

Scrum (méthode)

Un article de Wikipédia, l'encyclopédie libre.

Aller à : [Navigation](#),[rechercher](#)

Pour les articles homonymes, voir [Scrum](#).

Scrum est une [méthode agile](#) dédiée à la gestion de projets.

La méthode s'appuie sur le découpage d'un projet en incréments, nommés "sprint", ainsi que l'auto-organisation de l'équipe de développement. Les sprints peuvent durer entre quelques heures et un mois (avec une préférence pour deux semaines). Chaque sprint commence par une estimation suivie d'une planification opérationnelle. Le sprint se termine par une démonstration de ce qui a été achevé, et contribue à augmenter la valeur d'affaires du produit. Avant de démarrer un nouveau sprint, l'équipe réalise une rétrospective : elle analyse ce qui s'est passé durant ce sprint, afin de s'améliorer pour le prochain.

Scrum est issu des méthodes incrémentales (telles que le [modèle en spirale](#)) qui permettent de maîtriser une production planifiée. Scrum n'autorise pas l'aspect "adaptatif " car il ne propose pas de pratiques permettant de mesurer les modifications importantes et leurs incidences sur le planning de réalisation. La version 2011 du guide de Scrum permet par contre l'affinement (itératif) des exigences en cours de développement.

La méthode Scrum ne couvre aucune technique d'ingénierie du logiciel. Pour l'utiliser afin de développer une application, il est nécessaire de la compléter avec des pratiques de qualité du logiciel. Par exemple, on pourra utiliser des pratiques issues de l'[Extrême Programming](#), de la phase de Construction structurée de la méthode RAD, ou un ensemble de pratiques de qualité du logiciel.

Sommaire

- [1 Historique](#)
- [2 Caractéristiques](#)
- [3 Les trois piliers de Scrum](#)
 - [3.1 La transparence](#)
 - [3.2 L'inspection](#)
 - [3.3 L'adaptation.](#)
- [4 Rôles](#)
 - [4.1 Propriétaire du produit](#)
 - [4.2 ScrumMaster](#)
 - [4.3 Développeur](#)
- [5 Évènements](#)
 - [5.1 Le Sprint](#)
 - [5.2 Mêlée Quotidienne](#)
 - [5.3 Réunion de Planification de Sprint](#)
 - [5.4 Revue de sprint](#)
 - [5.5 Rétrospective du sprint](#)
- [6 Artéfacts de Scrum](#)

- [6.1 Carnet du produit](#)
 - [6.2 Carnet de sprint](#)
 - [6.3 Incrément](#)
 - [6.4 Définition de complété](#)
- [7 Compléments](#)
 - [7.1 Estimations](#)
 - [7.1.1 Items de backlog de produit](#)
 - [7.1.2 Calcul de vélocité](#)
 - [7.1.3 Items de backlog de sprint](#)
 - [7.2 Lancement du projet](#)
 - [7.3 Vue globale](#)
 - [7.4 Burndown Charts](#)
 - [7.4.1 Sprint Burndown Chart](#)
 - [7.4.2 Release Burndown Chart](#)
 - [7.4.3 Interprétation](#)
 - [7.5 Qualité de l'environnement de travail](#)
 - [7.6 Risques de la méthodologie](#)
 - [7.6.1 Un Scrum Master directif](#)
 - [7.6.2 La communication constructive laisse la place aux reproches](#)
 - [7.7 Documentation de projet](#)
 - [7.8 Outils pour Scrum](#)
 - [7.8.1 Papier et crayon / tableur](#)
 - [7.8.2 Autres outils](#)
- [8 Conclusion](#)
 - [8.1 Scrum](#)
 - [8.2 Mise en garde](#)
- [9 Glossaire](#)
- [10 Voir aussi](#)
 - [10.1 Articles connexes](#)
 - [10.2 Liens externes](#)
- [11 Bibliographie anglophone](#)
- [12 Bibliographie francophone](#)
- [13 Notes et références](#)

[[modifier](#)] Historique

La métaphore de Scrum ([mêlée](#) du rugby) apparaît pour la première fois dans une publication de Takeuchi et Nonaka intitulée *The New New Product Development Game*^[1] qui s'appliquait à l'époque au monde industriel.

En [1986](#),[Hirotaka Takeuchi](#) et [Ikujiro Nonaka](#) décrivent une nouvelle approche [holistique](#) qui augmenterait la vitesse et la flexibilité dans le développement de nouveaux produits^[2]. Dans celle-ci les phases se chevauchent fortement et l'ensemble du processus est réalisé par une équipe aux compétences croisées à travers différentes phases. Ils ont comparé cette nouvelle approche au [rugby à XV](#), où l'équipe essaye d'avancer unie, en faisant circuler la balle (« tries to go to the distance as a unit, passing the ball back and forth »).

En [1991](#), DeGrace et Stahl, dans "Wicked problems, righteous solutions"^[3], font référence à cette approche sous l'appellation « Scrum » ([mêlée](#), en anglais), un terme de rugby mentionné dans l'article de Takeuchi et Nonaka.

En [1995](#), Ken Schwaber présente une communication décrivant Scrum à l'[OOPSLA](#), à [Austin](#), aux [États-Unis](#).

Références de publications officielles requises

Schwaber et Sutherland auraient collaboré au cours des années suivantes pour fusionner les publications, leurs expériences et les meilleures pratiques du secteur en ce qui est maintenant connu comme Scrum.

