

Du Meilleur Scrum



Des idées et conseils sur la manière de bien mettre Scrum en oeuvre

Écrit par Coach et Formateur Certifié Scrum Peter Hundermark



traduit par Stéphane Wojewoda (#stephanewww), Matthieu NGuyen et Yann Le Moal



Sommaire

Pourquoi ce guide?	4
Qu'est-ce que Scrum ?	6
Origine.....	6
Quel est le problème?.....	6
Comment Scrum résout cela?	7
L'essence de Scrum	8
Scrum et son application	10
Pourquoi Scrum reste silencieux sur les pratiques de conception?..	10
Quel lien entre Scrum et les méthodes traditionnelles ?.....	10
Scrum va-t-il se disséminer dans mon entreprise?	11
2. Comprendre Scrum	13
L'Equipe Scrum (Scrum Team)	13
L'auto-organisation.....	13
Le Propriétaire du Produit (Product Owner)	13
Equipe de développement (Development Team)	15
Scrum Master.....	17
Autres rôles.....	18
Les événements Scrum	21
Le Sprint	21
Planification de Sprint (Sprint Planning)	23
Mêlée Quotidienne (Daily Scrum)	26
Revue de Sprint (Sprint Review)	27
Rétrospective du Sprint (Sprint Retrospective).....	28
Artefacts Scrum	31
Le Carnet de Produit (Product Backlog)	31
Carnet de Sprint (Sprint Backlog)	33
Tableau d'Atterrissage de Sprint (facultatif) (Sprint Burndown Chart) ..	33
Incrément.....	34
Courbe d'Avancement du Produit de livraison (facultatif) (Product or release burnup chart).....	35
3. Adopter Scrum	37

Commencer Scrum	37
#1 Former l'équipe Scrum	38
#2 Définir la vision	39
#3 Construire le Carnet de Produit initial	40
#4 Ranger les éléments du Carnet de Produit par ordre de valeurs	42
#5 Dimensionner les éléments du Carnet de Produit.....	42
#6 Ré-arranger le Carnet de Produit avec d'autres facteurs.....	44
#7 Créer un contenu de l'itération à grosse maille.....	44
#8 Planifier et commencer le premier Sprint	46
4. Colocalisation et l'équipe dans l'espace	47
5. Mesures	51
Objectifs et points d'attention.....	51
Premières métriques	52
6. Déployer Scrum à l'échelle de l'Organisation	55
Voici des dragons!	55
Méthodologies et modèles de changement d'échelle	56
Mon approche de mise à l'échelle	57
7. S'améliorer	60
Shu Ha Ri	60
Un modèle pour l'hyper-productivité	61
Les équipes stables	62
La météo d'hier	62
Mode essaim	62
Pattern (Motif, comme dans design pattern - NdT) d'interruption ..	63
Du Code Propre tous les soirs.....	63
Procédure d'urgence	64
Scrumer le Scrum.....	64
Indicateur de bonheur.....	65
Les équipes qui terminent tôt, accélèrent plus vite	65
Plus sur les obstacles	66
Cartographie des histoires (story mapping)	67
Planification des livraisons avec une courbe d'avancement (burn up chart).....	68

Le concept de vélocité	68
Utilisation d'une enveloppe de vélocité pour la planification des livraisons	68
8. De l'aide	71
La culture et l'éléphant	71
Qu'est-ce que le coaching?	72
Comment puis-je savoir quand j'ai besoin d'aide?	73
Quels sont les aspects économiques du coaching?	74
9. Un peu plus sur l'Agile et le Lean	76
Le Manifeste Agile	76
Développement de produits Agile et Lean	77
Autres méthodes Agile et Lean	78
Remerciements	81
Références	83

Pourquoi ce guide?

Jim York, Formateur (CST) et Coach (CSC) Scrum Certifié nous dit que: **Scrum est simple. Faire du Scrum est difficile.**SM

De nombreuses personnes rencontrées en entreprise me disent qu'elles ne savent pas comment commencer avec Scrum. D'autres ont des équipes qui suivent certaines pratiques Agiles, et sont loin de l'état *d'hyper-productivité* décrit par Jeff Sutherland.

Durant les quatre dernières années écoulées depuis la première édition de ce petit guide, beaucoup de nouveaux et d'excellents livres ont été écrits sur Scrum. Parmi ceux que j'ai lu et que je recommande à mes étudiants on trouve *Succeeding with Agile* [Cohn 2009], *Agile Product Management with Scrum* [Pichler 2010], *Coaching Agile Teams* [Adkins 2010], et *Essential Scrum* [Rubin 2012]. En fait beaucoup ne les ont pas lus et il subsiste donc une attente pour des guides comme celui-ci.

Mon premier objectif avec cette nouvelle édition est de fournir une référence plus étendue et complète sur le monde de l'agile et de Scrum, en conservant l'essentiel. Un second objectif est de mettre à jour les pratiques décrites pour les aligner avec ce que j'enseigne et la manière dont je coache aujourd'hui. L'Agile est un ensemble émergent de motifs et de pratiques qui, par définition, changent avec le temps. Le troisième et dernier objectif est de conserver l'aspect factuel et sans fioritures de ce guide. Pour ceux qui veulent en savoir plus, la section références a été significativement étendue.

Dans la première version j'écrivais : "J'espère que ce livret sera une source d'inspiration pour vous aider à pratiquer Scrum et l'Agile un peu mieux chaque jour. Mais plus encore, j'espère que cela vous encouragera, vous, votre équipe et toute votre organisation à vous éloigner des sentiers battus qui, ne marchent, simplement pas, et d'en trouver de nouveaux qui vous guideront vers une plus grande qualité, des livraisons plus rapides, et par dessus tout, plus de fun". Ce but n'a pas changé.

Une chose reste certaine : vous ne deviendrez jamais meilleur avec Scrum (ou n'importe quelle autre méthode agile) sans pratiquer. Qu'attendez-vous ? Allez-y maintenant !

Peter Hundermark

Troisième édition, Cape Town, February 2014

Qu'est-ce que Scrum ?

Origine

Scrum est un framework permettant de développer des produits complexes. Scrum est issu de travaux en gestion des connaissances, en systèmes complexes adaptatifs et en théorie des processus de contrôles empiriques. Sa genèse reconnue est un article de Nonaka et Takeuchi [Nonaka 1986] "*The New, New Product Development Game*". De manière indirecte, Scrum s'inspire de ce qu'on nomme généralement le Lean.

Scrum est de loin la plus populaire des méthodes Agiles. En 2013, 72% des équipes qui se disent agiles utilisent Scrum seul ou en combinaison avec d'autres méthodes [VersionOne 2013]. Pour plus d'informations à propos de l'Agile et des autres méthodes, reportez vous au chapitre 9 – *Plus sur l'Agile et le Lean*.

Quel est le problème?

- Les versions sont trop longues à sortir
- La phase de stabilisation prend trop de temps
- Il est difficile de faire des changements
- La qualité décroît au fil du temps
- Les marches de la mort détériorent le moral des équipes

Depuis des décennies, les développeurs essayent d'utiliser des méthodes prédictives pour gérer les projets. Les méthodes prédictives sont tout à fait appropriées lorsque les entrants sont bien définis et que la méthode pour les convertir en livrables est standardisée. De telles méthodes ne conviennent pas pour le développement logiciel ou pour toutes autres formes de travaux complexes. L'illustration en est le fort taux d'échecs des projets et l'insatisfaction des clients.

Comment Scrum résout cela?

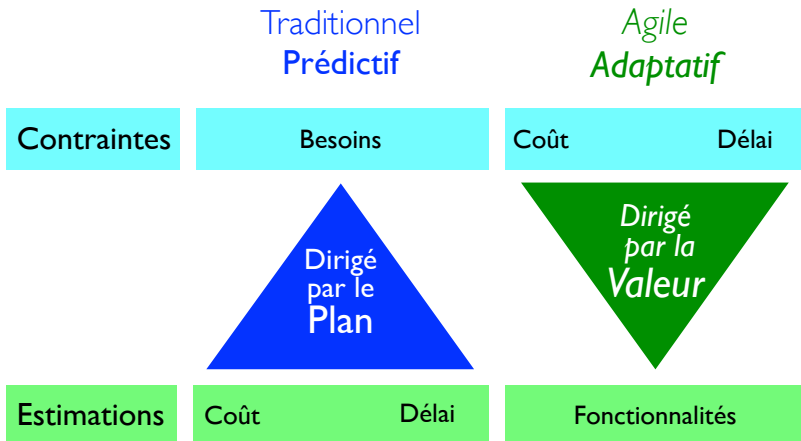
Alistair Cockburn [Cockburn 2008] décrit le développement de logiciels comme "un jeu d'invention et de communication coopératif".

Les méthodes traditionnelles de développement s'appuient sur des documents pour archiver et transmettre des connaissances d'un spécialiste à l'autre. Les feedbacks sont trop longs voire inexistantes. Les décennies de projet sous-performants montrent que ces méthodes de travail sont un échec total.

En Août 2012, le Gartner a publié un document de recherche intitulé *The End of Waterfall as We Know It* qui affirme que les longs projets en cascade sont la manière la plus risquée de construire des logiciels [Gartner 2012].

Scrum fournit un socle pour que les gens travaillent ensemble efficacement et met constamment en lumière tous les problèmes qui sont sur le chemin.

J'ai constaté que les dirigeants d'entreprise comprennent et identifient bien les différences entre la gestion de projet classique et le modèle Agile avec le schéma suivant.



Dans un modèle traditionnel, le chef de projet tente de figer le périmètre projet pour proposer une estimation fiable en temps et coût. Dans la

pratique, la situation rencontre une «triple contrainte» ou un «triangle d'incompatibilité» où la qualité devient une variable non maîtrisée. Dans une vision Agile, le temps et le coût sont les contraintes réelles. Le chef de produit travaille ensuite avec l'équipe de manière itérative et incrémentale pour maximiser la valeur de ce qui est livré. Le plan est régulièrement mis à jour pour correspondre à la réalité. (Comment croire que l'inverse pourrait fonctionner ?)

Je constate que les dirigeants sont ravis de découvrir que le temps et le coût peuvent être définis ou du moins contrôlés en raisonnant par incrément. Naturellement, il y a une certaine anxiété initiale du fait du périmètre variable. Je surmonte cela en leur demandant la permission de réaliser un test avec des contraintes clairement définies.

