

Le Framework Scrum : Un Cadre De Travail Pour Réduire Le Risque Et Délivrer De La Valeur Plus Tôt

Un aperçu du cadre de travail (framework) Scrum, pour les nouveaux convertis à Scrum et ceux qui désirent en rafraîchir leur compréhension.



**The
Liberators**

Dans ce livre blanc



Avant-propos

Rappel historique

Les Piliers de l'Empirisme

Scrum : un cadre de travail pour le contrôle de processus empirique

Les Artefacts Scrum

Les Cérémonies Scrum

Les Responsabilités de Scrum

Deux Principes Directeurs

Les Cinq Valeurs Fondamentales qui Permettent l'Empirisme

Ensemble, un Cadre de Travail Simple

Avant-propos



Il nous arrive souvent de dire aux gens "nous nous fichons de Scrum". Cela provoque assurément des surprises, mais c'est notre façon de dire que l'important n'est pas le cadre Scrum, mais plus ce qu'il rend possible. En prenant cet angle de vue, de nombreuses questions théoriques deviennent évidentes ou caduques. "Est-ce que toutes les items du Sprint Backlog doivent avoir un lien avec le Sprint Goal?", "Est-ce qu'un bug doit intégrer le Product Backlog ?" Ou "Dois-je stopper le daily dès qu'il atteint 15 minutes ?" Il est aisément possible de se perdre dans ces détails quand on oublie (ou qu'on ne voit pas) la situation dans son ensemble.

Nous aimons le Framework Scrum pour ce qu'il rend possible. Le Scrum Guide officiel fait déjà un boulot considérable dans la compréhension du cadre et ce de manière concise. Nous voulons ajouter à cela une explication avec nos propres mots, un peu comme nous le faisons en travaillant avec nos Scrum Teams et en tant que Formateurs Professionnels Scrum pour Scrum.org. Notre but était d'écrire de manière pratique et concrète ce que, de notre point de vue, le cadre Scrum rend possible. Nous espérons qu'il sera facile à lire, qu'il dissipera les confusions potentielles et enrichira votre connaissance de Scrum.

Nous avons rédigé ce livre blanc avec Johannes Schartau qui a également contribué à notre livre 'The Zombie Scrum Survival Guide' (Le guide de survie au Zombie Scrum).



Rappel historique

Le [cadre de travail Scrum](#) a été développé par Ken Schwaber et Jeff Sutherland dans les années 90. Il est formalisé pour la première fois en 1995, pour répondre à la complexité inhérente aux produits et logiciels de développement. Par la suite, Scrum a été déployée avec succès pour résoudre des problèmes complexes dans une grande variété de domaines différents, allant du marketing au changement organisationnel, ou encore de la recherche scientifique au développement logiciel. Le cadre Scrum, partout où il est utilisé, repose sur 3 piliers fondamentaux qui permettent un contrôle de processus empirique.

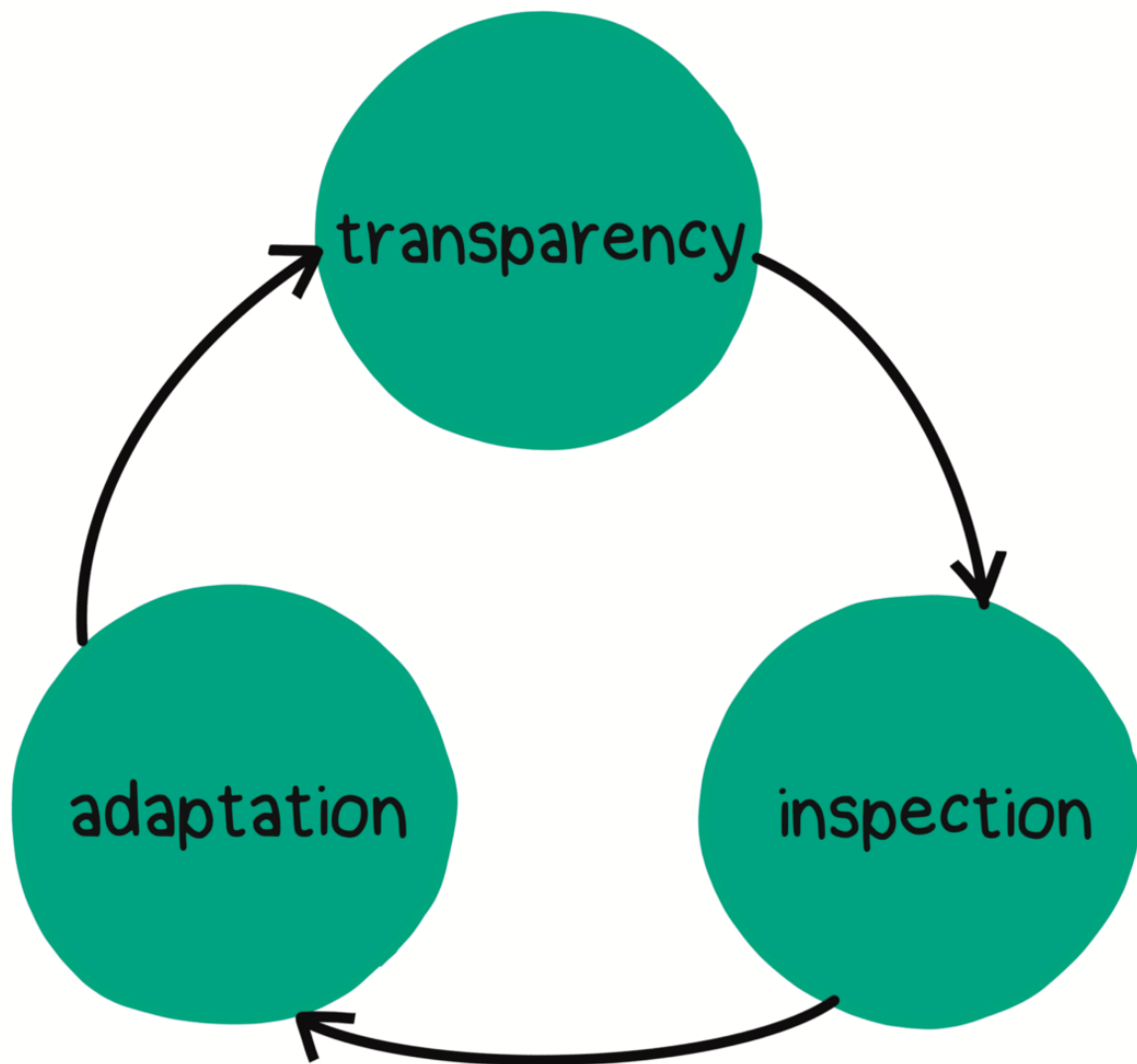
Les Piliers de l'Empirisme

Le cadre de travail Scrum s'appuie sur 3 piliers qui permettent un contrôle de processus empirique (processus qui favorisent qu'on agisse sur la base de faits, d'expériences, et de preuves) :

- **Transparence** : vous collectez des données – comme des métriques, des commentaires, et autres expériences – pour savoir ce qui se passe ;
- **Inspection** : vous inspectez régulièrement les processus avec toutes les personnes engagées et vous décidez ce que cela implique pour vos objectifs ;
- **Adaptation** : vous faites les changements nécessaires à l'atteinte de vos objectifs.

Ce cycle se répète aussi souvent que nécessaire pour déceler les écarts, les découvertes inattendues, et les opportunités potentielles qui émergent au fur et mesure que le travail avance. Cela ne doit pas arriver qu'une fois par an ou une fois que le projet est terminé mais plutôt en continu (sur une base quotidienne, hebdomadaire ou mensuelle). Plutôt que de prendre des décisions basées sur des hypothèses à propos de futurs potentiels, vous allez prendre des décisions basées sur les données que vous avez collectées jusque là. C'est empirique. Et sur ce point, vous allez découvrir comment tout dans Scrum est conçu autour de ces 3 piliers.

"Plutôt que de prendre des décisions basées sur des hypothèses à propos de futurs potentiels, vous allez prendre des décisions basées sur les données que vous avez collectées jusque là."