En [2001](#), Ken Schwaber fait équipe avec Mike Beedle pour décrire la méthode dans le livre « Agile Software Development With Scrum ».

[\[modifier\]](#) Caractéristiques

Le terme *Scrum* est emprunté au [rugby à XV](#) et signifie *mêlée*. Ce processus s'articule en effet autour d'une équipe soudée, qui cherche à atteindre un but, comme c'est le cas en rugby pour avancer avec le ballon pendant une mêlée.

La méthode Scrum a été conçue lors de projets de développement de logiciels. Elle peut aussi être utilisée par des équipes de maintenance. Dans le cas de très grands projets, les équipes se multiplient et on parle alors de *Scrum de Scrums*. La méthode Scrum peut théoriquement s'appliquer à n'importe quel contexte ou à un groupe de personnes qui travaillent ensemble pour atteindre un but commun comme planifier un mariage, gérer des projets de recherche scientifique, des écoles et même des églises comme le précise le [site](#) de son principal promoteur Jeff Sutherland.

Le principe de base de Scrum est de focaliser l'équipe sur une partie limitée et maîtrisable des fonctionnalités à réaliser. Ces incréments se réalisent successivement lors de périodes de durée fixe de une à quatre semaines, appelées **sprints**. Chaque sprint possède, préalablement à son exécution, un **but** à atteindre, défini par le *propriétaire du produit*, à partir duquel sont choisies les fonctionnalités à implémenter dans cet incrément. Un sprint aboutit toujours à la livraison d'un produit partiel fonctionnel. Pendant ce temps, le *ScrumMaster* a la charge de former le directeur de produit, l'équipe et l'organisation entière à la méthode Scrum.

Un principe fort des méthodes Agiles est la participation active du client. Cela permet de choisir plus finement les fonctionnalités réalisées à chaque incrément. Le directeur de produit peut à tout moment compléter ou modifier la liste des fonctionnalités à produire pour les prochains sprints. Sans modifier le but du sprint en cours, celui-ci peut être affiné et faire l'objet d'une renégociation entre le propriétaire du produit et l'équipe de développement suite à de nouvelles connaissances. Si le but du sprint devient obsolète, le propriétaire du produit a la capacité d'annuler un sprint en cours.

Chaque sprint constitue donc un incrément, facilitant le pilotage du projet. La notion d'[itération](#) couvre l'adaptabilité au quotidien. Cette adaptabilité est limitée par le but immuable d'un sprint.

[\[modifier\]](#) Les trois piliers de Scrum

Scrum est un processus empirique : il se base sur l'expérience du terrain. Il s'appuie sur trois piliers. ^[4]

Il suit également les principes des [méthodes agiles](#).

[\[modifier\]](#) La transparence

Scrum met l'accent sur le fait d'avoir un langage commun entre l'équipe et le management. Ce langage commun doit permettre à tout observateur d'obtenir rapidement une bonne compréhension du projet.

[\[modifier\]](#) L'inspection

A intervalle régulier, Scrum propose de faire le point sur les différents artefacts produits, afin de détecter toute variation indésirable.

Ces inspections ne doivent pas être faites trop fréquemment, ou par un inspecteur mal formé : cela nuirait à l'avancement du projet.

[\[modifier\]](#) L'adaptation.

Si une dérive est constatée pendant l'inspection, le processus doit alors être adapté. Scrum fournit des rituels, durant lesquels cette adaptation est possible. Il s'agit de la *réunion de planification de sprint*, de la *mélée quotidienne*, de la *revue de sprint* ainsi que de la *rétrospective de sprint*.

[\[modifier\]](#) Rôles

Scrum définit trois rôles : le **Propriétaire du produit** (*Product Owner*), le **ScrumMaster** et le **Développeur**. Il est à noter que le rôle de développeur couvre plusieurs métiers d'une organisation traditionnelle.

[\[modifier\]](#) Propriétaire du produit

Le **Propriétaire du produit** (*Product Owner*) est le représentant des clients et des utilisateurs. Son objectif est de maximiser la valeur du produit développé.

De ce fait, cet acteur se charge de différents rôles et responsabilités.

- Il **explique** les éléments (Items) du [Carnet du produit](#). Ces composantes sont exprimées sous forme d'Histoires Utilisateur (User Stories). Son rôle est alors de rédiger les spécifications et les cas de tests qui les accompagnent.
- C'est lui qui définit l'**ordre** dans lequel les fonctionnalités seront développées. Il prend les décisions importantes concernant l'orientation du projet.
- Il s'assure que le Carnet du produit est **visible et compris de tous**. Cela permet qu'à tout instant, que chacun sache sur quoi travailler.

- Enfin, il **valide** fonctionnellement les développements.

C'est également lui qui, en accord avec l'équipe; fixe les objectifs d'un incrément (Sprint) au début de celui ci. Si ces objectifs deviennent obsolètes pendant le sprint, il a alors la responsabilité d'annuler le sprint en cours.

Dans l'idéal, le Product Owner travaille dans la même pièce que l'équipe. Il est important qu'il reste très disponible pour répondre aux questions de l'équipe et pour lui donner son avis sur divers aspects du logiciel (interface par exemple).