L'essence de Scrum

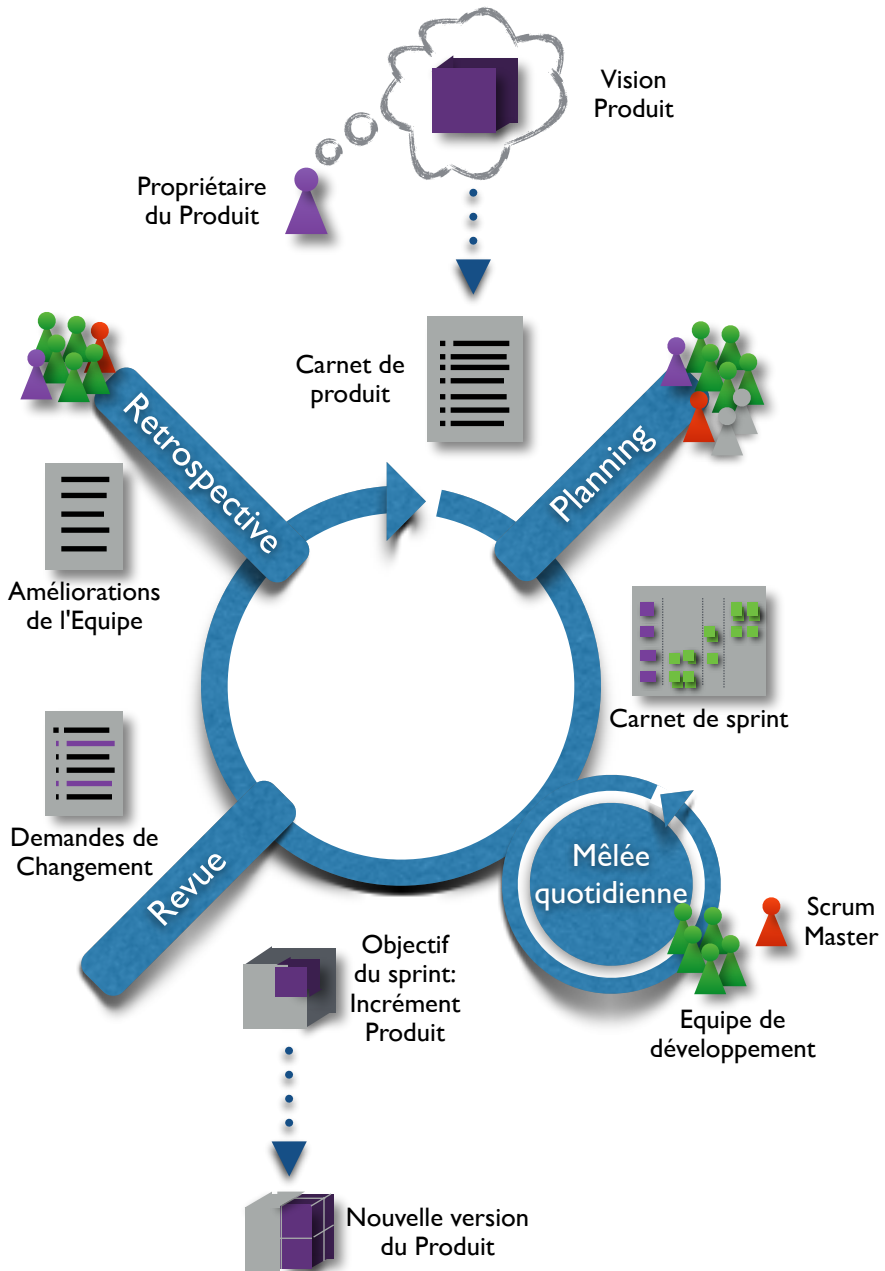
L'essence de Scrum se traduit par:

- L'équipe a des objectifs clairs
- L'équipe s'auto-organise pour travailler
- L'équipe fournit régulièrement les fonctionnalités avec la plus grande valeur ajoutée
- L'équipe reçoit régulièrement les feedback extérieurs
- L'équipe analyse sa façon de travailler afin de s'améliorer
- L'avancement de l'équipe est transparente pour toute l'organisation
- L'équipe et le management communiquent honnêtement sur les progrès et les risques

Cette façon de travailler est basée sur les valeurs d'ouverture, de concentration, d'engagement, de respect et de courage décrites dans le premier livre sur Scrum [Schwaber 2001].

Des descriptions formalisées et actualisées du noyau de Scrum sont disponibles dans de nombreuses langues avec l'Atlas Agile [Jeffries 2013] et le Guide Scrum [Schwaber 2013].

La page suivante modélise les cycles d'événements, les flux d'objets et les rôles des participants dans un cycle Scrum.



Scrum flux

Scrum et son application

Bien que Scrum soit issu du monde du développement logiciels, il est adapté à tous les types de travaux complexes. Il est aujourd'hui utilisé pour gérer le développement logiciel et matériel, l'assistance technique, la publicité et le marketing , les églises et des organisations entières.

Pourquoi Scrum reste silencieux sur les pratiques de conception?

Scrum ne cherche pas à imposer les pratiques de travail aux équipes. Scrum implique que les équipes fassent ce qui nécessaire pour livrer le produit attendu. Il leur donne le cadre pour le faire. Les pratiques de conception et les outils changent et s'améliorent tout le temps, les bonnes équipes les intégrerons pour en tirer constamment avantage.

Si vous développez des logiciels, vous aurez besoin d'identifier et d'appliquer un ensemble de pratiques de conception de logiciels pour compléter le cadre qu'est Scrum. Heureusement Kent Beck et ses collaborateurs ont déjà regroupé un ensemble de pratiques avec l'eXtreme Programming (XP) [Beck 2005].

Si vous travaillez dans un domaine autre que la conception de logiciels, vous aurez besoin de chercher votre propre ensemble de pratiques complémentaires.

Quel lien entre Scrum et les méthodes traditionnelles ?

En bref, il n'y en a pas. L'Agile et Scrum sont basés sur des paradigmes différents. Les fondateurs Jeff Sutherland et Ken Schwaber ont souvent affirmé que le rapprochement des méthodes prédictives (normatives) et empiriques (adaptatives) est futile [Sutherland 2007].

Une question plus pertinente pourrait être : "Comment opérer la transition de mon équipe ou de mon organisation vers Scrum ? Sur quelles forces et réussites puis-je m'appuyer ? Quels sont les défis récurrents et comment puis-je me préparer à les surmonter ?"

Pour une discussion étayée sur l'Agile par rapport aux méthodes traditionnelles, je vous invite à lire l'article de Martin Fowler, *The New Methodology* [Fowler 2005].

Scrum va-t-il se disséminer dans mon entreprise?

On me demande souvent des preuves de la réussite des méthodes Agiles. Habituellement ce que le demandeur recherche est un succès dans une industrie , organisation , une culture, etc. similaire. Parfois, je peux leur donner des exemples que je connais personnellement . Je les renvoie également à des rapports publiés et des études de cas qui montrent que des milliers d'équipes sur tous les continents et dans tous les secteurs réussissent à améliorer leurs conditions de travail.

Je pense que le plus important est de pouvoir les orienter vers les gens que j'ai croisé durant les huit dernières années, et qui ont réalisé la transition d'un mode de fonctionnement emprunt de frustration personnelle vers une joie retrouvée au travail.

Néanmoins le succès de l'Agile est entre vos mains ! L'introduction de l'Agile implique d'énormes et d'inévitables changements au sein de l'organisation. La réussite ou l'échec du déploiement de l'Agile dans votre organisation dépend de la compréhension et de l'acceptation de ces changements.

Apprendre les règles de l'une ou l'autre méthode Agile est la partie facile. Les déterminants de la réussite ou de l'échec est la gestion des vagues successives de changement à tous les niveaux de l'organisation, y compris les impacts sur la culture de l'organisation.

Scrum est un moyen pour l'organisation de révéler le potentiel illimité de ses membres. Scrum (et toute méthode Agile) implique également des coûts associés, qui prennent différentes formes, dont une est l'argent. Vous aurez besoin d'investir dans la compréhension de la proposition valeur que l'Agile représente avant de généraliser les pratiques au-delà des premières expériences .

Enfin, la transition organisationnelle vers l'Agile prendra des années. Peu importe la taille de votre organisation , l'alignement supposé à la culture,

ou tout autre facteur auquel vous pouvez penser pour aplanir le chemin, il faudra de cinq à dix ans pour devenir vraiment bon et institutionnaliser ces nouvelles manières de travailler. Alors préparez-vous pour un long voyage. Et profitez de la balade !

Résumé du Chapitre

- ◆ Scrum est un cadre basé sur un processus empirique
- ◆ Scrum s'appuie sur un cycle d'évènements, un flux de travail et une définition claire des rôles
- ◆ Scrum peut s'appliquer largement
- ◆ La réussite de scrum dépend de la capacité et la volonté de l'organisation à changer

2. Comprendre Scrum

L'Equipe Scrum (Scrum Team)

Il y a seulement trois rôles dans une Equipe Scrum : le Propriétaire du Produit (Product Owner / PO), l'Equipe et le Scrum Master. La composition et l'interconnexion de ces trois rôles sont fondamentales et essentielles à l'efficacité du modèle Scrum. L'Equipe Scrum s'auto-organise pour accomplir le travail.

L'auto-organisation

L'auto-organisation n'est pas une absence d'organisation, au contraire, les Equipes auto-organisées sont très disciplinées. L'auto-organisation se construit dans un cadre de contraintes consenties, d'un laissez-faire (en français dans le texte - NdT). Au coeur de ce cadre les Equipes sont complètement autonomes, et prennent des engagements vis-à-vis des autres parties prenantes. Elles prennent leurs responsabilités quant aux livraisons sur lesquelles elles se sont engagées, dans la limite de leur périmètre. Elles sont incitées à prendre des risques raisonnables et à apprendre par l'échec et l'introspection. Les Equipes auto-organisées présentent un haut niveau de confiance et de motivation intrinsèque.

Les nouvelles Equipes Scrum ont besoin d'encouragements pour explorer leur nouveau périmètre et se l'approprier. Elles ont souvent besoin de dépasser les mauvais réflexes avec lesquels elles ont travaillé et ont été gérées, parfois pendant de nombreuses années.

L'auto-organisation n'est pas une option dans Scrum, c'est un principe de base. Sans cela, il n'y aura pas d'Equipes très performantes. *Caveat emptor!*

Le Propriétaire du Produit (Product Owner)

La principale responsabilité du Propriétaire du Produit (PO) Scrum est d'optimiser le retour sur investissement (ROI) en veillant à ce que les membres de l'Equipe Scrum s'engagent dans la réalisation des fonctionnalités Produit à plus forte valeur ajoutée.