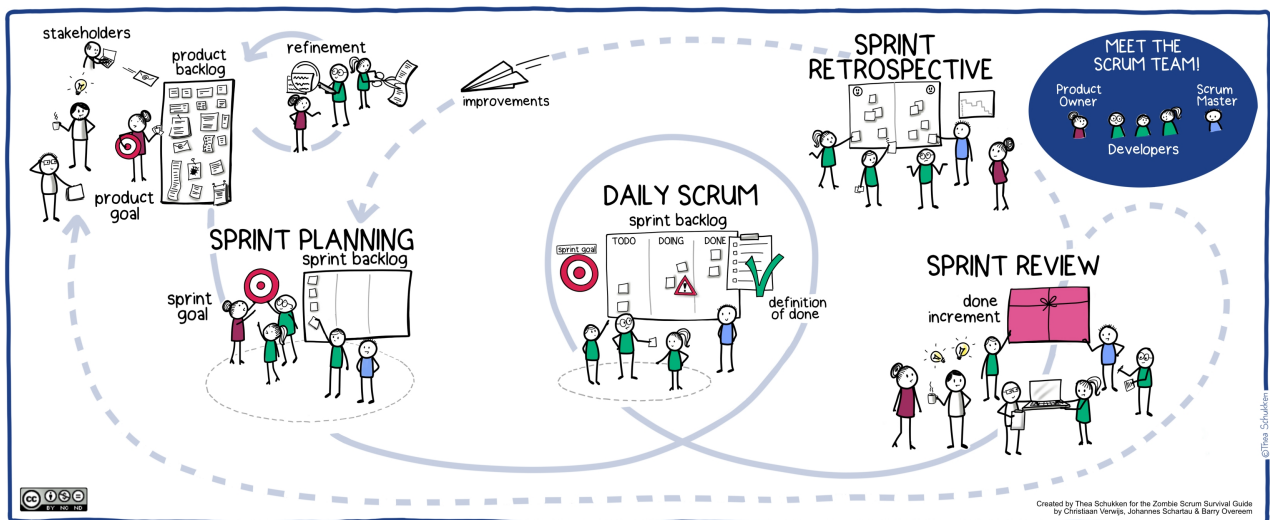


Created by Thea Schukken for the Zombie Scrum Survival Guide
by Christiaan Verwijs, Johannes Schartau & Barry Overeem



Scrum : un cadre de travail pour le contrôle de processus empirique

C'est une chose de dire que vous avez besoin de transparence quand vous développez un produit, que devez souvent inspecter votre travail et que les idées qui en découleront vont favoriser l'adaptation. C'est une autre de mettre cela en pratique de manière tangible et concrète. C'est cela qui fait de Scrum un cadre de travail ; articulé autour de cinq événements répétitifs pour travailler sur trois artefacts, Scrum s'appuie sur trois responsabilités (rôles) ainsi que plusieurs principes et règles qui mis tous ensemble créent un tout cohérent.



Les Artefacts Scrum

Le premier pilier du contrôle de processus empirique est la 'Transparence'. Afin d'inspecter fréquemment la progression du travail sur un produit et de prendre des décisions à propos de ce qui est (encore) nécessaire, nous avons besoin d'avoir quelque chose qui peut être inspecté. Quand on parle de travailler sur un produit 'transparent', on entend le rendre disponible sous une forme qui permet aux personnes qui ont un intérêt pour lui de le tester, on pourra ainsi valider les hypothèses en l'utilisant et en faire émerger de nouvelles idées.

Scrum contraint les équipes à rendre transparents au moins 3 éléments de leur travail sur le produit. Nous appelons cela : les 'artefacts'. Ce sont les principaux vecteurs de collecte des données et d'expériences dont nous avons besoin pour éclairer de notre prise de décision pour l'avenir.

Le Backlog Produit

Le premier artefact est le Product Backlog (ou Backlog Produit). Il met en évidence, en transparence, l'ensemble du travail encore nécessaire à réaliser pour le produit pour réaliser l'objectif Produit (Product Goal). Le Product Goal met l'accent sur le travail sur le Product Backlog en permettant à l'équipe Scrum de s'engager sur un seul et unique objectif à long terme pour le Produit. Au fur et à mesure que l'équipe Scrum s'approche de l'objectif Produit, le Product Backlog change pour refléter les nouvelles idées et opportunités qui émergent en travaillant. Un nouvel objectif Produit ne peut être créé que si le précédent a été achevé ou abandonné, sinon la concentration et l'engagement en souffriront.

Chaque idée, hypothèse, fonctionnalité, bogue ou tâche que l'équipe Scrum juge nécessaire pour atteindre l'objectif de Sprint est représenté dans le Product Backlog comme une item unique. Certaines seront très vastes et peu claires, quand d'autres seront spécifiques et petites. Toutes les items sont ordonnées en fonction de leur pertinence par rapport aux ambitions pour le produit et celles de ses parties prenantes.

Un produit a un seul Product Backlog avec un Product Goal, quelque soit le nombre d'équipes qui travaillent dessus. À défaut, c'est la transparence et la concentration qui en souffriront car il n'y aura pas une seule et unique "source de vérité" à propos du travail nécessaire et de l'ordre dans lequel il devra être fait.

Le Backlog de Sprint

Le deuxième artefact est le Sprint Backlog (ou Backlog de Sprint). C'est un ensemble d'items que l'équipe de Développement sélectionne dans le Product Backlog qu'ils considèrent comme nécessaires pour atteindre l'Objectif du Sprint (Sprint Goal) auquel ils se sont engagés sur ce Sprint. Le Sprint Backlog rend transparent tout le travail sur lequel l'équipe Scrum est en train de travailler ou va travailler au cours du sprint. Chaque équipe Scrum a son propre Sprint Backlog, même quand le travail sur un produit unique est partagé entre plusieurs équipes. Le Sprint Backlog n'est pas statique et des changements peuvent intervenir au fur et à mesure des apprentissages de l'équipe au cours du Sprint. En collaboration étroite avec le Product Owner, les Développeurs peuvent ajouter ou retirer des items en fonction du temps restant dans le sprint et de la pertinence ou la nécessité de ces items pour atteindre le Sprint Goal.

L'Incrément

Le troisième artefact est l'Incrément. Chaque Incrément représente un jalon en direction de l'objectif Produit global. Dans un sprint, un increment est créé à chaque fois qu'un item du backlog Produit est achevé dans le sprint Backlog. Pour éviter la confusion et les attentes divergentes, un item est considéré comme terminé uniquement quand il est totalement validé et qu'il répond à la Definition of Done sur laquelle l'équipe Scrum s'est engagée. Les équipes Scrum peuvent créer plusieurs Incrément durant un Sprint et peuvent même les délivrer à leur Partie prenantes avant la revue de sprint.

Le but de chaque Incrément est de valider les hypothèses à propos du travail qui a été terminé à ce jour. Les personnes qui ont un intérêt pour le produit (parties prenantes, utilisateurs...) peuvent ainsi l'examiner et déterminer si cela répond à leurs besoins, si ils comprennent comment cela fonctionne et si cela satisfait à leurs attentes. Cet Incrément est une source de nouvelles idées, car avoir quelque chose de tangible avec lequel interagir est un bon moyen pour les gens de voir de nouvelles perspectives.

L'Incrément est le moyen principal par lequel vous pourrez exercer le contrôle de processus empirique pour des problèmes complexes. Chaque Incrément commence par une intuition, une hypothèse, ou une idée potentiellement précieuse de la façon dont le produit peut s'approcher de ses ambitions. Le Sprint Goal (objectif de Sprint) résume cela. Il correspond au travail nécessaire pour atteindre l'objectif du sprint Backlog.

Si il y a plusieurs équipes Scrum qui travaillent sur un unique produit, il faut s'assurer de bien intégrer leur travail dans un unique Incrément pour chaque sprint afin que l'inspection puisse t'avoir lieu.

Quoi d'autre ?