[\[modifier\]](#) ScrumMaster

Le **ScrumMaster** est responsable de la méthode. Il doit s'assurer que celle-ci est comprise, et bien mise en application. Ce n'est pas un chef de projet, ni un intermédiaire de communication avec les clients.

En tant que facilitateur, il aide l'équipe à déterminer quelles interactions avec l'extérieur lui sont utiles, et lesquelles sont freinantes. Il aide alors à maximiser la valeur produite par l'équipe.

Parmi ses attributions :

- Communiquer la vision et les objectifs à l'équipe
- Apprendre au directeur de produit à rédiger les composantes du carnet de produit
- Faciliter les rituels de Scrum
- Coacher l'équipe de développement
- Faciliter son intégration à l'entreprise, surtout si celle ci n'est pas pleinement agile
- Écarter les éléments pouvant perturber l'équipe
- Aider l'adoption d'Agile au niveau de l'entreprise
- Travailler avec les autres Facilitateurs/animateurs pour coordonner plusieurs équipes, s'il y a lieu.

On parle parfois d'**équipe étendue**, qui intègre en plus le *ScrumMaster* et le directeur de produit. Ce concept renforce l'idée que client et fournisseur travaillent d'un commun effort vers le succès du projet.

[\[modifier\]](#) Développeur

L'équipe ne comporte pas de rôles prédéfinis, elle est **auto-gérée**, et **pluridisciplinaire**.

Une équipe **auto-gérée** choisit la façon d'accomplir son travail, sans que ce soit imposé par une personne externe. Il n'y a pas non plus de notion de hiérarchie interne : toutes les décisions sont prises ensemble.

Une équipe **pluridisciplinaire** comporte toutes les compétences pour réaliser son projet, sans faire appel à des personnes externes à celle ci.

L'objectif de l'équipe est de livrer le produit par petits incréments. Ainsi, à tout instant, il existe une version du produit "potentiellement utilisable" disponible.

L'équipe s'adresse directement au directeur de produit. Il est conseillé qu'elle lui montre le plus souvent possible le logiciel développé pour qu'il puisse ajuster les détails d'ergonomie et d'interface par exemple.

[\[modifier\]](#) Évènements

[\[modifier\]](#) Le Sprint

Le Sprint est une période d'un mois maximum, au bout de laquelle l'équipe délivre un incrément du produit, potentiellement livrable. Une fois la durée choisie, elle reste constante pendant toute la durée du développement. Un nouveau sprint démarre dès la fin du précédent.

Chaque sprint possède un **but** et on lui associe une liste d'*élément du carnet du produit* (fonctionnalités) à réaliser.

Durant un sprint :

- Le but du sprint ne peut être modifié
- La composition de l'équipe reste constante
- La Qualité n'est pas négociable
- La liste d'items est sujette à négociations entre le *Propriétaire du produit* et les *Développeurs*.

La limitation est à un mois afin de limiter la complexité, et donc les risques liés au sprint.

Si le but du sprint devient obsolète pendant celui ci, le propriétaire du produit peut décider de l'annuler. Dans ce cas, les développements terminés sont revus par le propriétaire du produit, qui peut décider de les accepter. Tous les éléments du sprint n'étant pas acceptés sont réestimés, et remis dans le carnet du produit. Un nouveau sprint démarre alors.

[\[modifier\]](#) Mêlée Quotidienne

Au quotidien, une réunion, la **mêlée quotidienne** (*Daily Scrum*), permet aux Développeurs de faire un point de coordination sur les tâches en cours, et sur les difficultés rencontrées. Cette réunion dure 15 minutes maximum. Le Scrum Master s'assure que la réunion ait lieu à heure fixe. Le Propriétaire du produit n'y est pas présent.

A tour de rôle, chaque membre répond à 3 questions :

- *Qu'est-ce que j'ai fait hier ?*
- *Qu'est-ce que je compte faire aujourd'hui ?*
- *Quelles sont les difficultés que je rencontre ?*

Le tour de parole doit être scrupuleusement respecté pour éviter que le Scrum dérive sur des discussions techniques et déborde des 15 minutes. Si le besoin s'en fait sentir, des discussions sont alors menées librement après le Scrum.

Cette réunion a un but de synchronisation pour l'équipe et ne doit pas être vécue comme un *reporting* d'activité. C'est le niveau quotidien du principe *inspect and adapt* de Scrum.

[\[modifier\]](#) Réunion de Planification de Sprint

Tout le monde est présent à cette réunion, qui ne doit pas durer plus de 8 heures pour un sprint d'un mois. Pour un sprint plus court, la durée est réduite proportionnellement.

La **réunion de planification** (*Sprint Planning Meeting*) se passe en deux temps. Dans la première partie, l'équipe de développement cherche à prévoir ce qui sera développé durant le prochain Sprint. En entrée de cette phase, l'équipe doit avoir à sa disposition le carnet du produit priorisé, l'incrément réalisé à la dernière itération, la capacité de production de l'équipe lors des dernières itérations, ainsi que la capacité de production prévue pour la prochaine itération. L'équipe et le propriétaire du produit cherchent alors à déterminer le but du sprint.

Dans un second temps l'équipe se focalise sur la manière dont ils atteindront le but du sprint. Ils prévoient alors ce qui sera fait pendant le sprint, construisant ainsi une estimation. Cela constitue l'engagement de l'équipe.

