quinze minutes, quoiqu'il arrive. Certaines réunions peuvent être plus courtes. Pour les Sprints, cela dure exactement ce temps.