La tâche principale du Propriétaire du Produit est de se concentrer sur l'efficacité, c'est-à-dire la construction du bon Produit pour les clients.

Les responsabilités du rôle "Propriétaire du Produit" sont:

- D'assurer l'existence d'une vision partagée pour le Produit
- De gérer et hiérarchiser le Carnet de Produit
- D'aider les membres de l'Equipe à comprendre ce qu'il faut construire et pourquoi
- De valider à la fin de chaque itération les nouvelles fonctionnalités ajoutés au Produit
- De gérer le planning des versions
- D'informer sur l'avancement et de gérer les attentes des parties prenantes
- De maximiser la valeur du Produit

Il n'y a qu'un seul Propriétaire du Produit dans une Equipe Scrum. Tobias Mayer l'appelle la «**Voix du Quoi**» [Mayer 2009]. Les autres parties prenantes qui s'intéressent au Produit sont... Eh bien... des parties prenantes !

Métaphore: Le Propriétaire du Produit est un directeur général.

- ☑ *Pour paraphraser une célèbre citation d'Highlander: il ne peut y avoir qu'un Propriétaire du Produit dans une Equipe Scrum ! Tous les autres sont des parties prenantes*
- ☑ *Le problème inhérent à ce rôle est le Propriétaire (canard) boiteux du Produit auquel les parties prenantes (surtout le client et le sponsor) n'ont pas donné les moyens de prendre des décisions. En conséquence, l'Equipe ne parvient pas à obtenir des réponses fiables de sa part. Le respect et la confiance en souffrent et la motivation diminue*
- ☑ *Un second challenge classique est le Propriétaire du Produit "porté disparu". Il est introuvable et l'Equipe attend un cap à suivre. C'est ainsi que l'Equipe finit par ralentir ou faire des suppositions*

Equipe de développement (Development Team)

L'Equipe de développement est l'ensemble des personnes responsables des Incréments fonctionnels du Produit potentiellement livrable à l'issu de chaque Sprint.

Le premier travail de l'Equipe de développement est de se concentrer sur l'efficacité, c'est-à-dire sur la construction du bon Produit pour le Propriétaire du Produit et les utilisateurs.

Les responsabilités de l'Equipe de développement sont:

- De travailler avec le Propriétaire du Produit et les autres parties prenantes (souvent les utilisateurs) pour affiner progressivement les éléments du Carnet de Produit afin que chacun soit bien compris et suffisamment petit pour être réalisé en un Sprint
- De s'engager auprès du Propriétaire du Produit sur un ensemble minimum d'éléments à livrer à la fin du Sprint
- De s'auto-organiser de manière à livrer les éléments promis et...d'y arriver!
- De suivre quotidiennement son avancement vers l'objectif du Sprint
- De veiller à ce que la conception du Produit reste robuste et que le code (ou les autres éléments du Produit) reste maintenable du fait d'un refactoring permanent

Tobias Mayer appelle l'Equipe : «**La tribu du comment**» [Mayer 2009].

- L'Equipe se compose de développeurs, testeurs, analystes, architectes, designers, rédacteurs, et même des utilisateurs - toute personne qui contribue à faire le travail*
- Il n'y a pas de hiérarchie imposée au sein de l'Equipe. Les meneurs apparaîtront en fonction des situations à gérer*

- ☑ L'Equipe de développement est pluridisciplinaire, ce qui signifie que l'ensemble de ses membres possède les compétences nécessaires à la livraison de l'Incrément du Produit décidé. Une Equipe pluridisciplinaire ne signifie pas que tout le monde peut effectuer chaque tâche. Néanmoins, les meilleurs membres de l'Equipe peuvent être «en forme de T» par opposition à «en forme de I» [Reinertsen 2009]. Les gens en forme de I ont une seule compétence spécifique pour l'Equipe, alors que les gens en forme de T ajoute à cette compétence d'autres plus générales. De la sorte, ils sont capables d'aider leurs co-équipiers pour parvenir à livrer l'ensemble minimum d'élément promis
- ☑ L'Equipe Scrum comprend trois rôles : un Propriétaire du Produit, un Scrum Master et trois à neuf membres. Avoir une Equipe de moins de trois membres engendre des inconvénients comme le manque de résistance, la difficulté à être pluridisciplinaire et à accroître ses compétences. Une Equipe de plus de neuf membres peinera à travailler comme une équipe soudée et le coût pour maintenir un même niveau d'information est trop élevé. Imaginez l'Equipe comme une famille ou une tribu qui se tient chaud autour d'un feu de camp!
- ☑ Si vous avez plusieurs très petites Equipes chacune dédiée à un Produit ou un projet, envisagez plutôt la création de quelques Equipes Scrum bien construites et consolidez leur travail dans un seul Carnet de Produit
- ☑ Je vous recommande vivement d'éviter la tentation de remplir plusieurs rôles dans une Equipe Scrum ou un seul dans plusieurs Equipes. Vous risquez de compromettre fortement l'efficacité du modèle Scrum. Désolé d'être grossier, mais si vous êtes un novice, qu'est-ce qui vous permet de penser que vous pouvez améliorer le modèle?
- ☑ Lorsque le travail demandé exige plus de membres dans l'Equipe qu'il n'est possible d'en faire travailler efficacement ensemble, il faut recourir à une "mise à l'échelle" (scaling) de Scrum. La Communauté Scrum a développé un certain nombre de modèles de mise à l'échelle très pratiques. Vous seriez bien avisé de consulter l'excellente littérature sur ce sujet ou de faire appel à un expert pour vous éviter certaines peines ! Pour plus d'informations sur le sujet, voir le chapitre 6 "Mettre Scrum à l'échelle dans l'Organisation"

Scrum Master

Le Scrum Master gère tous les aspects de processus pour l'Equipe. Puisque l'Equipe s'auto-organise autour du travail, le Scrum Master influence plutôt qu'il ne dirige. C'est un changement complet de paradigme quant aux objectifs de «diriger» pour la plupart des organisations. C'est à la fois la chose la plus difficile à saisir dans Scrum et son plus puissant catalyseur d'amélioration de la performance.

La tâche principale du Scrum Master est d'aider l'Equipe à *s'améliorer continuellement*, en raccourcissant le temps des boucles d'apprentissage (feedback loops). Cela aide l'Equipe à devenir plus forte en tant qu'équipe et ainsi à travailler mieux et plus vite.

Les responsabilités du rôle de Scrum Master sont:

- D'accompagner l'Equipe vers l'auto-organisation et l'amélioration continue
- D'éliminer les obstacles organisationnels que rencontre l'Equipe
- D'aider le Propriétaire du Produit à comprendre et à accomplir son rôle
- De fluidifier les événements de Scrum
- De diffuser Scrum à un niveau supérieur dans l'organisation

Métaphore : Le Scrum Master est un facilitateur, un entraîneur, un mentor et bulldozer! En fait, il est souvent appelé le coach de l'Equipe, le rapprochement avec le coach sportif s'appliquant bien.

Le Scrum Master doit aider à débloquer les trois facteurs de motivation intrinsèques de l'Equipe, c'est-à-dire: l'autonomie, la maîtrise et le but, décrits par Daniel Pink dans le séminaire TED *The Puzzle of Motivation* [Pink 2009] et son livre *Drive: The Surprising Truth About What Motivates Us* [Pink 2011].

Le Scrum Master a un rôle de leadership, mais évite de dire aux membres de l'Equipe ce qu'il faut faire ou comment le faire. Cet aspect est décrit avec justesse dans «*Servant Leadership*» [Greenleaf 2012].

- ☑ *Beaucoup de managers traditionnels imaginent le rôle de Scrum Master comme une sorte de «secrétaire», ou un chef de projet junior. J'ai vu cette erreur de compréhension aboutir à des difficultés d'adoption de Scrum. Les promoteurs d'un changement vers l'Agile doivent rester vigilants pour identifier de tels symptômes et immédiatement sensibiliser les managers à la valeur de ce nouveau rôle.*
- ☑ *Une maladie similaire consiste à affecter un unique Scrum Master à plusieurs Equipes. Les Equipes novices ont généralement besoin d'un Scrum Master à temps plein. Ceci est particulièrement vrai lorsque le Scrum Master est également inexpérimenté et que l'organisation est en phase d'adoption de Scrum.*
- ☑ *Michael James propose une liste utile de contrôles pour toutes les activités qu'un Scrum Master pourraient faire [James 2006].*

Autres rôles

Il n'y a pas de rôle de chef de projet dans Scrum. Les responsabilités traditionnelles du chef de projet sont réparties dans les trois rôles de l'Equipe Scrum:

- Le Propriétaire du Produit gère le *Produit* (et le retour sur investissement)
- Le Scrum Master gère le *processus*
- L'Equipe de développement se *gère elle-même*

Ce fonctionnement représente un défi pour ceux qui ont actuellement un rôle de chef de projet et pour les managers des organisations dans lesquelles ils travaillent. Michele Sliger et Stacia Broderick ont écrit un guide utile pour expliquer la transition de chef de projet à Coach Agile [Sliger 2008]. Un chef de projet adopte généralement l'un des rôles Scrum : il peut-être Propriétaire du Produit s'il connaît le domaine, il peut être Scrum Master s'il a de bonnes compétences «douces», et membre de l'Equipe de développement s'il aime résoudre les problèmes.

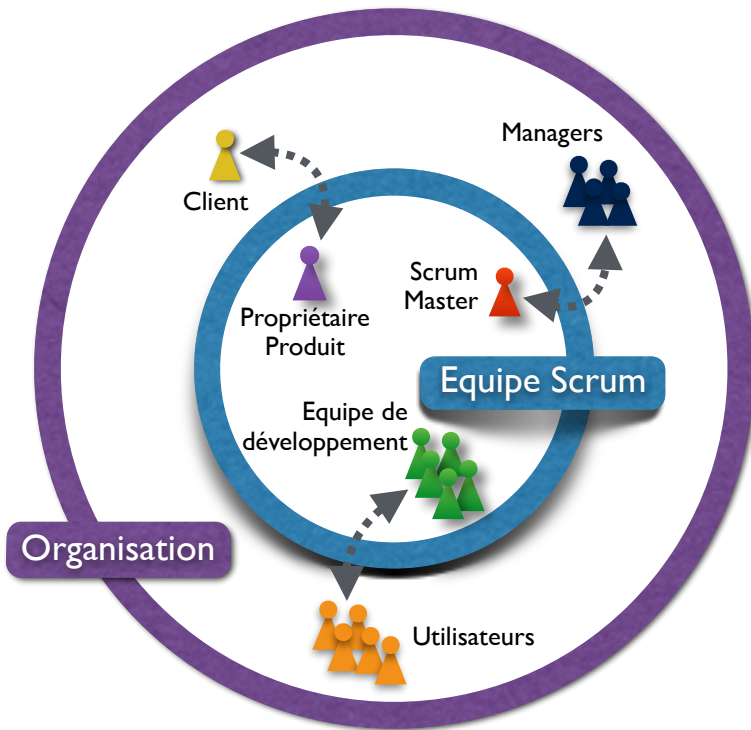
Il n'y a pas de leader désigné dans l'Equipe Scrum à part le Propriétaire du Produit et le Scrum Master : il n'y en a pas besoin. Le besoin en

management intermédiaire est réduit, les Equipes se gérant en majorité elles-mêmes. J'ai vu des équipes de 40 membres rendre compte directement à un manager dans une organisation ayant réalisé la transition vers Agile.