Bien entendu, il y a de nombreux autres éléments du travail de l'équipe que l'on peut rendre transparent. Par exemple, vous pouvez créer une carte des parties prenantes pour donner du sens à propos de qui ils sont et où ils se trouvent. Vous pouvez rassembler des métriques sur un tableau de bord et les inspecter fréquemment. Ce sont de bons artefacts, dignes d'inspection. Mais si on veut construire des produits de manière empirique, le cadre Scrum recommande aux équipes de commencer à créer la transparence avec les 3 artefacts de base que sont le Product Backlog, le Sprint Backlog et l'Incrément.



Les Cérémonies Scrum



Afin d'éclairer vos décisions sur les suites à donner, vous devez fréquemment donner du sens au Product Backlog, au Sprint Backlog et à l'Incrément en présence de toutes celles et ceux qui ont un intérêt avec le produit. C'est là tout le sens du mot 'Inspection'. Et cela prend plusieurs formes. Nous pouvons donner du sens au Product Backlog en l'observant et en tirant des conclusions – comme "c'est trop long", "nous ne devrions pas faire cette fonctionnalité avant celle-là" ou "quand pouvons-nous livrer ceci à une partie prenante ?". Nous pouvons donner du sens à l'Incrément en le mettant entre les mains d'utilisateurs et en validant nos hypothèses avec eux – comme "Les utilisateurs comprennent-ils cette nouvelle fonctionnalité ?" ou "cette fonctionnalité répond-elle au besoin ?". Et nous pouvons aussi donner du sens au Sprint Backlog en nous assurant que son contenu est réalisable et qu'il reflète notre plan pour le sprint en cours.

Quelque soit la forme qu'elle prend, l'inspection offre les meilleures bases de décisions quand elles sont établies sur des éléments tangibles et concrets. Dans le cas du développement d'un produit, inspecter une fonctionnalité terminée et déployée est le meilleur moyen de réellement valider vos hypothèses à propos de cette fonctionnalité et de comment elle est utilisée. Voir une présentation ou une proposition de design pour une fonctionnalité peut sembler utile, mais toutes celles et ceux en demande continueront d'avoir de nombreuses – et probablement différentes – hypothèses à propos de comment cela marchera quand elle sera implémentée.

Pour travailler de manière empirique, le cadre Scrum suggère que les équipes devraient avoir au moins cinq moments répétés pour que l'inspection ait lieu. Chacun d'eux a une boîte de temps (timebox) spécifique et offre des perspectives spécifiques sur le travail qui est en train d'être mené. Ensemble, ces moments offrent un panorama complet. Ce sont les cinq Cérémonies Scrum. Bien qu'elles soient souvent nommées 'réunions' – même dans le Scrum Guide – elles ne sont pas conçues comme des réunions au sens traditionnel du terme, où un groupe de personnes s'assoient autour d'une table et s'ennuient à mourir. En réalité, quand les Cérémonies Scrum sont menées en accord avec leur objectif respectif, le besoin d'autres réunions formelles diminue et laisse place à une collaboration naturelle et à une coordination instantanée au fur et à mesure du travail de l'équipe.

Le Sprint

Le premier événement est la timebox du Sprint lui-même. Le but de chaque Sprint est de résoudre une petite partie d'un problème complexe, un peu comme quand on résout un puzzle compliqué en commençant par de petites zones. Cette solution partielle représente une version incrémentale du produit – l'Incrément – qui peut être inspectée pour décider de ce qui devrait se passer ensuite. Cet Incrément devrait au minimum être dans un état où les parties prenantes peuvent le voir aussitôt après la fin du Sprint dès que le Product Owner aura décidé de leur donner accès. Mieux encore, les équipes en capacité à livrer tout au long du sprint, accélèrent leur capacité à apprendre plus encore. Bien que le succès des Sprints varie – même si un simple Sprint implique beaucoup de travail complexe et imprévisible – on en apprendra toujours beaucoup sur le puzzle malgré tout.

"Le but de chaque Sprint est de résoudre une petite partie d'un problème complexe, un peu comme quand on résout un puzzle compliqué en commençant par de petites zones."



Les Sprints devraient toujours être de la même durée pour créer une cadence et une prévisibilité pour l'équipe et ses parties prenantes. Quand le Sprint devient trop long, des opportunités précieuses peuvent se perdre pour valider les hypothèses et être sûr que le travail progresse toujours dans la bonne direction. Plus la longueur du Sprint s'accroît, plus le risque de perdre du temps à construire de mauvaises choses est grand. Le cadre Scrum ne spécifie pas la longueur des Sprints, à l'exception du fait qu'ils ne doivent pas excéder un mois. C'est à l'équipe de décider à quelle vitesse ils/elles peuvent, et veulent apprendre. D'une façon générale, les Sprints doivent être aussi courts que possible tout en permettant à l'équipe de Développement de délivrer une version significative du produit permettant d'atteindre l'objectif du Sprint.

Ainsi, si le Sprint est une opportunité limitée dans le temps d'explorer une partie particulière d'un problème complexe dans le développement d'un produit, les 4 autres événements Scrum donnent des opportunités spécifiques de promouvoir l'inspection et l'adaptation au sein du Sprint.



La Planification du Sprint

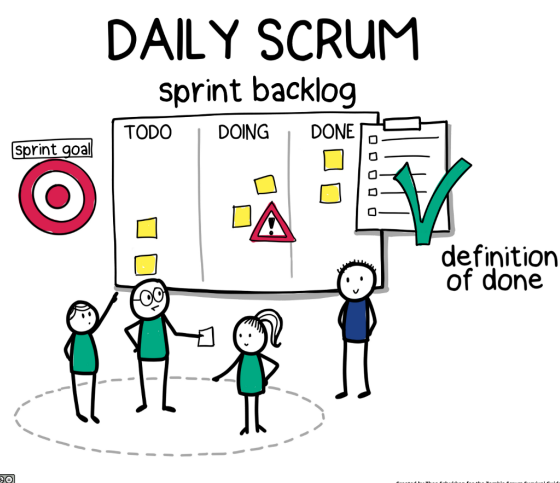
Chaque Sprint dans Scrum démarre avec un moment où l'équipe Scrum fait un plan basique pour le Sprint à venir. C'est le Sprint Planning (ou Planification du Sprint). La première et la plus importante question est quel sera l'objectif du Sprint (ou Sprint Goal). Sans objectif, il n'y aura pas de but clair pour encourager l'équipe, pour les unir et les rassembler pendant le Sprint. Un objectif pour un Sprint peut être de fournir un ensemble cohérent de fonctionnalités qui résout un problème particulier. Cela peut être un moyen d'adresser les besoins d'un groupe de parties prenantes. Cela peut aussi être une façon pour le Product Owner de vérifier ou non une hypothèse avec l'Incrément qui sort du Sprint.

Bien que le Product Owner arrive au Sprint Planning avec un objectif en tête, l'équipe Scrum travaille collectivement à affiner cet objectif afin qu'il soit autant faisable que de valeur pour les utilisateurs dans le temps imparti du Sprint. Les Développeurs travaillent avec le Product Owner à sélectionner dans le Product Backlog le travail dans le Product Backlog, qui devra être fait au cours du Sprint pour atteindre l'objectif, en réordonnant le Product Backlog si nécessaire. Cette sélection devient le Sprint Backlog. Comme les Développeurs font le travail, ils sont détenteurs du Sprint Backlog et ont le dernier mot sur ce qui va le composer.

Pour des Sprints d'un mois, le Sprint Planning devrait ne pas excéder plus de 8 heures. Mais pour un Sprint plus court, il prendra moins de temps. Le Sprint Planning est terminé quand on a un aperçu du Sprint à venir et un plan plus détaillé pour les 2 premiers jours. Ce plan est en général une décomposition du travail des premiers jours du Sprint. Au fur et à mesure que le Sprint progresse, les Développeurs traduisent l'ébauche – tel que défini dans le Sprint Goal et le Sprint Backlog – en un plan plus concret sur la façon de travailler collectivement à y parvenir. Au plus tard, cela devra se faire en Daily meeting.