[\[modifier\]](#) Revue de sprint

À la fin du sprint, tout le monde se réunit pour effectuer la **revue de sprint**, qui dure au maximum 4 heures. L'objectif de la revue de sprint est de valider le logiciel qui a été produit pendant le sprint. L'équipe commence par énoncer les items du carnet de produit qu'elle a réalisés. Elle effectue ensuite une démonstration du logiciel produit. C'est sur la base de cette démonstration que le directeur de produit valide chaque fonctionnalité planifiée pour ce sprint.

Une fois le bilan du sprint réalisé, l'équipe et le propriétaire de produit proposent des aménagements sur carnet du produit et sur la planification provisoire de la release. Il est probable qu'à ce moment des items soient ajoutés, modifiés ou réestimés, en conséquence de ce qui a été découvert.

[\[modifier\]](#) Rétrospective du sprint

La **rétrospective du sprint** est faite en interne à l'équipe (incluant le ScrumMaster). Elle dure 3 heures pour un sprint d'un mois, et réduit selon la durée du sprint.

L'objectif est de comprendre ce qui n'a pas bien marché dans le sprint, les erreurs commises et de prendre des décisions pour s'améliorer. Il est tout à fait possible d'apporter des aménagements à la méthode Scrum dans le but de s'améliorer. Il faut être très vigilant à ne pas retomber dans des pratiques rigides des méthodologies plus classiques.

[\[modifier\]](#) Artéfacts de Scrum

[\[modifier\]](#) Carnet du produit

Scrum utilise une approche fonctionnelle pour récolter les besoins des utilisateurs. L'objectif est d'établir une liste de fonctionnalités à réaliser, que l'on appelle **carnet du produit** (NDT : Le terme *backlog* anglais évoque aussi une *réserve*, un *retard accumulé*).

À chaque élément du carnet sont associés trois attributs : une description, une estimation (souvent exprimée en **points arbitraires**, cf [Estimation](#)) et un ordre, qui est défini par le directeur de produit. Il peut changer cet ordre en cours de projet et même ajouter, modifier ou supprimer des éléments dans le carnet. Si plusieurs équipes travaillent sur le même produit, on ajoute alors un nouvel attribut indiquant l'équipe concernée.

Il arrive souvent qu'on utilise dans Scrum les *User Stories* de la méthode [Extreme Programming](#), associées au [Planning poker](#) les estimer). [\[réf. souhaitée\]](#)

[\[modifier\]](#) **Carnet de sprint**

En début de sprint, un but est décidé. Pour atteindre cet objectif, l'équipe choisi quels éléments du carnet de produit seront réalisés. Ces éléments sont alors groupés dans un **carnet de sprint**.

Chaque équipe met à jours régulièrement le carnet de sprint. Celui ci donne donc une vision de ce que l'équipe estime nécessaire à réaliser pour atteindre l'objectif du sprint. Seule l'équipe peut le modifier à tout moment.

[\[modifier\]](#) **Incrément**

Un incrément est constitué de l'ensemble des fonctionnalités "complétées" pendant le sprint en cours, ainsi que pendant les sprint précédents. À la fin de chaque sprint, cet incrément est considéré comme complété. Il doit être utilisable, qu'il soit ou non publié.

[\[modifier\]](#) **Définition de complété**

L'objectif est de compléter une fonctionnalité ou un incrément. Selon l'équipe, la définition de ce qui est "fini fini" peut varier. Elle doit cependant être clairement affichée.

Lorsque l'équipe gagne en maturité, cette définition deviendra de plus en plus stricte.

[\[modifier\]](#) **Compléments**

[\[modifier\]](#) **Estimations**

Scrum ne définit pas spécialement d'unités pour les items des backlogs. Néanmoins, certaines techniques se sont imposées de fait.

[\[modifier\]](#) **Items de backlog de produit**

Les items de backlog de produit sont souvent des *User Stories* empruntées à [Extreme Programming](#). Ces User Stories sont estimées en points relatifs, sans unité. L'équipe prend un item représentatif et lui affecte un nombre de points arbitraire. Cela devient un référentiel pour estimer les autres items. Par exemple, un item qui vaut 2 points représente deux fois plus de travail qu'un item qui en vaut 1. Pour les valeurs, on utilise souvent les premières valeurs de la [suite de Fibonacci](#) (1,2,3,5,8,13), qui évitent les difficultés entre valeurs proches (8 et 9 par exemple).

L'intérêt de cette démarche est d'avoir une idée du travail requis pour réaliser chaque fonctionnalité sans pour autant lui donner une valeur en jours que le directeur de produit serait tenté de considérer comme définitivement acquise. En revanche, on utilise la *vélocité* pour planifier le projet à l'échelle macroscopique de façon fiable et précise.

[\[modifier\]](#) Calcul de vélocité

Une fois que tous les items de backlog de produit ont été estimés, on attribue un certain nombre d'items à réaliser aux sprints successifs. Ainsi, une fois un sprint terminé, on sait combien de points ont été réalisés et on définit alors la **vélocité** de l'équipe, c'est-à-dire le nombre de points qu'elle peut réaliser en un sprint.

En partant de cette vélocité et du total de points à réaliser, on peut déterminer le nombre de sprints qui seront nécessaires pour terminer le projet (ou la release en cours). L'intérêt, c'est qu'on a une vision de plus en plus fiable (retours d'expérience de sprint en sprint) de la date d'aboutissement du projet, tout en permettant d'aménager les items de backlog du produit en cours de route.