En dehors de l'Equipe Scrum il y a naturellement d'autres rôles au sein de l'organisation. Je trouve utile d'en distinguer trois et de montrer leurs interactions avec les rôles Scrum dans un modèle de communication simplifié:

- Le *Client*, qui finance les travaux, et qu'on appelle parfois commanditaire ou Sponsor. Il parle surtout avec le Propriétaire du Produit. Heureusement il participe également à l'évènement Revue du Sprint
- Les *Utilisateurs* qui utiliseront les livrables de l'Equipe. Ils interagissent principalement avec cette dernière pour exprimer leurs besoins et lui fournir un retour direct sur ce qui a déjà été livré. Les utilisateurs collaborent souvent en rédigeant les Récits Utilisateurs (User Stories), en affinant le Carnet de Produit, en faisant des retours au cours de la planification du Sprint et des commentaires pendant les Revues de Sprint
- Les *Managers*, qui fournissent le cadre organisationnel dans lequel l'Equipe Scrum évolue. Le Scrum Master s'occupe le plus souvent des managers et des autres parties prenantes pour leur expliquer la meilleure manière de soutenir l'Equipe et la rendre efficace

Les interactions entre ces rôles sont présentées dans le diagramme suivant. Notez que ce n'est qu'un modèle qui illustre les principaux flux de communication entre les trois rôles de l'Equipe Scrum et les principaux groupes dans l'organisation. Dans la pratique les réseaux de communication dans les organisations sont beaucoup plus complexes.



Flux simplifié de communication avec Scrum

Les événements Scrum

Le Sprint

Le Sprint rythme le cycle Scrum et encapsule tous les autres événements. Il est délimité par la planification de Sprint au début et la revue de Sprint ainsi que la rétrospective de Sprint à la fin.




Le but du Sprint est d'atteindre l'Objectif du Sprint. Ceci passe par la planification et la livraison des éléments du Carnet de Produit (Product Backlog) au Propriétaire du Produit. Les éléments du Carnet de Produit finis sont ajoutés au Produit. Le Propriétaire du Produit peut décider de lancer une nouvelle version à tout moment avec l'ensemble de ces éléments. La qualité de l'Incrément est assuré du fait d'un accord sur la *Définition du Terminé* (Definition of Done) qui permet d'évaluer la réussite et de mesurer chaque élément. Le Sprint sert à limiter le risque que l'organisation dépense des compétences et des ressources rares pour un travail inutile au moment où il est fait.

- La Définition du Terminé (Definition of Done, DoD) est un accord construit entre l'Equipe et le Propriétaire du Produit. Il est rendu public pour que tout le monde sache ce qu'il inclut et ce qu'il ne couvre pas*
- La définition du fait est une proposition générale de l'Equipe qui s'applique à tous les éléments du Carnet de Produit*
- Chaque élément du Carnet de Produit aura des conditions de satisfaction et des critères d'acceptance*

Les Equipes Scrum choisissent une durée de Sprint d'une, deux, trois ou quatre semaines. La longueur du Sprint est fixe et n'est jamais prolongée une fois que le Sprint a commencé. Cela devient le rythme de travail de l'Equipe. Chaque événement dans Scrum est strictement limité dans le temps. Cette durée est une borne maximale, qu'il n'est pas nécessaire d'utiliser complètement. La planification (phases 1 & 2), la revue et la rétrospective ont généralement une durée d'une heure par semaine de Sprint. Par exemple, pour un Sprint de deux semaines, chacun de ces quatre événements a une durée maximale de deux heures. Durant toute la

durée du Sprint, l'Equipe se retrouve pour une Mêlée Quotidienne (Daily Scrum).

Si à un moment quelconque du Sprint, l'Equipe ou les parties prenantes s'aperçoivent que l'objectif du Sprint n'est plus réalisable ou ne servira plus les besoins de l'organisation, le Propriétaire du Produit est autorisé à arrêter le Sprint immédiatement. Après cet arrêt, tout élément Terminé (Done) sera soumis à une revue. Je propose également que l'Equipe réalise une rétrospective, l'arrêt brutal d'un Sprint étant un événement traumatisant pour eux. Les arrêts de Sprints sont coûteux et rares. Certains attributs majeurs des événements sont exposés dans les sections suivantes. Avant cela cependant, j'ai rassemblé quelques expériences qui, je pense, méritent d'être partagées.

-  *Je trouve souvent que des Sprints de deux semaines sont une bonne durée de départ. Cette durée est un bon équilibre entre une semaine, qui peu paraître extrêmement court pour une Equipe novice, et trois à quatre semaines qui peuvent mener à des "mini cycles en V". Après trois Sprints, laissez l'Equipe ré-évaluer la durée des Sprints et en essayer une nouvelle si elle le souhaite.*
-  *Néanmoins, je fais souvent commencer de nouvelles Equipes Scrum avec une durée d'une semaine lorsqu'elles ont des échéances court terme. Ces cycles courts permettent plus de feedbacks, qui sont cruciaux pour aider l'Equipe à se concentrer sur la livraison rapide des éléments les plus importants pour les parties prenantes*
-  *Les Equipes ont besoin d'au moins trois Sprints pour appréhender les nouveaux concepts, rompre avec leurs vieilles habitudes et prendre la forme d'une Equipe. Cette règle des trois Sprints est valable pour chaque changement, que ce soit pour ajouter ou retirer un membre, modifier la durée des Sprints, etc.*

☑ Comme le conseille Jean Tabaka [Tabaka 2006], ne faites jamais de planification de Sprint le lundi matin. L'Equipe n'est pas encore à son meilleur niveau, et c'est le meilleur jour pour être en vacances ou malade. De la même manière, ne faites jamais de revues ou de rétrospectives le vendredi après-midi. L'Equipe est fatiguée et pense surtout au week-end. En conséquence, positionnez les limites du Sprint entre le lundi et le jeudi.

☑ Une Equipe travaillant sur des Sprints de deux semaines pourrait être tentée de regrouper tous les événements Scrum sur une seule journée. En d'autres termes, l'Equipe commence par exemple la journée par une revue de Sprint, puis une rétrospective. Après le déjeuner, elle enchaîne avec les phases 1 et 2 de la planification du Sprint. La réflexion sous-jacente est de se débarrasser de toutes les réunions en une journée pour avoir 9 jours complets de travail. D'après mon expérience, ce type de fonctionnement a deux inconvénients:

- ◆ L'Equipe ne comprend pas que ces réunions font partie du travail - en fait une des parties les plus importantes pour bien faire
- ◆ Durant la dernière partie de la journée, la seconde partie de la planification, l'Equipe n'a plus de cerveau

Dans une telle situation, je vous conseille de réaliser la revue et la rétrospective en après-midi et de planifier le lendemain matin. Là encore, laissez l'Equipe essayer ce qu'elle souhaite!

Planification de Sprint (Sprint Planning)

La planification de Sprint marque le départ de chaque Sprint. La planification permet à l'Equipe de planifier le travail à faire au cours du Sprint. Elle est généralement divisée en deux parties, chacune ayant une finalité spécifique.

La première partie de planification de Sprint (souvent désignée comme SP1) permet de répondre à la question: "Que pouvons-nous livrer d'ici la fin de ce Sprint". Nous appelons donc souvent SP1 l'événement "**Quoi**". C'est vraiment un atelier pour détailler les exigences. En partant du haut du Carnet de Produit, le Propriétaire du Produit présente l'ensemble des

fonctionnalités qu'il souhaiterait obtenir. L'Equipe de développement pose des questions pour affiner suffisamment les exigences de façon à prévoir ce qu'elle peut développer durant le Sprint. L'Equipe de développement décide seule de ce qui est réalisable, en tenant compte de la durée du Sprint, de la taille et des capacités actuelles de ses membres, de sa définition du fait, des vacances et des jours de congés, des interruptions potentielles de travail, et surtout des améliorations définies lors de la rétrospective qui a eu lieu juste avant cet événement.

Le Propriétaire du Produit est présent tout au long de SP1 pour diriger l'Equipe de développement dans la bonne direction et répondre aux questions, et il y en aura beaucoup. Le Scrum Master s'assure que tous les intervenants nécessaires à l'Equipe de développement pour comprendre les besoins sont présents ou disponibles par téléphone.

Toutes les nouvelles fonctionnalités non prévues jusqu'à lors, et qui n'ont pas été estimées, seront immédiatement dimensionnées lors de cet événement. Ce ne doit cependant pas devenir une excuse pour ne pas affiner le Carnet de Produit (voir ci-dessous).

A la fin de la première partie, l'Equipe s'engage auprès du Propriétaire du Produit sur un périmètre qu'elle pense livrer sous la forme de fonctionnalités opérationnelles présentant des tests d'acceptance automatisés (running tested features). Une Equipe ayant un historique stable de fonctionnalités livrées par Sprint peut utiliser cet indicateur de vitesse comme un indicateur (plus connu sous le nom de "prévision météo"). Ce mode de fonctionnement s'appelle planification basée sur la vitesse (velocity-based planning). Les Equipes qui débutent peuvent utiliser une planification basée sur l'engagement (commitment-based planning), c'est-à-dire que l'Equipe utilise son intuition ou ses tripes pour savoir ce qu'elle peut faire. Cela semble peu scientifique, mais à ma connaissance, cela oblige les membres à réfléchir et conduit à d'assez bons résultats. Les éléments du Carnet de Produit (Product Backlog) que l'Equipe prévoit de réaliser au cours du Sprint constituent le Carnet de Sprint (Sprint Backlog).