Le Daily Scrum

Le Daily Scrum, ou Daily Stand-up ou point quotidien, a lieu toutes les 24 heures pour permettre aux Développeurs d'éclaircir les complexités embarquées dans le Sprint. Ils/Elles inspectent la progression de leur travail pour atteindre le Sprint Goal, rendu transparent par leur Sprint Backlog, et apportent des ajustements en conséquence. Par exemple, ils/elles peuvent avoir rencontré des problèmes inattendus qui requièrent une collaboration rapprochée ou bien découvrir qu'ils doivent ajouter une item au Sprint Backlog nécessaire pour atteindre le Sprint Goal. Ils/Elles peuvent se confronter à des obstacles; des problèmes qui bloquent l'atteinte du Sprint Goal et qui sont hors de leur capacité de résolution.



Le Daily Scrum devrait prendre maximum 15 minutes. C'est un moment court et une opportunité minimale pour coordonner les efforts des 24 prochaines heures. Si plus de coordination est utile, les Développeurs peuvent le faire au cours de la journée. Le cadre de travail Scrum ne décrit pas précisément comment réaliser concrètement le Daily Scrum, et l'équipe de Développement est encouragée à trouver le meilleur moyen de faire un Daily qui fonctionne pour eux.

La Revue de Sprint

La Revue de Sprint (ou Sprint Review) arrive à la fin du Sprint, avant la retrospective. Son objet est d'inspecter le travail qui a été achevé à ce jour et de décider quelles sont les prochaines étapes qui donneront du sens en se basant sur ce qui a été appris jusqu'à là. La Sprint Review est le moment du Sprint où, au plus tard, les personnes qui construisent le produit et les personnes qui ont un intérêt pour lui inspectent collectivement ce qui est sorti du Sprint. Avec les conditions actuelles du marché, les changements de l'organisation, de budget et de calendrier, ils/elles décident ensemble des prochaines étapes.

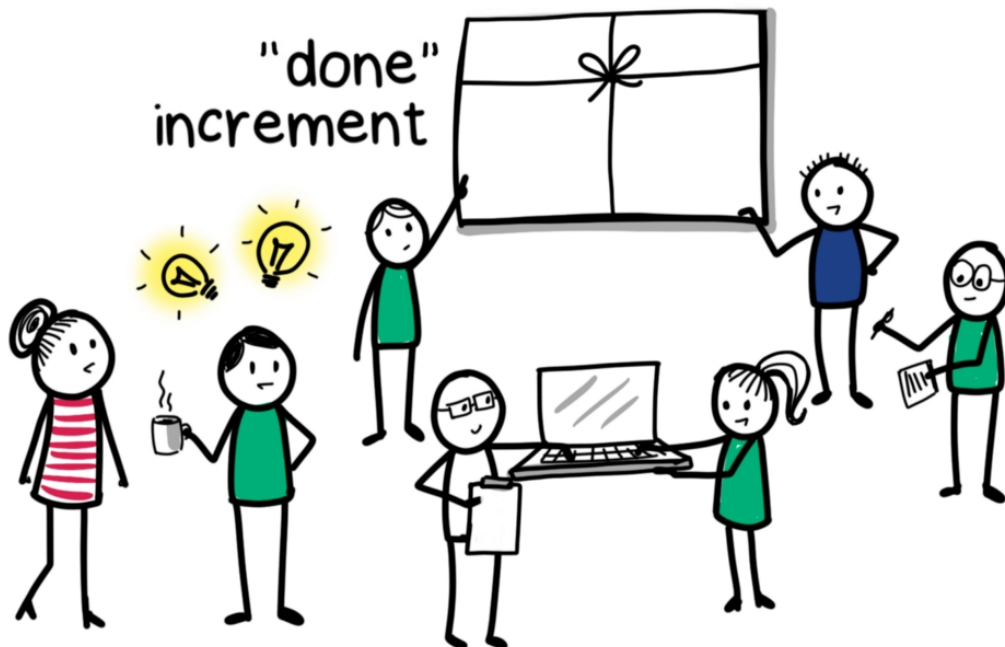
La Revue de Sprint ne devrait pas prendre plus de 4 heures pour un sprint de un mois. Pour un Sprint plus court, la Revue de Sprint tendra à être plus courte. Les éléments en sortie de la Revue de Sprint sont en général des ajustements du Product Backlog basés sur ce que l'équipe a appris. Cela peut inclure des nouvelles idées qui émergent au cours de la Revue de Sprint, des bogues découverts, des changements apportés à des items déjà dans le Product Backlog ou la repriorisation du Product Backlog lui-même. Dans un sens, la Revue de Sprint devrait répondre à la question suivante : "Basé sur ce que vous avez appris du Sprint, quelles sont les prochaines étapes ?". C'est ce qui contribuera à fournir des éléments de valeur en entrée de Sprint Planning et des objectifs de Sprint potentiels pour les Sprints à venir.

En Revue de Sprint, faire une présentation unidirectionnelle de ce qui a été réalisé par l'équipe Développement ne constitue pas une 'inspection'. En réalité, inspecter un incrément signifie le tester ensemble et donner du feedback. On préfère décrire la Revue de Sprint comme une 'collecte de feedbacks' plutôt que comme une 'démonstration'. La Revue de Sprint ne doit pas être le premier moment où le Product Owner voit le travail de des Développeurs terminé. À l'inverse, la Revue de Sprint est un moment important et récurrent où le Product Owner invite ses parties prenantes à inspecter le produit avec lui.

"En Revue de Sprint, faire une présentation unidirectionnelle de ce qui a été réalisé par l'équipe Développement ne constitue pas une 'inspection'."



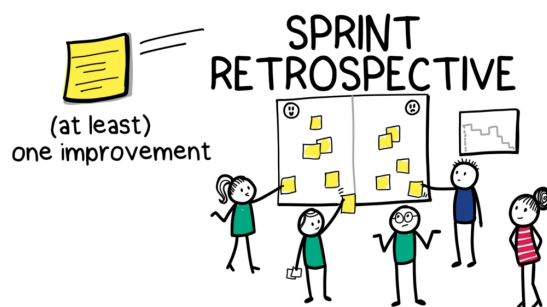
SPRINT REVIEW



Created by Thea Schukken for the Zombie Scrum Survival Guide
by Christiaan Verwijs, Johannes Schartau & Barry Overeem

La Rétrospective de Sprint

La Rétrospective se déroule à la fin du Sprint, en général, juste après la Revue de Sprint. Tous les membres de l'équipe Scrum y participent. L'objectif est d'inspecter comment l'équipe Scrum travaille ensemble à atteindre le Sprint Goal, et d'identifier des étapes concrètes qui peuvent être prises dans le prochain Sprint pour accroître l'efficacité et la qualité. Là où la Revue de Sprint est concentrée sur l'inspection du produit et le contenu du travail réalisé, la Rétrospective vise à inspecter le processus de réalisation de ce travail.



Created by Thea Schukken for the Zombie Scrum Survival Guide
by Christiaan Verwijs, Johannes Schartau & Barry Overeem

Pour des Sprints d'un mois, une Rétrospective devrait prendre au maximum 3 heures. Bien sûr pour un sprint plus court, la timebox le sera aussi. En se basant sur les apprentissages de l'équipe Scrum, en inspectant sa collaboration, ils/elles pourront décider des ajustements pour travailler de manière plus efficace. Cela peut impliquer une mise à jour de la Definition of Done, rechercher des nouveaux outils ou technologies, provoquer un changement dans l'organisation du travail ou une composition différente de l'équipe. A minima, une amélioration devrait intégrer le Sprint Backlog pour le prochain Sprint.