[\[modifier\]](#) Items de backlog de sprint

Les items de backlog de sprint sont généralement exprimés en heures et ne doivent pas dépasser 2 journées de travail, sinon il convient de les décomposer en plusieurs items. Par abus de langage, on emploie le terme de *tâches*, les concepts étant très proches.

[\[modifier\]](#) Lancement du projet

Scrum présuppose que le backlog de produit est déjà défini au début du projet. Une approche possible pour constituer ce backlog est de réaliser une **phase de lancement**. Cette phase de lancement s'articule autour de deux axes de réflexion : l'étude d'opportunité et l'expression initiale des besoins.

L'étude d'opportunité est très variable d'un projet à l'autre, tout dépend du contexte de l'entreprise, de la nature du projet (sous-traitance ou interne), etc. Chaque entreprise a sa propre politique sur cette activité.

L'expression initiale des besoins n'est pas un élément défini dans Scrum et n'est **surtout pas** une activité de contractualisation d'un cahier des charges. L'esprit d'une telle activité dans les méthodes Agiles est de définir d'une part le **cadre du projet** (pour que l'équipe s'imprègne du contexte métier) et d'autre part une **première analyse globale** des besoins. L'objectif est d'identifier un maximum de fonctionnalités que le logiciel devra implémenter, en se limitant à un niveau de précision assez grossier. On peut par exemple utiliser une approche par raffinages successifs, en partant des secteurs métiers concernés par l'application, puis en identifiant les activités métier, qu'on décompose en tâches métier qui correspondent à des fonctionnalités que l'on doit/peut ou non implémenter dans le logiciel final.

L'objectif pour Scrum est de produire la première version du backlog de produit.

[\[modifier\]](#) Vue globale

Vue synthétique du processus Scrum

[\[modifier\]](#) **Burndown Charts**

Les **burndown charts** (*graphiques d'avancement*) permettent de visualiser graphiquement l'avancement du travail. Une interprétation simple permet d'avoir une idée sur les échéances futures.

[\[modifier\]](#) **Sprint Burndown Chart**

Exemple de Sprint Burndown Chart

Ce graphique représente la quantité totale d'heures restantes à faire dans le sprint, au fil des jours. Il permet d'avoir une vue sur l'avancement du sprint.

[\[modifier\]](#) Release Burndown Chart

Exemple de Release Burndown Chart

Ce graphique représente la quantité totale de points restant à faire dans la release, au fil des sprints. Il permet d'avoir une vue sur l'avancement de la release.

[\[modifier\]](#) Interprétation

Ces graphiques sont très intéressants à analyser et interpréter. Outre le fait de montrer l'avancement concret du travail, ils permettent d'anticiper de façon relativement fiable les échéances futures en cours du sprint ou de la release.

On peut observer de légères augmentations du reste à faire sur le burndown chart du sprint. Cela correspond généralement à une réestimation à la hausse, suite à la prise en compte de contraintes techniques que l'on n'avait pas vues lors de l'estimation initiale. Si c'est le cas, il est indispensable de comprendre la cause de ces augmentations. Le même phénomène peut s'observer sur des légères diminutions, pour les mêmes raisons.

On peut utiliser en cours du sprint la tendance de la courbe pour avoir une idée approximative de la fin du sprint. Cela consiste à prendre un segment de droite dont la pente est représentative des valeurs déjà recensées et de le prolonger jusqu'à son point d'intersection avec l'axe des abscisses. Cela nous donne alors la date *a priori* de la fin du sprint et nous permet alors de prendre une décision sur la suppression (ou l'ajout) d'un item de backlog du produit à réaliser dans ce sprint. C'est ce qui s'est probablement passé le 15 mai 2002 sur le graphique de sprint ci-dessus : le reste à faire diminue dans ce cas assez brutalement.

C'est exactement la même chose pour les burndown charts de release. Si la date de publication de la release est clairement au-delà de ce que l'on espérait, on peut aviser en enlevant des items de backlog du produit ou changeant leur ordre, de sorte que les fonctionnalités les moins importantes soient celles qui risquent de ne pas être développées à temps.

Cette approche, bien que basée sur des tendances approximatives, permet d'identifier très tôt les risques de défaillance et d'agir en conséquence, en conservant à l'esprit le caractère critique des fonctionnalités à développer. Ces décisions importantes relèvent complètement du directeur de produit.

Une dernière chose importante : la fiabilité de la vélocité de l'équipe et des estimations qu'elle a faites augmente au fil des sprints. On élimine de cette façon les risques majeurs au plus tôt dans le projet.

[\[modifier\]](#) Qualité de l'environnement de travail

Un concept fort de Scrum est la qualité de l'environnement de travail de l'équipe. Cela inclut :

- Pas de changements imposés pendant un sprint
- Toute l'équipe dans une même pièce
- Un tableau blanc et/ou en liège
- Un bon outil de suivi du projet
- Prévenir des interventions extérieures (téléphone, irruption dans la pièce, etc.)
- Tout ce qui peut rendre l'équipe plus sereine et efficace

[\[modifier\]](#) Risques de la méthodologie

Dans Scrum, les risques, en particulier de dérives sont multiples. Ils ne sont pas présents et dans la même proportion dans tout projet. Néanmoins, ces risques deviennent de plus en plus courant avec la montée de Scrum. Le risque majeur dans l'utilisation de Scrum est avant tout une application allant à l'encontre même des principes fondateurs de son fonctionnement. On peut en citer deux, mais il en existe bien d'autres :

- un Scrum Master directif.
- La communication constructive laisse la place aux reproches.