Si la première partie est un atelier concernant les besoins, la deuxième partie de la planification du Sprint (SP2) est un atelier de conception qui permet de répondre à la question: "Comment ferons-nous le travail?". Nous appelons donc souvent cette partie de la planification du Sprint (SP2) l'évènement «**comment**».

Dans cette session, l'Equipe de développement collabore pour créer une conception de haut niveau des fonctionnalités qu'elle s'est engagée à livrer. Au cours de cette deuxième partie, l'Equipe peut avoir des questions supplémentaires concernant les besoins. Le Scrum Master doit veiller à ce que le Propriétaire du Produit et, le cas échéant, d'autres acteurs soient disponibles.

La conception est émergente, comme tout en Agile. De plus, l'événement a une durée limitée. Il est donc normal que la conception de l'Equipe ne soit pas parfaite à l'issue de la session et qu'elle découvre d'autres tâches au cours du Sprint. Ce n'est pas le signe d'un dysfonctionnement. L'Equipe va juste prendre des Post-it et des crayons pour créer des tâches lorsque cela est nécessaire au cours du Sprint.

Un des extraits de la planification du Sprint est le Carnet de Sprint, contenant les éléments fonctionnels à livrer, la liste des tâches et le plan dont l'Equipe a besoin pour développer collectivement les éléments du Carnet de Produit et les transformer en fonctionnalités opérationnelles présentant des tests d'acceptance automatisés. Le Carnet de Sprint prend le plus souvent la forme d'un vrai tableau de tâches (un tableau avec des post-it - NdT). Les Equipes Scrum qui ne sont pas installées sur un même site utiliseront probablement un logiciel pour visualiser leur tableau de tâches.



Vous saurez que votre seconde partie de planification de Sprint fonctionne bien lorsque les membres de l'Equipe seront réunis autour du tableau blanc, discutant bruyamment ou même argumentant à propos de la "meilleure" ou de la "bonne" manière d'implémenter une fonctionnalité

Résumé:

But

Planifier le travail pour le Sprint courant

Timing

Au début de Sprint, une heure par semaine de Sprint pour chaque partie

Public

L'Equipe Scrum ainsi que tous les acteurs concernés

Intrants

Carnet de Produit affiné; axes d'amélioration de la précédente rétrospective de Sprint; Objectif du Sprint (si déjà connu).

Extrants

Carnet de Sprint

Mêlée Quotidienne (Daily Scrum)

La Mêlée Quotidienne (Daily Scrum) est l'un des premiers leviers d'inspection et d'adaptation dans Scrum. L'Equipe de développement communique pour synchroniser ses travaux et créer un plan pour les prochaines 24 heures. Etant donné que l'Equipe s'auto-organise pour réaliser l'Objectif du Sprint, cet échange est essentiel pour garantir la démarche d'amélioration continue et éviter les blocages dans le travail

L'événement de Mêlée Quotidienne ne doit pas servir de compte-rendu d'avancement pour le Scrum Master ou qui que ce soit d'autre. En fait, le Guide Scrum indique que le Scrum Master doit veiller à ce que seuls les membres de l'Equipe participent [Schwaber 2013]!

Le Scrum Master doit veiller à ce que la Mêlée Quotidienne ait lieu. Il doit également veiller à ce que tout obstacle sur la route de l'Equipe soit aplani le plus vite possible. Le Scrum Master s'assure également que l'événement ne dure pas plus de 15 minutes.

La Mêlée Quotidienne est parfois appelé le stand-up, car elle doit se dérouler debout. Ceci oblige à faire court ! Jason Yip [Yip 2006] propose un guide utile pour aider les Scrum Masters à bien gérer cet événement.



Depuis la version de juillet 2013 du Guide Scrum, le Propriétaire du Produit et les autres parties prenantes ne sont plus acceptés dans la Mêlée Quotidienne. Etant donné l'objectif de cet évènement, je ne comprends pas pourquoi ils voudraient y participer.

Résumé:

But

Synchroniser l'Equipe de développement, planifier le travail pour la journée en cours

Timing

Tous les jours à la même heure; durée de 15 minutes au maximum

Public

L'Equipe de développement; Scrum Master sert de facilitateur

Intrants

Carnet de Sprint

Extrants

Mise à jour du Carnet de Sprint; (éventuellement) mise à jour de la courbe d'atterissage de Sprint (Sprint Burndown Chart)

Revue de Sprint (Sprint Review)

L'objectif de la revue de Sprint est le *Produit* que l'Equipe a construit.

La revue de Sprint est parfois, et à tort, appelée "Sprint demo". Bien qu'on y retrouve une démonstration des nouvelles fonctionnalités développées au cours du Sprint, son objectif principal est de présenter ce que l'Equipe a livré et de recueillir les commentaires des participants pour adapter le plan du prochain Sprint. Ainsi, elle est plus qu'une simple démonstration : c'est la pierre angulaire qui assure que la trajectoire du Produit prend en compte les feedbacks au moins mensuellement.

Lorsqu'on me demande qui devrait être invité à la revue de Sprint, je réponds "le monde entier". Mon intention est ici d'aider le Scrum Master et l'ensemble de l'organisation à comprendre que l'attention directe et les remarques d'une base élargie de l'organisation permet de maximiser la valeur que l'Equipe va livrer dans les Sprints suivants. Les Revues de Sprint ont de nombreux résultats possibles, y compris l'arrêt du projet. Dans la plupart des cas, l'Equipe est autorisée à débiter un autre Sprint. Il me semble que les Equipes matures peuvent déjà établir le but de leur prochain Sprint lors de cette Revue. Elles trouvent que cela aide à maintenir le rythme d'un Sprint à l'autre.

☑ *La revue de Sprint est un évènement particulièrement coûteux étant donné qu'elle regroupe de nombreuses parties prenantes et personnes expérimentées. L'Equipe doit respecter ceci en préparant et vérifiant l'environnement de démonstration avant le rendez-vous*

☑ *J'apprends aux Equipes à ne pas surprendre le Propriétaire du Produit pendant la Revue de Sprint. Il devrait être avisé, avant le rendez-vous, des éléments du Carnet de Produit terminés qui seront présentés*

Résumé:

But

L'Equipe Scrum présente l'Incrément et adapte leur plan en fonction des retours des intervenants

Timing

À la fin de Sprint, maximum une heure par semaine de Sprint

Public

L'Equipe Scrum ainsi que toutes les parties prenantes

Intrants

Incrément; Carnet de Produit raffiné

Extrants

Mise à jour du Carnet de Produit, mise à jour de la courbe d'avancement du Produit / livraison; But du prochain Sprint (idéalement)

Rétrospective du Sprint (Sprint Retrospective)



La rétrospective de Sprint (Sprint Retrospective) est l'évènement final du Sprint. Il suit immédiatement la revue de Sprint. Il n'est jamais omis, peu importe ce qui arrive durant le Sprint!

Si la revue de Sprint est axée sur le Produit, la rétrospective se concentre sur le processus - la manière dont l'Equipe Scrum travaille ensemble, y compris les compétences techniques et les pratiques et outils qu'ils utilisent pour le développement. Si la revue de Sprint est ouverte à tout le

monde, la rétrospective est limitée aux seuls membres de l'Equipe, c'est à dire le Propriétaire du Produit, les membres de l'Equipe et le Scrum Master. Les autres, y compris les managers de tous les niveaux de l'organisation, ne doivent pas être présents sauf si, par consensus, l'Equipe les a expressément invités.

Cette règle doit être comprise au regard de l'objectif de l'événement, qui est d'analyser finement la manière dont l'Equipe collabore et exécute son travail et pour définir les axes d'amélioration. Ceci exige souvent un partage profond et une introspection, qui à son tour nécessite un environnement sûr et sécurisant. C'est ce que sous-entend Norman Kerth avec sa Première Directive de Rétrospective : "En dépit de ce que nous découvrons, nous comprenons vraiment et croyons que tout le monde a fait de son mieux, compte tenu de ce qu'il savait à l'époque, de ses compétences et capacités, des ressources disponibles et la situation présente." [Kerth 2001]

Boris Gloger [Gloger 2008] propose un modèle simple appelé "Heartbeat Retrospectives" (Pouls des Rétrospectives - NdT) pour apprendre aux nouvelles Equipes comment faire des rétrospectives intéressantes. Esther Derby et Diana Larsen [Derby et Larsen 2006] fournissent un excellent modèle et une ensemble d'activités utiles pour les animateurs de rétrospectives. Jeff Sutherland [Sutherland 2011] a produit une très courte vidéo (2:30) qui capture l'essence de ce qu'est une rétrospective de Sprint.

-  *La présence du Propriétaire du Produit durant la Rétrospective de Sprint est nécessaire. Il est un membre à part entière de l'Equipe. En tant que tel, sa participation à l'évaluation et l'amélioration du processus de l'Equipe Scrum est aussi importante que celle des autres membres*
-  *Le rôle essentiel du Scrum Master est de garantir la sécurité psychologique de chacun durant la rétrospective. Les Equipes novices ayant un Propriétaire du Produit autoritaire (cela arrive !) peuvent avoir besoin de tenir leurs premières rétrospectives sans lui jusqu'à ce qu'une sécurité suffisante soit garantie. De telles actions doivent toujours être temporaires et ne sont que des étapes sur la voie vers l'agilité*



Norm Kerth recommande que les Equipes investissent 2% de leur temps de travail dans les rétrospectives. Je trouve que c'est un bon conseil. Et ce n'est certainement pas un coût trop élevé à payer pour apprendre à travailler plus efficacement et prendre plus de plaisir ensemble. Malgré cela, il y a des Equipes qui ne le prennent pas ou passent cette étape. Alan Cyment, Formateur (CST) va jusqu'à dire que tout ce dont vous avez besoin pour Scrum est un excellent Scrum Master et commencer à tenir des rétrospectives!

Résumé:

But

L'Equipe Scrum trouve les manières d'améliorer continuellement sa façon de travailler

Timing

À la fin du Sprint (après la revue de Sprint), maximum une heure par semaine de Sprint

Public

L'Equipe Scrum (d'autres uniquement sur invitation de l'Equipe Scrum)

Intrants

Tout ce qui peut être utile provenant du Sprint courant

Extrants

Éléments d'amélioration à mettre en œuvre au cours du prochain Sprint

Artefacts Scrum

Scrum propose uniquement les artefacts suivants:

- Le Carnet de Produit (Product Backlog)
- Le Carnet de Sprint (Sprint Backlog)
- L'Incrément

Bien que n'étant pas considéré comme artefact Scrum, je considère que la *Vision du Produit* de l'Equipe et sa *Définition du Terminé* (Definition of Done) sont des éléments importants. J'enseigne à toutes les Equipes Scrum qu'il faut créer et rendre ces deux éléments très visibles dans leur espace de travail.