L'affinage du Backlog Produit

Qu'en est-il de l'Affinage du Product Backlog ? Le Scrum Guide le mentionne comme quelque chose qui doit arriver à un moment. Cela n'en fait-il pas un événement ? Alors oui, bien que ce soit clairement un élément essentiel du cadre de travail Scrum, cela n'en fait pas un événement comme les autres. C'est plutôt une activité au long court. Et même si l'on semble là jouer sur les mots, cela dénote d'une importante différence. En premier lieu, il convient de comprendre les différentes formes d'affinage du Product Backlog existantes : – **Détail** : Clarifier les éléments du Product Backlog qui ne sont pas suffisamment détaillés pour être commencés. Il est préférable de faire cela directement avec les personnes pour qui on construit les items (les parties prenantes); – **Découpage** : Décomposer les éléments du Product Backlog qui sont trop gros pour être terminés en un seul Sprint (ce qui veut en général également dire qu'ils ne sont pas clairs); – **Émergence** : Ajouter ou retirer des items au Product Backlog comme les nouvelles idées émergentes; – **Estimation** : Estimer l'effort nécessaire pour implémenter des items particuliers. Cela n'a pas à être aussi 'formel' que l'attribution de story points (une pratique optionnelle de Scrum), de tailles de T-shirt, ou tout autre technique d'estimation que vous utilisez. Un sentiment instinctif est aussi bon ("Ouais, nous savons assez bien ce qu'il y a à faire et cela rentre dans un sprint"); – **Priorisation** : Re-prioriser le travail dans le Product Backlog en fonction des besoins pour rendre les Sprints à venir aussi fluides et valorisables que possible; *En un mot rendre le Product Backlog, plus DEEP – plus étendu et donc compréhensible.*

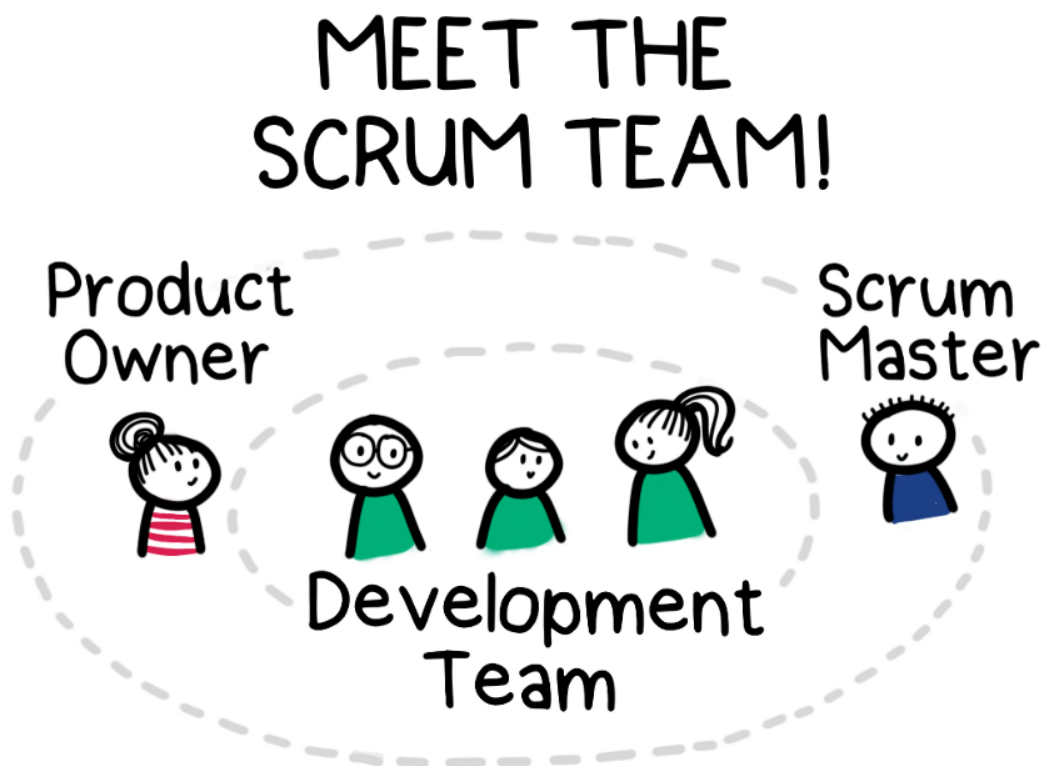
Dans un sens, les items du Product Backlog sont un pense-bête pour les "conversations que vous allez avoir besoin d'avoir dans le futur". L'affinage est un processus continu de ces conversations. Parfois, cela signifie discuter avec les parties prenantes à propos d'une item qui devrait être terminée dans le prochain Sprint, tandis qu'à d'autres moments, cela peut être un moyen de clarifier un élément sur lequel l'équipe travaille déjà. Certaines de ces conversations ont lieu avec toute l'équipe de Développement, d'autres en groupe plus restreint. Certains affinages peuvent parfois se faire individuellement.

Donc, plutôt que de comprendre l'affinage du Product Backlog comme une réunion formelle qui arrive une fois par Sprint, à laquelle chaque membre de l'équipe assiste, cela doit être vu comme une série d'activités au long court permettant d'affiner le travail dans le Product Backlog pour les Sprints à venir. Ces activités au long court se produisent sous différentes formes avec différentes configurations de participants. Et c'est entièrement à l'équipe Scrum de déterminer le meilleur moyen de faire cela.



Les Responsabilités de Scrum

La collaboration entre des professionnels qualifiés est essentielle pour résoudre des problèmes complexes. Les solutions créatives et originales sont plus faciles à découvrir lorsque vous réunissez les points de vue de professionnels qualifiés. Afin de faciliter cette collaboration et de réduire la complexité de la communication et des prises de décisions, le cadre Scrum limite délibérément les responsabilités (ou "roles") à trois. Plutôt que des rôles au sens hiérarchique, où quelqu'un a une autorité sur les autres, chacun représente une perspective différente qui nécessite d'être prise en compte un minimum pour travailler empiriquement. Ensemble, les trois sont suffisants pour travailler empiriquement dans n'importe quel environnement. Aucune autre responsabilité ou rôle n'est nécessaire (et honnêtement, ils/elles risquent de gêner).



Created by Thea Schukken for the Zombie Scrum Survival Guide
by Christiaan Verwijns, Johannes Schartau & Barry Overeem

Le Product Owner

Le Product Owner (ou Responsable Produit) représente la vision de ce qui a de la valeur (et ce qui n'en a pas) en ce qui concerne les ambitions pour le produit. Alors que la Scrum team dépense son temps et son argent à travailler sur le produit, le Product Owner est là pour être sûr que cet investissement rapportera la valeur attendue par les parties prenantes. Avoir une étroite collaboration avec les personnes qui ont un intérêt pour le produit, comme avec l'équipe de Développement, est important pour définir ce qui a de la valeur et ce qui n'en a pas.

"Le Product Owner représente la vision de ce qui a de la valeur (et ce qui n'en a pas) en ce qui concerne les ambitions pour le produit."

Un produit a un Product Owner et un Product Backlog avec un Product Goal. Afin de conserver le rythme auquel les décisions sont prises à un niveau élevé, et de s'y adapter rapidement, les Product Owners doivent avoir un mandat complet sur le produit. Ils/elles ont le dernier mot à propos des ambitions pour le produit, sur ce qui va dans le Product Backlog et ce qui n'y va pas, et comment dépenser le budget (voire même comment le fixer).

Le Product Owner s'assure qu'il y a un Product Goal, qu'il existe un Product Backlog, qu'il est priorisé et qu'il est partagé tant avec la Scrum team qu'avec les parties prenantes. Cela ne signifie pas qu'il est la seule personne de l'équipe Scrum à faire cela. Cependant, afin de maximiser la valeur du travail terminé par les Développeurs dans chaque Sprint, cela a donc du sens pour le Product Owner de travailler activement avec les développeurs à écrire les items, les affiner et les prioriser. Tout est question de collaboration après tout.

Les Développeur.se.s

Les développeurs sont tous les membres de la Scrum Team qui contribuent activement à produire un Increment qui complète le Sprint Goal. Le type de travail qu'ils font ou leur nom de poste officiel n'a pas d'importance, pour le cadre Scrum ils sont tous "développeurs". Ensemble, ils construisent une perspective commune sur la façon de faire le travail nécessaire pour réaliser les ambitions d'un produit et maintenir sa qualité élevée.

Parce que le développement d'un produit est un travail complexe, même le futur proche d'un unique Sprint est difficile à prédire. Il est fort probable que des questions, des défis, et des problèmes puissent surgir et entraveront la capacité de l'équipe à délivrer ce nouvel Incrément. Il est également probable que de nouvelles idées émergent durant le Sprint, à propos de ce qui pourrait être inclus ou sorti de l'Incrément. Cette nature imprévisible, même d'un seul Sprint, a 3 exigences importantes sur la façon dont les Développeurs travaillent et s'auto-organisent.