[\[modifier\]](#) Un Scrum Master directif

L'un des plus graves risques de Scrum est la violation du principe même de son fonctionnement qui met l'équipe de développement au cœur même du processus. Il n'est pas rare que le Scrum Master soit un chef de projet déguisé ou officiel et qu'il applique un contrôle trop strict sur les membres de son équipe^[5].

Dans ce contexte, on demande aux développeurs de s'impliquer comme s'ils étaient responsables du projet mais en réalité ils ne le sont pas, puisque le Scrum Master qui est le chef de projet a le dernier mot pour tout et peut effectuer des choix arbitraires. Ce qui peut apporter des frustrations côté développeur. En outre, dans ce type de configuration, le Scrum Master qui n'en est plus un, n'endosse plus les responsabilités du Scrum Master : être le protecteur de l'équipe, objectif, garde-fou, garant de la méthodologie appliquée, etc... Les bénéfices de l'agilité peuvent ainsi

disparaître : défiance de l'équipe vis-à-vis du Scrum Master, conflit à l'intérieur même de l'équipe, non-dit, désengagement qui engendre baisse de la qualité et absence de bonnes pratiques.

[\[modifier\]](#) La communication constructive laisse la place aux reproches

La surexposition dans Scrum prend sa source dans les différentes réunions d'équipe telles que la quotidienne matinale, censée informer et apporter des solutions aux problèmes de la veille. Sur certains projets, ces réunions se transforment en blâme des membres n'ayant pas réussi à mener à bien leurs tâches de la veille. Les conséquences dans ce cas peuvent prendre la forme de discussions houleuses et ou de conflits plus ou moins durables entre membres de l'équipe. On constate aussi parfois à l'inverse une manière différente d'appréhender la réunion quotidienne : non-dit ou propos vagues en ce qui concerne les problèmes rencontrés. Ainsi, les réunions peuvent se passer de façon plus courtoises mais un risque de ne plus chercher le support de l'équipe en cas de difficultés se développe. Ce qui va à l'encontre même des principes de Scrum.

À l'inverse, sur les projets cycle en V, on dispose d'un effet de lissage, qui permet de compenser un jour ce qu'on a moins bien fait la veille. En outre, les difficultés rencontrées un jour par un membre de l'équipe ne sont pas mises en avant aux yeux de tous. Le membre en question peut trouver une solution seul le lendemain ou demander de l'aide auprès de collègues. En clair, il a le choix de choisir sa solution et de ne pas être potentiellement rabaissé par les autres membres de l'équipe.

[\[modifier\]](#) Documentation de projet

Scrum n'impose aucune documentation particulière pour les projets. Des documents sont implicitement produits (backlogs, burndown charts), mais ils ont une vocation avant tout utilitaire.

Produire de la documentation est souvent utile mais aussi souvent inutile. En plus, il faut la maintenir à jour, quelque chose qui est rarement fait sur place. Pour savoir s'il faut rédiger un document, on peut se poser une question très simple : **Est-ce que ce document va m'être vraiment utile et tout de suite ?**

Voici quelques exemples de documents utiles et dans quels cas :

- diagrammes métiers (processus, objets, etc.), associé au backlog de produit : uniquement si la logique métier du client qui concerne l'application est vraiment complexe. Dans ce cas, l'équipe devrait produire ce document avec lui ;
- diagramme de séquence, associé à un item du backlog du produit : uniquement si la fonctionnalité aura une utilisation complexe, tant au niveau métier qu'applicatif ;
- diagrammes d'architecture du logiciel (classes, modules, composants, etc.), pour le projet : indispensable pour avoir toujours sous les yeux une vue de l'architecture et s'assurer ainsi qu'elle est de qualité ;
- les manuels utilisateur, à chaque sprint : les manuels sont produits à chaque sprint et pas en fin de projet. Utiliser des vidéos de démonstrations commentées est une solution efficace ;
- les FAQ pour le centre d'appel : des cas classiques où les utilisateurs ne vont pas comprendre un comportement métier. Cela permet de traiter un maximum de problèmes au niveau du centre d'appel, avant que cela n'arrive aux équipes de

développement/maintenance.

Bref, un document ne doit être produit que si **son utilité est réelle et immédiate**.

[\[modifier\]](#) Outils pour Scrum

Un bon artisan n'est rien sans un bon outil. Il n'y a pas aujourd'hui d'outil *ultime* pour Scrum. Voici un petit panel des possibilités.

[\[modifier\]](#) Papier et crayon / tableur

Scrum peut être mis en pratique avec trois fois rien : deux listes suffisent. La liste des items du backlog de produit et la liste des items du backlog de sprint. La saisie et la mise à jour des données est simplement un peu moins agréable.

[\[modifier\]](#) Autres outils

- Voir les [ressources ScrumAlliance](#).