Scrum passe volontairement sous silence tous les autres documents et artefacts. Cela conduit parfois à un malentendu où les Equipes agiles n'auraient pas besoin de rédiger de documentation. J'entraîne les Equipes à ne produire que les éléments qui leurs sont vraiment nécessaires ou pourront être requis à d'autres dans l'avenir. Les questions qui permettent de juger de la nécessité sont: "Pour qui est-ce? " et "Que vont-ils en faire?". Cela réduit le travail inutile et la déforestation!

Le Carnet de Produit (Product Backlog)





Le Carnet de Produit est simplement une liste d'éléments, décrits à un niveau fonctionnel, qui doivent être réalisés au fil du temps. Les éléments peuvent être ajoutés par n'importe qui, et seul le Propriétaire du Produit a le droit (et le devoir) de déterminer l'ordre dans lequel ils seront développés par l'Equipe. Bien sûr, un bon Propriétaire du Produit va négocier cet ordre avec les parties prenantes et l'Equipe de développement.

Les besoins sont *émergents*, ce qui signifie que nous ne pouvons pas connaître à l'avance tous les détails du Produit. Par conséquent, le Carnet de Produit est un document vivant et nécessite un *raffinement* constant pour le conserver à jour et garder son utilité. Beaucoup de nouveaux éléments seront ajoutés au fil du temps. Certains éléments seront découpés en plusieurs plus petits, d'autres peuvent être enlevés lorsqu'ils seront perçus comme non nécessaires. En outre, il faudra peut être les dimensionner pour déterminer une relation probable entre leur valeur, leur

temps de développement et le coût induit. Et, bien sûr, l'ordre des éléments dans le Carnet de Produit va changer du fait que la valeur relative des éléments va évoluer au fil du temps.

Par conséquent le raffinement du Carnet de Produit est une activité continue parallèle à chaque Sprint. Le Propriétaire du Produit organise des réunions régulières où l'Equipe Scrum et, si nécessaire, d'autres intervenants, vont affiner le Carnet de Produit pour les Sprints à venir.

Dans de nombreux cas, il est suffisant, et souvent préférable, de créer et maintenir le Carnet de Produit sous la forme d'histoires écrites sur des cartes de 125 x 75 mm. Ron Jeffries [Jeffries 2005] a inventé le triplet allitératif Carte, Confirmation, Conversation (3C) pour expliquer la manière de travailler avec des histoires. Les histoires sont écrites du point de vue d'un des protagonistes. Le livre de Mike Cohn vous explique comment vous servir des histoires utilisateurs (User Stories) [Cohn 2004].

-  *Un bon Propriétaire du Produit sera capable de dire «non» à l'ajout d'éléments non essentiels au Carnet de Produit. Il s'agit d'une activité essentielle pour renforcer la simplicité, dixième principe du Manifeste Agile*
-  *D'après mon expérience, les membres de l'Equipe ont besoin de consacrer entre 5 et 10% d'un Sprint à préparer les suivants. C'est une pratique importante pour éviter les phénomènes d'à-coup (stop-start) entre chaque Sprint. La conséquence est bien sûr, que les Equipes n'ont que 90 à 95% de leur temps disponible pour le Sprint en cours*
-  *J'entraîne les Equipes à tenir des réunions de raffinement hebdomadaires, quelle que soit la longueur de leur Sprint. Généralement, ces réunions durent une à deux heures et ont lieu à la même heure chaque semaine. Je constate que des réunions régulières et courtes sont plus efficaces et moins énergivores pour les participants*
-  *Entre ces réunions, le Propriétaire du Produit avec les business analystes, si l'organisation en a, doit travailler continuellement à affiner le Carnet de Produit. Ceci passe entre autre par la définition des conditions de satisfaction (également connu sous le nom de scénarios) pour les éléments à venir du Carnet de Produit*



Un des résultats les plus surprenants du raffinement est la montée en compétence de l'Equipe sur le domaine métier pour lequel elle travaille. L'impact positif de ceci est une meilleure adéquation aux besoins métier et un temps moindre à refaire le travail (une forme de gaspillage). Il s'avère que tout le monde est gagnant lorsque le Carnet de Produit est maintenu en bonne condition!

Carnet de Sprint (Sprint Backlog)

La plupart des Equipes de développement visualisent leur Carnet de Sprint sur un *tableau de tâches*, qui est la liste des éléments que l'Equipe s'est engagée à faire pendant le Sprint. Le tableau de tâches est un exemple de *kanban*, un mot japonais signifiant "signal visible". Il montre à l'Equipe Scrum et à quiconque, le travail prévu pendant le Sprint et l'état d'avancement actuel

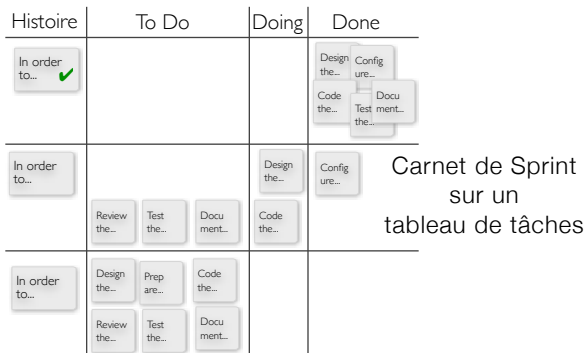


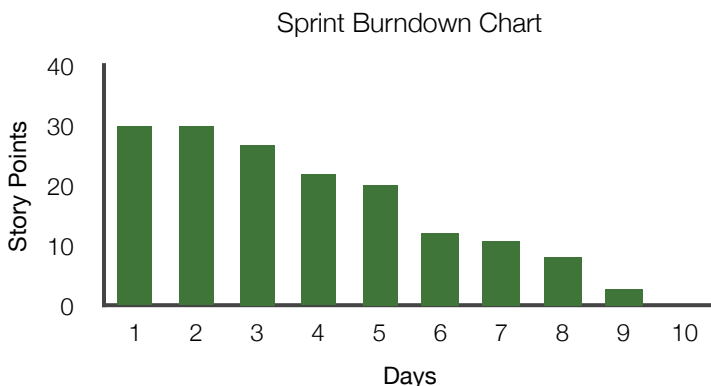
Tableau d'Atterrissage de Sprint (facultatif) (Sprint Burndown Chart)

Le Tableau d'Atterrissage de Sprint n'est pas obligatoire dans Scrum, mais l'Equipe est responsable de l'atteinte de l'objectif Sprint. Le Tableau d'Atterrissage de Sprint est conçu comme un indicateur permettant de savoir si l'Equipe va atteindre l'Objectif du Sprint.

J'encourage les Equipes à construire un Tableau d'Atterrissage à base de Points d'Histoire (Story Points). Les motivations sous-jacentes sont que:

- L'estimation de nouvelles tâches et la ré-estimation des tâches en cours exige des efforts
- Les estimations pour les tâches individuelles sont très imprécises
- L'estimation en heures/journée renvoie aux mauvaises habitudes de travail traditionnelles
- L'achèvement d'une tâche ne fournit pas de valeur. Seules les histoires complétées (fonctionnalités) en apportent
- Par conséquent, tous les efforts d'estimation fine n'apportent pas de valeur ajoutée et augmente simplement le gaspillage

Donc, dans mon expérience de coaching le Tableau d'Atterrissage de Sprint est similaire au Tableau d'Atterrissage de Produit (ou le Tableau d'Avancement - Burnup Chart), sauf que l'axe des abscisses représente des jours plutôt que des Sprints. Jeff Sutherland soutient cette pratique depuis 2009 [Sutherland 2009].



Les détracteurs soulignent qu'un Tableau d'Atterrissage calculé en points est irrégulier plutôt que lisse, et peut avoir tendance à plafonner jusqu'à la fin du Sprint. C'est tout à fait vrai! Dès lors, il y a quelque chose d'évident à améliorer.

Incrément

L'Incrément est la somme de tous les éléments du Carnet de Produit réalisé avant la fin du Sprint et répondant à la définition du Terminé (Done). L'Equipe le présente durant la Revue de Sprint. Le Propriétaire du Produit détermine le moment où il faut livrer un Produit complet.

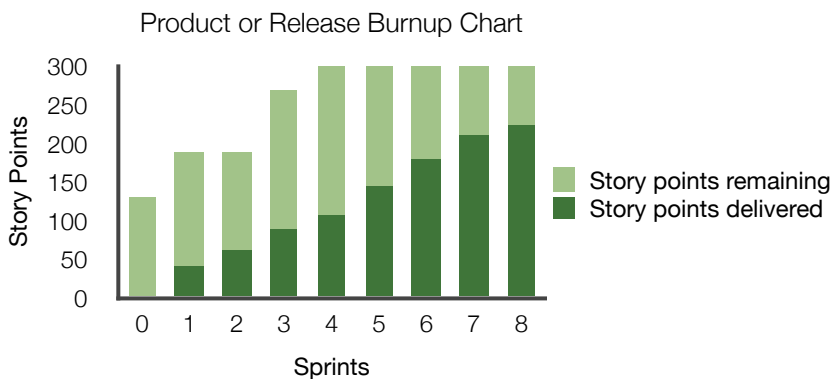
Courbe d'Avancement du Produit de livraison (facultatif) (Product or release burnup chart)

Le Courbe d'Avancement du Produit, parfois appelé Courbe d'Avancement de livraison, mesure le taux de livraison d'éléments fonctionnels et testés au fil du temps. Ce taux est souvent appelé "Vélocité" de l'Equipe. Comme les fonctionnalités dépendent de l'effort requis pour les développer, nous utilisons une échelle pour en comparer la taille. La méthode la plus connue est l'adoption d'une unité de mesure relative appelée points d'histoire (story points). Une fois qu'une Equipe a travaillé ensemble pendant quelques Sprints, elle établit généralement sa vélocité pour un intervalle de temps. Les Propriétaires du Produit utilisent alors cette vélocité pour prédire le nombre de fonctionnalités que l'Equipe va proposer au fil du temps, ce qui conduit à des plans de livraison crédibles.

Je conseille aux Propriétaires du Produit d'utiliser une Courbe d'Avancement plutôt qu'un Tableau d'Atterissage afin qu'ils puissent suivre simultanément l'avancement et les changements du Produit. Cet aspect est essentiel compte-tenu de la nature dynamique de l'artefact.