"Les Développeurs construisent une perspective commune sur la façon de faire le travail nécessaire pour réaliser les ambitions d'un produit et maintenir sa qualité élevée."

La première exigence est qu'ils/elles travaillent dur pour minimiser les dépendances avec d'autres personnes, départements, et compétences qui ne font pas partie de l'équipe mais qui sont régulièrement nécessaires pour délivrer un incrément fini. Chaque dépendance est quelque chose sur laquelle ils/elles ont peu de contrôle, et qui peut les ralentir ou les bloquer totalement dans leur capacité à délivrer un incrément terminé à chaque sprint. Cela a impact négatif sur leur capacité à travailler de manière empirique. Les dépendances peuvent être minimisées de différentes manières, de l'automatisation (par ex : pour le déploiement et les tests) à la formation à de nouvelles compétences en passant par l'intégration des personnes avec les compétences nécessaires à l'équipe, il existe de nombreuses solutions.



La deuxième exigence est que les Développeurs doivent réduire les goulots d'étranglement en redistribuant les compétences dans leur équipe. Avoir un seul testeur dans l'équipe Scrum est mieux que ne pas en avoir. Mais si le testeur est embourbé dans son travail, l'ensemble de l'équipe va s'en trouver ralenti. Il en est de même pour n'importe quelle autre compétence, comme le développement back-end, le design et l'analyse. L'équipe travaille dur pour s'assurer qu'un ou plusieurs de ses membres soient en capacité de faire un travail particulier. Le but ici n'est pas d'avoir tous les membres de l'équipe capable de faire n'importe quel travail – ce serait évidemment irréaliste et irrespectueux des compétences que certaines personnes auraient pu mettre des années à acquérir – mais d'être sûr de chacun et chacune peut venir en aide aux autres ou prendre le relais si nécessaire. Le cadre de travail Scrum répond à cela dans la phrase suivante 'Les équipes de Développement sont transverses et pluridisciplinaires'.

La troisième exigence est que les Développeurs doivent fréquemment s'appuyer sur leur intelligence et leur créativité pour surmonter de nombreuses choses inattendues qui pourraient arriver durant le Sprint. C'est pourquoi l'équipe Scrum devrait être auto-organisée dans la mesure où elle devrait prendre les décisions concernant la façon dont est fait le travail et comment améliorer cette façon de faire. En auto-organisation, il n'y a pas de 'chef' qui prend les décisions pour l'équipe. Toutes les personnes qui font le travail – c'est-à-dire toute les Développeurs – travaillent ensemble à prendre les décisions dans les limites du cadre de travail Scrum.

Le Scrum Master

Le Scrum Master est garant des décisions autour de la maîtrise empirique des processus et de la qualité par laquelle la transparence, l'inspection et l'adaptation s'organisent autour de l'équipe Scrum. Le Scrum Master est là pour faire que les éléments du cadre Scrum s'intègrent dans la vie de l'équipe et plus largement dans toute l'entreprise. Pour cela, les Scrum Masters adoptent un certain nombre de postures, qui dépendent de la situation qu'ils rencontrent face à eux :

- **Enseignant** : ils/elles enseignent et expliquent l'objet du cadre Scrum comme un moyen de travailler de façon empirique. Ils/Elles travaillent dur à s'assurer que chacun comprenne comment les artefacts, les cérémonies, les rôles et les principes favorisent l'empirisme et l'agilité;
- **Facilitateur** : ils/elles explicitent les piliers de Scrum afin que les opportunités de transparence, d'inspection et d'adaptation soient maximisées. Un exemple de ceci est la facilitation des cérémonies Scrum demandées par l'équipe ou qui lui sont nécessaires. Un autre exemple est qu'ils/elles peuvent aider le Product Owner à trouver et utiliser des techniques pour organiser son Product Backlog ou mieux collaborer avec les parties prenantes;
- **Leveur d'obstacles** : ils/elles (aident à supprimer ou) suppriment les problèmes qui bloquent l'équipe Scrum pour atteindre leur Sprint Goal. Le Scrum Master aide l'équipe Scrum à accroître sa capacité à résoudre les problèmes par elle-même. C'est quelque chose que l'équipe a à apprendre, et que le Scrum Master les aide à faire. Ce qui peut être considéré comme un obstacle lors du premier Sprint, peut devenir un problème que l'équipe va facilement résoudre elle-même durant le Sprint suivant.
- **Agent du Changement** : ils/elles aident à supprimer les obstacles dans l'environnement de l'équipe Scrum qui entravent l'empirisme et l'agilité. Par exemple, certaines organisations séparent 'test' et 'déploiement' dans différentes équipes ou départements. Ou encore, les pratiques RH récompensent les individus plutôt que les équipes. La suppression de ce type de contraintes n'est pas quelque chose que le Scrum Master peut gérer seul, il travaillera de concert avec les autres Scrum Masters, les Product Owners et autres pour rendre cela possible;
- **Coach & Mentor** : ils/elles coachent l'équipe Scrum en posant des questions ouvertes puissantes. Ils/elles sont les mentors d'autres Scrum Masters et aident les membres de l'équipe Scrum à trouver les mentors qui ont l'expérience et les compétences pour les aider;

Le Scrum Master est un vrai leader. Plutôt que d'attirer l'attention sur eux/elles-mêmes, ils/elles aident les autres à devenir aussi efficaces que possible. Ils/Elles ne sont ni les managers ni les chefs d'équipe en leur disant quoi faire ou comment le faire. Le seul domaine dans lequel le Scrum Master devrait avoir une posture forte concerne les moments où des décisions sont prises qui ont un impact négatif sur le processus empirique ou sur la mesure dans laquelle les membres de l'équipe se sentent en sécurité pour prendre des risques interpersonnels.



"Le Scrum Master est garant des décisions autour de la maîtrise empirique des processus et de la qualité par laquelle la transparence, l'inspection et l'adaptation s'organisent autour de l'équipe Scrum."



Ensemble, l'équipe Scrum

Ensemble, ces trois responsabilités constituent la Scrum Team (ou équipe Scrum). Ils/Elles ont toute autorité pour prendre les décisions pertinentes pour leur produit. Le Product Owner décide comment utiliser le budget disponible pour maximiser la valeur pour les parties prenantes. Les Développeurs décident comment construire ce produit et a les compétences requises pour le faire. Et le Scrum Master garantit que tout cela est fait de façon à maximiser l'empirisme. Aucun autre rôle n'est nécessaire.