[\[modifier\]](#) Conclusion

[\[modifier\]](#) Scrum

Scrum est un processus de développement de logiciels qui s'intéresse plutôt à l'organisation du projet qu'aux aspects techniques. Son cadre de travail est sa force, si bien que cette méthode pourrait être appliquée à d'autres domaines. Son approche incrémentale et basée sur les besoins priorités du client lui confèrent une flexibilité extrême. Elle incarne bien par là l'état d'esprit de la mêlée de rugby : avancer tous ensemble vers un but commun, la réussite du projet.

Les pratiques de Scrum sont essentiellement orientées sur la maîtrise d'une livraison d'incréments (sprint) mais réfutent la possibilité de modifier les fonctionnalités en cours de réalisation (dans le backlog de sprint). Cette limite interdit la mise en œuvre d'une conception émergente comme celle d'XP ainsi que celle d' **itération**, base de l'adaptabilité (possibilité d'affinement par modification permanente). Scrum, ne disposant pas de métrique de gestion du changement à ce niveau, nécessite donc une importante spécification préalable à la mise en production (backlog produit) et, du fait de cette prédictibilité imposée, ne peut pas être considéré comme réellement itératif, donc adaptatif.

Scrum est un processus intéressant comme premier pas vers les méthodes Agiles : il est facile à comprendre et à pratiquer. La seule difficulté relève plutôt de l'environnement organisationnel (voir la mise en garde ci-dessous).

[\[modifier\]](#) Mise en garde

On entend de plus en plus de sociétés clamer qu'elles sont Agiles, comme argument commercial à la mode, parce qu'elles utilisent quelques pratiques des méthodes Agiles. Mais **être Agile**, c'est bien plus que de mettre en pratique une méthode. L'Agilité requiert des dispositions particulières qui sont loin d'être faciles à mettre en place :

- Une **organisation adaptée** : il est très difficile de convaincre les organes décisionnels d'une entreprise de travailler de façon Agile. En effet, cela signifie de ne pas disposer dès le départ d'une date et d'un budget très précis, mais plutôt d'un ordre de grandeur.
- Un **état d'esprit** : être Agile, c'est privilégier l'esprit d'équipe et pas seulement dans la réalisation technique. Le client doit comprendre que l'on attend de lui un investissement personnel bien supérieur à celui des méthodes plus classiques, en testant le logiciel souvent et en participant à toutes les réunions.
- Un **environnement favorable** : éliminer les contraintes dans une entreprise n'est pas toujours évident. Comment limiter les appels téléphoniques trop fréquents, les interruptions dans la pièce de l'équipe, les opérations "urgentes" demandées aléatoirement par la direction, etc. ?

Bref, utiliser une méthode comme Scrum au niveau de l'équipe technique ne suffit pas. L'Agilité est une façon de travailler différente de ce dont on a l'habitude, c'est l'attitude du changement, de la flexibilité, de l'adaptation et de l'amélioration continue. Ce n'est pas aussi simple qu'une méthode...

[\[modifier\]](#) Glossaire

Directeur de produit (*Product Owner*)

personne responsable de produire et maintenir à jour le backlog de produit. C'est lui qui en détermine les priorités et qui prend les décisions concernant l'orientation du projet.

Chef de mêlée (*ScrumMaster*)

membre de l'équipe dont l'objectif principal est de la protéger des perturbations extérieures. Il est complètement transparent pour la communication entre l'équipe et les clients et n'a aucun pouvoir hiérarchique sur l'équipe. C'est en revanche un facilitateur pour les problèmes non techniques de l'équipe.

Backlog de produit (*Product Backlog*)

liste des fonctionnalités qui devront être réalisées par le logiciel.

Backlog de sprint (*Sprint Backlog*)

liste des tâches à accomplir pendant un sprint. Elles correspondent à la réalisation des items de backlog du produit affectés au sprint.

Mêlée quotidienne (*Daily Scrum*)

réunion quotidienne de 15 minutes qui a pour but de faire le point sur ce qui a été fait depuis la dernière mêlée, ce qui est prévu de faire jusqu'à la prochaine et quelles sont les embûches rencontrées durant le travail.

Sprint

nom d'une itération dans Scrum. Cette itération dure 30 jours calendaires en théorie, mais en pratique on utilise plutôt entre 2 et 4 semaines. Pendant une itération, l'équipe doit développer une liste d'items du backlog de produit qui a été définie au début de ce sprint.

Graphique d'avancement (*Burndown Chart*)

graphique qui montre la tendance du reste à faire total de jour en jour (pour les sprints) ou de sprint en sprint (pour les releases).

[[modifier](#)] Voir aussi

[[modifier](#)] Articles connexes

- [Cycle de développement](#)
- [Méthode agile](#)
- [Extreme programming](#)

[[modifier](#)] Liens externes

Cet article ou cette section a trop de [liens externes](#).

Les liens externes doivent être des [sites de référence](#) dans le [domaine du sujet](#). Il est souhaitable — si cela présente un intérêt — de [citer ces liens comme source](#) et de les enlever du corps de l'article ou de la section « *Liens externes* ».

- **([en](#))** [Site officiel Scrum](#)
- **([en](#))** [Scrum Alliance](#) : Communauté autour de Scrum
- **([en](#))** [Liste de discussion par mail](#)
- **([fr](#))** [Liste de discussion par mail francophone](#)
- **([fr](#))** [Le French Scrum User Group](#) : le groupe d'utilisateurs de Scrum en France.