L'utilisation de cette Courbe permet aux Propriétaires du Produit d'explicitier les progrès, de déterminer les dates de livraison et de prévoir le contenu de chacune.

Encore une fois, les Courbes d'Avancement ne sont pas obligatoires dans Scrum. Cependant, je ne connais pas de meilleur moyen pour exploiter l'historique des performances d'Equipe au service d'une planification et d'une prise de décision précise et transparente. Cela renforce le principe de l'empirisme, fondamental dans Scrum.



Résumé du Chapitre

- ◆ L'auto-organisation est un prérequis à Scrum
- ◆ Scrum propose une structure d'Equipe définie avec trois rôles
- ◆ Scrum est construit autour de cycle d'évènements récurrents
- ◆ Scrum présente un nombre d'artefacts réduits

3. Adopter Scrum

Commencer Scrum

Ken Schwaber [Schwaber 2007] explique qu'il n'y a pas de prérequis avec Scrum. Mon interprétation est qu'il n'est pas nécessaire d'aménager les modes de fonctionnement existants contrairement à des méthodes plus complexes. Néanmoins, il y a peu de littérature sur la manière d'initier la démarche et nous avons tous lutté pour obtenir notre première équipe Scrum sans aide extérieure.

La meilleure chose que vous puissiez faire est d'engager un coach expérimenté. A défaut, vous pouvez essayer d'utiliser le modèle suivant, que j'ai fait fonctionner avec des dizaines d'équipes.

Évidemment, vous avez besoin d'une équipe Scrum. Cela signifie un Product Owner, un Scrum Master et trois à neuf membres pour l'équipe de développement. Si cette structure n'est pas encore en place, vous pouvez commencer avec quelques ateliers d'initiation pour vérifier l'état de préparation organisationnelle pour Scrum, identifier les parties prenantes du projet ou du produit et former l'équipe initiale qui fera le travail. Reportez-vous au chapitre 5 - Obtenir de l'aide pour plus d'informations sur ces activités.

Ensuite, suivez la séquence suivante, détaillée ci-après.

Jour 1 : Formation	#1 Former l'Equipe aux bases de Scrum
Jour 2 et 3 Atelier Produit	#2 Définir la vision
	#3 Construire le Carnet de Produit initial
	#4 Ranger les éléments du Carnet de Produit par ordre de valeurs
	#5 Dimensionner les éléments du Carnet de Produit
	#6 Ré-arranger le Carnet de Produit avec d'autres facteurs
	#7 Créer un contenu de l'itération à grosse maille

- ✓ Avec de nouvelles équipes, je commence avec une journée complète de formation sur les bases de l'Agile et de Scrum. Ces bases sont suffisantes pour une équipe si elle a un Scrum Master / coach agile avec un peu d'expérience pour l'accompagner au cours des premiers Sprints
- ✓ Les étapes deux (vision) à sept (planification de la première livraison) peuvent être réalisées facilement sur les deux jours d'atelier produit avec toute l'équipe. Les meilleurs ateliers intègrent des intervenants supplémentaires comme des architectes d'entreprise, des managers et des représentants des métiers
- ✓ Le quatrième jour l'équipe Scrum doit commencer le Sprint, sans faute!
- ✓ Si l'organisation met à disposition de l'Equipe un Scrum Master expérimenté, il sera un facilitateur durant les jours 2 à 4 tandis que je fais la formation. Dans le cas contraire, ce qui est souvent le cas, le Scrum Master novice observe. J'essaie de suivre le principe du "Un qui sait, un qui fait, un qui apprend"

#1 Former l'équipe Scrum

Dans mes formations pour les équipes, j'utilise beaucoup d'exercices de groupe et de jeux pour illustrer les principes.

Principaux sujets que je passe en revue pendant le jour 1:

Valeurs et principes de l'agile (Manifeste Agile)

Processus empiriques vs processus déterministes

Le flux Scrum (cycle des rencontres et artefacts)

Rôles et responsabilités (3 rôles Scrum et plus)

La puissance de l'auto-organisation

L'importance de l'unicité de lieu
L'importance de la confiance
Coût du multi-tasking, limitation du travail en cours
Sujets complémentaires que j'aborde si besoin pendant les jours 2 à 4:
Utiliser les histoires utilisateurs (User Story) pour comprendre les besoins
Utiliser les scénarios pour les tests d'acceptance
Concepts de taille et de vélocité
Estimation agile
Terminé! (Done)
Utiliser un tableau de tâches
Suivre l'avancement

#2 Définir la vision

Katzenbach et Smith [2002] ont confirmé que la création d'équipes très performantes passe par le partage d'objectifs clairs.

Le Propriétaire du Produit (Product Owner / PO) partage souvent sa vision du Produit. Si ce n'est pas le cas, l'Equipe va la co-crée. Une technique que j'utilise pour cela est de demander à chaque membre d'écrire sa propre version de la vision. Ensuite, les membres se mettent par deux et rédigent un seul énoncé avec leurs versions initiales. Le processus se poursuit jusqu'à ce que toute l'équipe Scrum collabore à la formulation d'une seule vision.

Cet exercice utilise la puissance du groupe et conduit à un engagement plus fort dans la vision qui en émerge. L'équipe l'affiche de façon bien visible dans leur lieu de travail. La construction de la vision prend une à trois heures.

#3 Construire le Carnet de Produit initial

L'étape suivante consiste à diriger un atelier de rédaction d'histoires utilisateur (User Story). Il est intéressant d'impliquer les parties prenantes métier dans cette étape. Toute l'équipe Scrum est évidemment impliquée.

Ecrire de bonnes histoires utilisateur est avant tout une question de pratique. Le principe de Pareto (la règle des 80 /20) s'applique ici, comme ailleurs.

Mike Cohn [Cohn 2004] propose un excellent guide de démarrage pour écrire de bonnes histoires utilisateur. J'encourage tous les Product Owner à avoir le leur sous la main !

J'aime utiliser un gabarit pour les histoires utilisateur mettant en avant la valeur proposée pour le protagoniste principal, comme dans l'exemple ci-dessous. Cela oblige à se concentrer d'abord sur les objectifs métier pour comprendre les besoins utilisateurs et planifier les travaux.

Modèle:

Afin de «raison», en tant que «rôle», je veux
"fonctionnalité"

Exemple:

Afin d'**obtenir de l'argent facilement**, en tant que **client de la banque**, je veux **retirer des billets à l'aide d'un distributeur automatique de billets (DAB)**

De plus, je suis favorable à la construction de scénarios pour spécifier les tests d'acceptation en utilisant un modèle de développement guidé par les comportements (BDD - Behavior-Driven Development - NDT) [North 2006], comme indiqué ci-dessous.

Modèle:

Etant donné que «conditions initiales»

Quand un «événement» apparaît

Alors «le résultat attendu est»

Exemple:

Etant donné que j'ai 20€ sur mon compte en banque ET
que je me suis identifié avec succès

Quand je demande un retrait de 10€

Alors je reçois deux billets de 5€ et un reçu imprimé

Bill Wake propose l'acronyme INVEST pour définir les attributs d'une bonne histoire utilisateur [Wake 2003]:

Indépendante—idéalement peut être implémentée dans n'importe quel ordre

Négociable—et négociée

créateur de **Valeur**—pour le client

Estimable—en termes d'importance relative et de planning

Suffisamment petit (Small)—et avec des descriptions courtes

Testable—je peux écrire des tests dessus

Au minimum, le Carnet de Produit doit contenir suffisamment d'éléments pour que l'équipe planifie le premier Sprint. Plus généralement, le Carnet de Produit contient tous les éléments qui devraient être livrés (on l'espère) pour la prochaine version du Produit.

A titre indicatif, le Carnet de Produit initial devrait être complet à 80% (mais pas forcément ordonné ou dimensionné) à la fin de la première journée. La première partie de la deuxième journée peut être utilisée pour finaliser les 20% restants et résoudre les questions en suspens. La nuit porte conseil et peut faire éclore de nouvelles réflexions sur le travail.

Jeff Patton propose une technique de Cartographie d'histoire (Story Mapping) pour une gestion avancée du Carnet de Produit. Vous trouverez plus de détails sur ce sujet dans le chapitre 7 – *S'améliorer*.

#4 Ranger les éléments du Carnet de Produit par ordre de valeurs

Le Carnet de Produit est maintenant trié par ordre de valeur pour l'entreprise. C'est bien sûr plus facile à dire qu'à faire. Beaucoup de Propriétaires de Produit que j'ai observés utilisent une combinaison d'OPePHS (Opinion de la Personne avec le Plus Haut Salaire), qui crie le plus fort, et "la roue de la fortune". Ces techniques ne sont pas fiables pour maximiser la création de valeur pour vos clients (sic ! - NDT).

Au minimum, le jugement subjectif du Propriétaire du Produit sur la valeur d'un élément pourrait être un point de départ. L'étape d'après serait une pondération relative et quasi-quantitative en utilisant des ordres de comparaison ou du planning poker.

Peu importe la manière, l'ordonnancement des éléments est une des principales responsabilités du Propriétaire du Produit. Le mieux serait encore de mesurer la valeur économique réelle obtenue par rapport à celle prévue.

#5 Dimensionner les éléments du Carnet de Produit

L'élément centrale de la planification d'un projet est toujours l'estimation. Les responsables de projet, comme moi, ont passé des semaines à étalonner des modèles complexes avec des données historiques.

Plus prosaïquement, la meilleure de ces techniques ne donnent pas de meilleurs résultats que des techniques plus simples et plus rapides comme le planning-poker et l'estimation par comparaison. Le planning-poker marche bien car il s'appuie sur de solides bases théoriques, et surtout parce que les gens qui estiment sont ceux qui feront le travail. Qui l'eut cru?

Le planning-poker est assez rapide. Une équipe qui a un peu de pratique peut estimer un élément toutes les 3 minutes environ. Le planning-poker est précis. Les estimations fondées sur le planning poker sont aussi bonnes que les meilleures méthodes traditionnelles. Et le planning-poker prend la forme d'un jeu. Il supprime la pénibilité souvent associée à l'exercice. Les distributeurs de cartes de planning-poker sont nombreux, avec un coût d'environ 10€ pour quatre paquets. Vous pouvez également imprimer vos propres cartes pour presque rien.

L'estimation par comparaison est même plus rapide que le planning poker. C'est une méthode idéale pour démarrer quand il faut dérouler l'ensemble du Carnet de Produit en peu de temps, avec un besoin de précision limité et la nécessité de partager l'information. J'emploie cette technique avec les équipes, qu'elles soient débutantes ou expérimentées, dès lors que l'estimation est considérée comme nécessaire et utile.