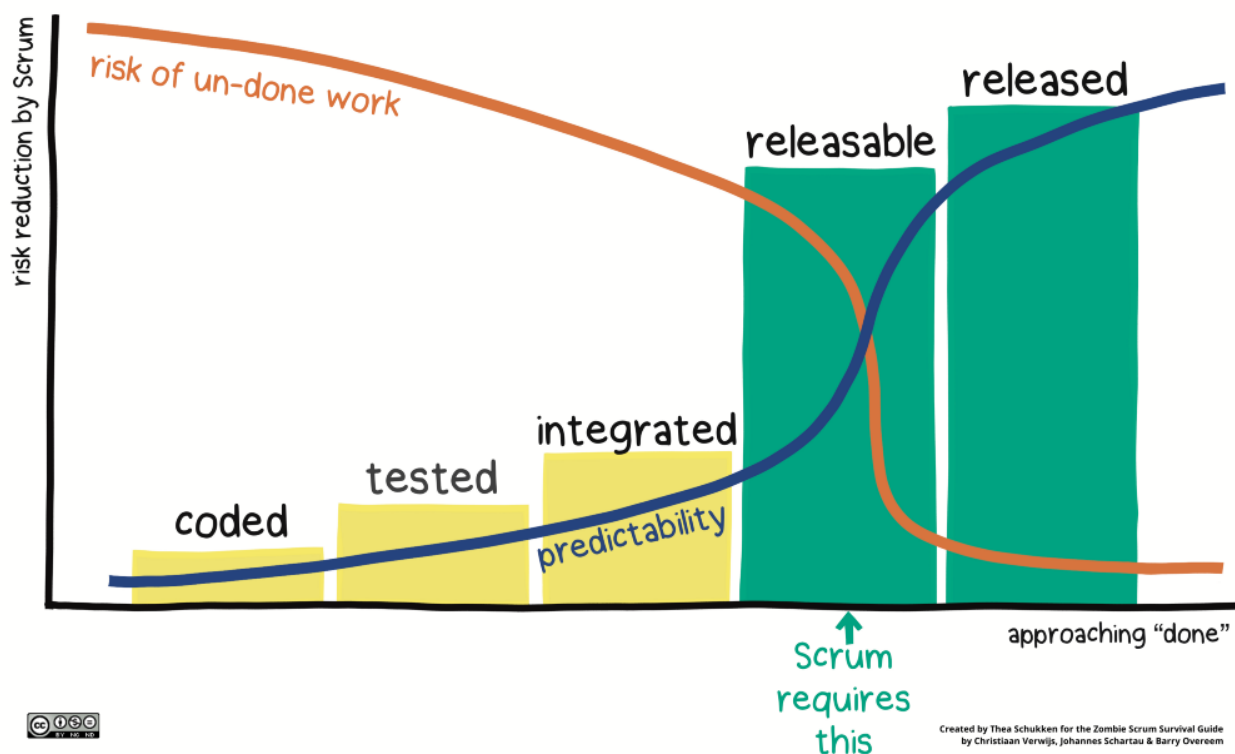
Le cadre Scrum est conçu d'une façon magnifiquement résumé par la citation suivante de [Gunther Verheyen](#); : "Face à la complexité, la simplicité sera notre chemin". Il pourrait être tentant d'ajouter des rôles, des règles, des pratiques et des structures pour le cadre Scrum car vous croyez votre organisation différente. Et tandis que certaines situations pourraient justifier cela, maintenir le focus à garder les choses aussi simple que possible est vraiment la meilleure façon d'avancer des environnements complexes. Cela s'applique aussi à la mise à l'échelle. Ajouter des équipes c'est ajouter de la complexité. Au lieu de cela, explorez des moyens de faire plus avec le minimum de personnes.



Deux Principes Directeurs

Quand les gens décrivent Scrum, ils/elles le font souvent en expliquant les cérémonies ou les rôles de Scrum. Bien que ces éléments soient certainement importants, leur efficacité est avant tout déterminée par la compréhension par les équipes des deux principes directeurs de Scrum.

Délivrer un Incrément "Fini" à Chaque Sprint



Si vous deviez résumer le but du cadre Scrum en une seule phrase, ce serait de travailler de manière empirique à délivrer au moins un Incrément "Fini" à chaque Sprint. C'est aussi pourquoi il est si important que l'équipe Scrum passe du temps à clarifier ce qui est nécessaire à la création d'un Incrément qui est "Fini". Quel travail est requis pour faire cela ? Quels contrôles et tests seront nécessaires pour se conformer aux directives qualité internes ? Qui doit être impliqué là dedans ? Cette compréhension partagée est appelée la "Definition of Done", et cela prend habituellement la forme d'une liste de points de contrôle (checklist).

Le travail sur le Product Backlog a tendance à être rempli d'hypothèses jusqu'à qu'il soit réalisé et mis entre les mains des utilisateurs. Par exemple, "Allons-nous implémenter cette item pour améliorer l'expérience utilisateur ?", "Est-ce que nos utilisateurs comprennent comment le produit fonctionne ?", "Est-ce que les performances sont suffisantes ?". D'autres hypothèses concerneront le travail à effectuer pour terminer d'implémenter l'item, comme "Dans quelle mesure est-il facile de tester et déployer cette item ?", "Quels problèmes pourraient survenir en travaillant sur cette item ?", "Comment bien comprendre ce sur quoi nous travaillons ?". Toutes ces hypothèses représentent un risque. Le risque de mal comprendre ce qu'un utilisateur a demandé et que vous deviez retravailler dessus. Le risque que vous rencontriez des limitations techniques quand vous commencez à coder. Le risque que la fonctionnalité ne s'exécute pas aussi bien qu'attendu. Le risque que l'écriture des tests automatisés s'avère plus difficile que prévu. Ou le risque de dépenser de l'argent et du temps sur une fonctionnalité arrêtée avant même d'être utilisée. Des travaux plus nombreux et inattendus voilà ce que tous ces risques partagent. Et ce genre de travail "reporté" a tendance à être invisible jusqu'au moment où il surgit là où on l'attend le moins, perturbant alors ce sur quoi vous êtes en train de travailler dans ce Sprint. C'est ce qu'on appelle le travail "non-fini".

Le meilleur moyen de prévenir le risque de travail "non-fini" est de s'assurer que chaque Sprint aboutit – au moins une fois – en un Incrément Fini qui peut potentiellement être livré. Cela signifie que l'Incrément a été intégralement testé et qu'il fonctionne. Les textes et visuels sont dans leur état définitif. La documentation et les packages de déploiement sont à jour. Et les performances et la sécurité sont conformes aux standards de l'organisation. A partir de là, l'incrément peut être livré aux utilisateurs en appuyant sur un seul bouton. Après cela, vous saurez qu'il y a peu ou pas de travail potentiel pouvant gâcher la concentration de l'équipe et la prévisibilité des futurs Sprints. Plus important encore, livrer l'incrément permet d'exercer un contrôle empirique des processus, créant un retour d'expériences, pour voir si les modifications du produit se rapproches de vos ambitions, plutôt que de supposer qu'il le fera.

À mesure que la capacité de l'équipe Scrum à fournir des incréments terminés s'accroît, le risque de travail non effectué diminue. Inspiré du travail de Gunther Verheyen. Créer un Incrément "Fin" à chaque Sprint est certainement un défi. Mais à dessein. Car mettre autant de pression sur le système créant le produit – faire tout le travail nécessaire pour y parvenir – montrera clairement où des améliorations sont nécessaires. Où les compétences, les outils, et les technologies sont manquantes. Où la bureaucratie fait obstacle. Où les obstacles ont besoin d'être ôtés afin de réellement travailler empiriquement et de réduire le risque de travail complexe. En gardant un focus maximum sur la livraison d'un Incrément Fini à chaque Sprint, le système va commencer à souffrir aux bons endroits et créer des opportunités pour l'inspection et l'adaptation.

"En gardant un focus maximum sur la livraison d'au moins un Incrément Fini à chaque Sprint, le système va commencer à souffrir aux bons endroits et créer des opportunités pour l'inspection et l'adaptation."

Avoir un But Commun pour Créer la Cohésion Pendant le Sprint

Le Scrum Guide fait référence 40 fois au mot "objectif" (ou Goal) – plus que n'importe quel autre élément du framework. Maintenant, on ne veut pas utiliser ce type de exégèse pour prendre des décisions sur la façon de travailler avec Scrum. Mais cela doit clairement vous indiquer l'importance des objectifs partagés.

La raison pour laquelle le Product Goal et le Sprint Goal sont si importants se réfère à la raison pour laquelle a été conçu Scrum : naviguer efficacement au sein de problèmes complexes et en réduire leur risque. Les Objectif nous aident à naviguer dans cette complexité de trois manières complémentaires.



Premièrement, ils guident l'équipe scrum quand elle doit prendre des décisions à propos du temps à passer dessus. Certains travaux du Sprint Backlog peuvent être plus importants que d'autres pour atteindre le Sprint Goal. Deuxièmement, ils permettent à l'équipe de garder le focus sur ce qui est important. Quand les choses deviennent difficiles, comme cela arrive souvent, l'équipe peut décider d'abandonner certains travaux et redonner la priorité à d'autres. Ou, quand le temps le permet, ils peuvent ajouter du travail, qu'ils ont découvert comme nécessaire à l'atteinte du Sprint Goal, en cours de Sprint. Enfin, troisièmement, le Sprint Goal encourage la collaboration en donnant à l'équipe Scrum un objectif clair et partagé pour s'auto-organiser plutôt que de travailler sur des initiatives individuelles. La collaboration rend aussi possible un type de pensées hors du commun et un esprit d'équipe nécessaire à résoudre des problèmes complexes.