[[modifier](#)] Bibliographie anglophone

- *Agile Software Development With Scrum* , Ken Schwaber, Mike Beedle, (Prentice Hall, October 21, 2001) ([ISBN 978-0130676344](#))
- *The Power of Scrum* , Jeff Sutherland, Rini van Solingen, Eelco Rustenberg, (Kindle Edition, Nov 10, 2011) ([ASIN](#) B007474YMC)

[[modifier](#)] Bibliographie francophone

De 2007à 2009, Jean-Pierre Vickoff publie trois ouvrages présentant Scrum et XP ainsi que les techniques permettant de les coupler et de les compléter :

- *AGILE, l'Entreprise et ses projets* ,Jean-Pierre Vickoff, QI, 2007. ([ISBN 978-2912843005](#))
- *PUMA Essentiel, Méthode Agile Optimale*,Jean-Pierre Vickoff, QI, 2008. ([ISBN 978-2912843067](#))
- *Méthode Agile, Les meilleures pratiques, Compréhension et mise en œuvre*,Jean-Pierre Vickoff, QI, 2009. ([ISBN 978-2912843074](#))

Le 21 janvier 2009 Guillaume Mathias, Bruno Orsier, Emmanuel Etasse et Christophe Bunn traduisent en français l'ouvrage de Henrik Kniberg "Scrum and XP from the Trenches" ^[6].

En février ^[7]2010,Claude Aubry publie « SCRUM : Le guide pratique de la méthode agile la plus populaire »^[7].

[[modifier](#)] Notes et références

- ↑ *The New New Product Development Game* Havard Business Review, jan-fév 1986
- ↑ *The New New Product Development Game* Havard Business Review, jan-fév 1986
- ↑ **(en)** DeGrace, Stahl et Hulet, *Wicked problems, righteous solutions*, Prentice Hall, 1^{er} octobre 1990 (ISBN 978-0-135-90126-7)
- ↑ "Three pillars uphold every implementation of empirical process control: transparency, inspection, and adaptation." ([Guide officiel de Scrum](#))
- ↑ *Empowering Teams: The ScrumMaster's Role*, 11 juin 2007
- ↑ http://www.infoq.com/minibooks/scrum-xp-from-the-trenches
- ↑ *SCRUM : Le guide pratique de la méthode agile la plus populaire*

- [Portail de l’informatique](#)

Ce document provient de «

[http://fr.wikipedia.org/w/index.php?title=Scrum_\(méthode\)&oldid=79307674](http://fr.wikipedia.org/w/index.php?title=Scrum_(méthode)&oldid=79307674) ».

Catégories :

- [Gestion de projet](#)
- [Méthode de développement logiciel](#)

Catégories cachées :

- [Article à référence souhaitée](#)
- [Liens externes à trier](#)
- [Portail:Informatique/Articles liés](#)

Outils personnels

- [Créer un compte ou se connecter](#)

Espaces de noms

- [Article](#)
- [Discussion](#)

Variantes

Affichages

- [Lire](#)
- [Modifier](#)
- [Afficher l’historique](#)

Actions

Rechercher

Navigation

- [Accueil](#)
- [Portails thématiques](#)
- [Index alphabétique](#)
- [Article au hasard](#)
- [Contacter Wikipédia](#)

Contribuer

- [Premiers pas](#)
- [Aide](#)
- [Communauté](#)
- [Modifications récentes](#)
- [Faire un don](#)

Imprimer / exporter

- [Créer un livre](#)
- [Télécharger comme PDF](#)
- [Version imprimable](#)

Boîte à outils

- [Pages liées](#)
- [Suivi des pages liées](#)
- [Importer un fichier](#)
- [Pages spéciales](#)
- [Adresse de cette version](#)
- [Citer cette page](#)

Autres langues

- [Đ·ŃŠĐ»Đ³Đ°Ń€Ń•Đ°Đ,](#)
- [Català](#)
- [Dansk](#)
- [Deutsch](#)
- [English](#)
- [Español](#)
- [ĭ¹/ĭ¹³ĭ¹-ĭ¹Žĭ»“](#)
- [Suomi](#)
- [×ª×™×·×´×ø](#)
- [Magyar](#)
- [Italiano](#)
- [æ—Ÿæœ-èªž](#)
- [í•œêµ-ì-´](#)
- [LietuviÅ³](#)
- [Nederlands](#)

- [Norsk \(bokmål\)](#)
- [Polski](#)
- [Português](#)
- [Română](#)
- [Srpskohrvatski / Српскохрватски / Srpski](#)
- [Svenska](#)
- [Türkçe](#)
- [Українська](#)
- [Español](#)
- Dernière modification de cette page le 1 juin 2012 à 09:41.
- **Droit d'auteur** : les textes sont disponibles sous [licence Creative Commons paternité partage à l'identique](#) ; d'autres conditions peuvent s'appliquer. Voyez les [conditions d'utilisation](#) pour plus de détails, ainsi que les [crédits graphiques](#). En cas de réutilisation des textes de cette page, voyez [comment citer les auteurs et mentionner la licence](#).
Wikipedia® est une marque déposée de la [Wikimedia Foundation, Inc.](#), organisation de bienfaisance régie par le paragraphe [501\(c\)\(3\)](#) du code fiscal des États-Unis.
- [Politique de confidentialité](#)
- [À propos de Wikipédia](#)
- [Avertissements](#)
- [Version mobile](#)
-
-