Le Product Goal et le Sprint Goal donnent aussi des couleurs et un but aux différents événements Scrum, les élevant au-delà de simples réunions de travail. Le Sprint-planning vise à déterminer le but du prochain Sprint et à sélectionner le travail nécessaire pour l'atteindre. Le Daily Scrum se concentre autour de la façon dont les Développeurs vont passer les 24 prochaines heures à travailler vers ce but, et quels obstacles bloquent sa capacité à faire cela. La Revue de Sprint sert à vérifier l'atteinte ou non du Sprint Goal avec les parties prenantes, comment il répond au Product Goal, et à travailler avec eux à identifier les prochains objectifs à retenir. La Rétrospective tend à découvrir les moyens de travailler ensemble plus efficacement à atteindre les objectifs de Sprint.

Mais le Product Goal et le Sprint Goal mettent aussi l'accent sur les responsabilités du cadre de travail Scrum. Il donne à aux Développeurs des orientations et un sens sur ce qui doit être auto-organisé dans l'équipe. Il permet au Product Owner de rester concentré sur ses ambitions pour le produit et comment les traduire en objectifs pour les différents Sprints, plutôt que de s'inquiéter du détail de comment implémenter cela. Enfin, Le Scrum Master peut augmenter l'efficacité du contrôle de processus empirique en s'appuyant sur le Product Goal et le Sprint Goal comme le canari des mines de charbon indiquait aux mineurs un danger imminent. Quand l'équipe lutte à créer un objectif, c'est une bonne raison pour le Scrum Master pour se glisser dans la peau de Sherlock et d'en explorer les causes. Les équipes peuvent être trop grosses ou trop petites, ou manquer de certaines compétences ou personnes. Le Product Owner peut avoir un mandat insuffisant ou manque de vision. Ou encore l'affinage n'a pas eu lieu.

Cette explication devrait rendre clair comme du cristal pourquoi il ne devrait y avoir qu'un seul Objectif produit par produit et qu'un seul objectif de Sprint par sprint. Multiplier les objectifs ne fera que diluer la concentration, nuire à l'engagement et limiter la transparence. Bien sûr, un nouvel objectif peut être formulé dès que le précédent est terminé ou abandonné.

La petite histoire est que dans un travail complexe, des objectifs partagés comme le Product goal ou le Sprint Goal est le phare qui aide à atteindre le port à travers un brouillard dense. Sans lui, vous risquez de vous perdre ou de courir à terre.



Les Cinq Valeurs Fondamentales qui Permettent l'Empirisme



Jusqu'ici, nous avons couvert tous les mécanismes et principes du cadre Scrum. Bien qu'ils fassent des merveilles pour permettre le contrôle de processus empirique, ils ne feront pas grande différence sans valeurs pour soutenir cela. A quel point l'inspection de la Revue de Sprint peut-elle être honnête, si l'équipe Scrum est effrayée à l'idée d'être ouverte à propos des challenges techniques qui se présentent, et qu'elle préfère dire que "tout va bien" ? Quelle efficacité peut avoir la Scrum Team, si leur Product Owner a peur de refuser ou rejeter des items dans le Product Backlog qui ne correspondent pas aux objectifs ? Combien de valeur peut être générée par l'équipe de Développement, si elle reste captivée par des technologies plus récentes et performantes ? Comment en changeant la composition de l'équipe sans leur consentement ou à leur insu sera affecté leur engagement et leur capacité à travailler en équipe ?

C'est difficile de travailler de manière empirique, spécialement dans les organisations qui n'y sont pas habituées. Ainsi, afin de donner à l'équipe et leurs parties prenantes quelque chose qui dirigera leurs décisions, le cadre Scrum souligne justement cinq valeurs directrices. Pour chaque décision prise, l'équipe peut se questionner sur comment leurs choix impactent ces cinq valeurs :

- **Focalisation** : Restez focalisés sur le Sprint Goal et les objectifs de l'équipe Scrum. Créez un espace où les personnes peuvent garder et maintenir leur concentration.
- **Ouverture** : Soyez ouverts à propos des choses qui arrivent. Qu'est-ce qui se passe bien ? Ou mal ? Où sont les défis et les opportunités ?
- **Respect** : Respectez les compétences, les expertises et l'intelligence de chaque membre de la Scrum Team. Croyez en leur capacité à s'auto-organiser face à des problèmes complexes. Enfin respectez l'incertitude inhérente aux travaux complexes.
- **Courage** : Soyez courageux pour faire les bons choix. Dites "Non" aux choses qui entravent les processus d'apprentissage (processus empirique). Faites preuve de courage en travaillant ensemble à des défis compliqués. Demandez et donnez des réactions (feedbacks) à propos des choses dont vous n'êtes pas sûrs. Posez des questions et admettez ce que vous ne savez pas ou dont vous n'êtes pas sûrs.
- **Engagement** : Créez un environnement où les personnes peuvent personnellement s'investir à travailler ensemble en équipe vers un objectif commun (le Sprint Goal).

Apprendre à intégrer ces valeurs dans nos comportements quotidiens est le parcours de toute une vie. Plus l'équipe sera compétente, plus la transparence, l'inspection et l'adaptation fréquentes seront effectives. La bonne nouvelle est que le cadre Scrum fournit un excellent ensemble de limites pour apprendre et grandir en adoptant ces valeurs.



Ensemble, un Cadre de Travail Simple

Scrum est un cadre de travail simple pour faire face à des problèmes complexes et adaptatifs. Il prescrit uniquement ce que les équipes doivent faire pour travailler empiriquement, mais pas comment elles doivent le faire. Parce que chaque équipe, chaque produit et chaque organisation est différente, les équipes doivent trouver leur propre façon de travailler ensemble. Quand elles le font, elles peuvent réduire le risque de travail complexe, commencer à délivrer de la valeur à leurs parties prenantes plus tôt, et devenir plus réactives. Cette aventure sera plus facile pour certaines équipes Scrum que pour d'autres, mais si vous persistez le changement arrivera.



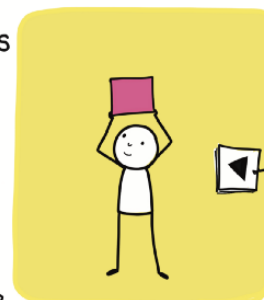
(1) In complex work more is unknown than known.



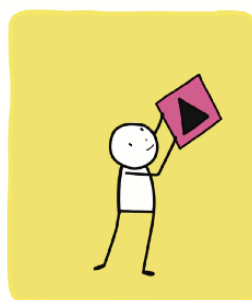
(2) The unknown is discovered by releasing “done” increments early and often.



(3) With these increments we validate assumptions.



(4) We learn what is needed and avoid the risk of spending time and money on the wrong things.



(5) As a result, we can deliver more value to our stakeholders sooner.



Created by Thea Schukken for the Zombie Scrum Survival Guide
by Christiaan Verwijs, Johannes Scharlau & Barry Overeem

zombiescrum.org

Unleashing Teams All Over The World

We are The Liberators – Barry Overeem and Christiaan Verwijs. Our mission is to create data-driven products to unleash the superpowers of teams all over the world. We do this together with a growing community of patrons.

Awesome content for awesome teams

We unleash teams with our [blogposts](#), our [podcast](#), our [newsletter](#), our [videos](#), and our frequent [meetups](#). While we offer most of this for free, we also have plenty of premium content in [our webshop](#) for you to explore.

Supported by the community

We are super proud that The Liberators is almost entirely funded by the community. If you appreciate our work too, you can already support us for 12 dollars/year by becoming a [patron](#). In return, you gain free access to premium content, we share our work-in-progress and involve you in creating more awesome content.



Thank you for respecting our work!

We work hard to create high-quality content that puts you in a position to unleash your team. We're sure you appreciate that a lot of our time and money goes into this. At the same time, content like this is our main source of income. It's what pays our bills :)

If you purchased this content, we ask only that you treat our work with respect and don't share it with people outside your team. If you stumbled on it elsewhere without paying for it, and it offers you value, would you consider [supporting us too](#)? You can also check [out our other offerings](#).