Reprenons quelques concepts clés de l'estimation agile :

- Nous utilisons des tailles d'éléments relatives. Pourquoi ? Parce qu'en tant qu'humains, cela nous semble plus naturel et les résultats sont plus fiables. Donc, nous trouvons plus facile de nous mettre d'accord sur le fait qu'une histoire demande deux fois plus d'effort que l'autre même si nous ne savons pas combien de temps il faudra pour chacune
- Nous définissons la taille des éléments en utilisant des unités d'effort plutôt que de temps. Pourquoi ? Parce que cela permet de distinguer la vitesse de l'équipe du volume de réalisation. Cela nous protège d'une modification d'estimation en fonction de qui fait le travail, ou des changements de compétences et de capacité dans les équipes au fil du temps. Nous utilisons des "Points d'histoire" (Story Points) comme unité
- Nous utilisons une échelle de points non linéaire car la différence entre un "1" et un "2" est évidemment plus significative, relativement, que entre un "20" et un "21". Ma préférence va aux valeurs de la suite de Fibonacci : 1, 2, 3, 5, 8, 13, 20, 40 et 100. Je définis les valeurs de 1 à 8 comme la gamme des tailles qu'une équipe peut réaliser au cours d'un Sprint. Pourquoi ? Parce qu'il s'avère que les humains sont assez doués pour évaluer la différence de taille dans le cadre d'un seul ordre de grandeur. Les valeurs plus élevées sont réservées aux grandes histoires (les sagas) qui devront être divisées en petites histoires avant d'être incluses dans un Sprint

#6 Ré-arranger le Carnet de Produit avec d'autres facteurs

Après le dimensionnement initial des éléments du Carnet de Produit nous pouvons constater que certains éléments doivent être réarranger. Les facteurs à prendre en compte en plus de la valeur pour l'entreprise sont:

- La taille : nous allons naturellement choisir de développer une histoire simple (petite, moins d'effort) avant une complexe (grande, plus l'effort) si elles ont une valeur similaire
- L'apprentissage : nous pouvons choisir de développer une histoire plus tôt pour aider l'équipe à apprendre davantage sur le domaine métier de l'entreprise ou sur une nouvelle technologie
- Les risques : nous pouvons choisir de développer une histoire au début car cela permettra de réduire un risque identifié. Parmi les exemples évidents, il y a tous les éléments qui aideront à établir les limites de performances et d'évolutivité

Vous devez vous rappeler que le Propriétaire du Produit a toujours le dernier mot sur le Carnet de Produit.

#7 Créer un contenu de l'itération à grosse maille




Une fois que nous avons ordonné et dimensionné le Carnet de Produit, la prochaine étape est de définir le contenu de l'itération initiale. Pour ce faire nous avons besoin d'estimer la vélocité de notre équipe. Cependant, comme nous n'avons pas encore travaillé ensemble, nous utilisons une technique simple appelée *planification basée sur l'engagement*.

Premièrement, bien sûr, nous devons être sûrs de la taille de l'équipe pour le Sprint - si un des membres prend des congés, est en formation, etc. Nous devons choisir la durée du Sprint - je recommande généralement deux semaines pour une nouvelle équipe. Ensuite nous devons créer une définition du *Terminé* pour l'équipe - que signifie avoir terminé / complété une histoire d'après l'équipe.

Une fois ceci fait, le Scrum Master prend l'élément le plus haut du Carnet de Produit et demande à l'équipe "Pouvez-vous réaliser cette histoire

pendant le Sprint ?". Il continue à le faire jusqu'à ce que l'équipe ne pense pas pouvoir en ajouter d'autres. Il reste à compter le nombre de points d'histoire de tous les éléments définis pour obtenir l'estimation initiale de *vélocité* de l'équipe.

Cette *vélocité* est utilisée pour répartir les éléments restants du Carnet de Produit (au moins ceux qui ont été dimensionnés) sur les Sprints suivants. Cette liste d'éléments par Sprints forme les itérations initiales (release plan). Sont-elles exactes ? Certainement pas, mais elles sont probablement aussi crédibles que toute prévision faite à un stade balbutiant par un chef de projet. Et comme nous réalisons le travail et le complétons un Sprint après l'autre, nous allons commencer à déterminer une *vélocité* réelle et l'utiliser comme facteur prévisionnel de la production future. Ainsi, le contenu de l'itération est un artefact vivant qui devient plus fiable à mesure que nous progressons. Voir le chapitre 7 – *S'améliorer* pour en savoir plus sur la planification des livraisons.

-  *Ne laissez pas l'atelier sur le Produit déborder. Pour n'importe quelle équipe et Produit, deux jours sont largement suffisant pour préparer assez d'éléments du Carnet de Produit pour les premiers Sprints. Il y a beaucoup plus de valeurs à créer en faisant, inspectant les résultats et adaptant vos itérations progressivement qu'en cherchant à rédiger un Carnet de Produit parfait*
-  *Avec une nouvelle équipe, un nouveau domaine ou une nouvelle technologie, essayez de ne pas créer un plan d'itérations (release plan) trop long avant d'avoir terminé les premiers Sprints. Un plan d'itération construit sur des hypothèses sauvages ne sert à rien. A la place, laissez l'équipe réaliser trois à cinq Sprints, pour offrir votre premier plan basé sur une *vélocité* observée*
-  *Scrum souligne que l'empirisme est la meilleure façon de gérer la complexité du développement de Produits. L'empirisme favorise la transparence et l'acceptation du principe de réalité. Les techniques traditionnelles de planification et de pilotage permettent de mesurer la conformité au plan. Faites attention à ne pas laisser vos techniques de planification et de pilotage pervertir les bénéfices que vous attendiez de Scrum*

#8 Planifier et commencer le premier Sprint

Ayant achevé votre atelier de Produit, il est temps de vous mettre dans les starting blocks et de commencer le travail. L'avantage avec Scrum est que nous pouvons toujours commencer un Sprint avec une nouvelle équipe au début du quatrième jour. Nous faisons une réunion de planification de Sprint (Sprint Planning Meeting), dès potron-minet. En supposant que le Sprint dure deux semaines, la réunion dure au plus deux heures pour chaque partie, soit quatre au total. Donc, pour l'heure du déjeuner ou en début d'après-midi, l'équipe peut commencer à travailler sur les premières tâches de la première histoire. C'est un exploit par rapport à la plupart des autres approches.

Résumé du Chapitre

- ◆ Scrum n'implique qu'une préparation minimale pour commencer
- ◆ Le motif pour démarrer une équipe tient en trois jours
- ◆ Il faut construire et ordonner le Carnet de Produit
- ◆ Il faut créer un contenu de l'itération initiale

4. Colocalisation et l'équipe dans l'espace

Alistair Cockburn [Cockburn 2007] définit le développement de logiciels comme *un jeu coopératif d'invention et de communication*. Le Manifeste Agile soutient que les développeurs et les personnes appartenant au métier doivent travailler ensemble tous les jours et que la conversation en face-à-face est la meilleure façon de partager des informations.

Afin de collaborer efficacement, il n'a pas de facteurs proposant plus d'intérêts que la c. Une étude portant sur des équipes de développement de logiciels colocalisées [Teasley, Covi, Krishnan, & Olson, 2000] montre qu'elles sont deux fois plus productives que les équipes géographiquement séparées.

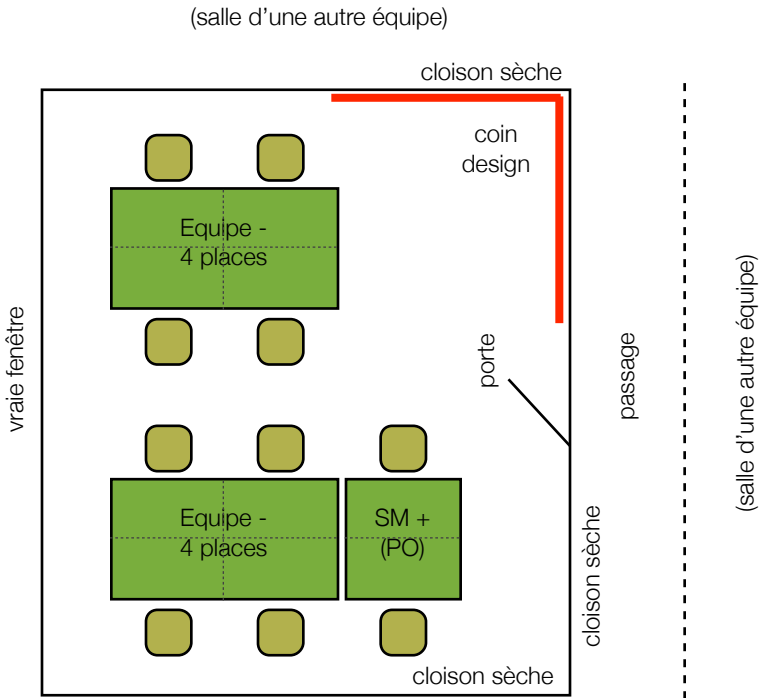
Ma définition de colocalisation implique que les membres de l'équipe sont assis à une distance maximale de 6 m (20 pieds) par rapport aux membres les plus éloignés. Cela permet de maintenir un contact visuel et verbal normal avec les autres membres de l'équipe. Il en découle que le nombre de personnes qui peuvent entrer dans un tel espace est limité. Dans la pratique, ce nombre est de 8 à 10, ce qui correspond (heureusement) à la taille maximale d'une équipe Scrum. En outre, chaque membre de l'équipe doit être en mesure de voir tous les autres membres, sans faire plus que tourner la tête ou pivoter sur sa chaise. Cela signifie qu'il n'y a pas de séparations entre les bureaux - au revoir Dilbertville !

Sans raison autre que celle de l'habitude, les organisations - et non pas les équipes - résistent au changement de leur environnement pour faciliter une colocalisation efficace. Ceci se fait en dépit des preuves qu'une équipe colocalisée ne nécessite, en moyenne, pas plus de surface au sol que pour les agencements courants. Votre management vous incite à construire des cloisons, croyant qu'elles inhibent la flexibilité. Ce n'est tout simplement pas vrai - la flexibilité est requise dans la structure de l'organisation et non dans les espaces d'équipe.

D'après mon expérience, 90% des équipes Scrum appartenant à des organisations déployant l'Agile, luttent sur la question des espaces de travail pendant un an avant que leur encadrement soit persuadé de faire

les changements nécessaires. Et une fois les changements effectués, tout le monde convient que l'amélioration est immédiate et mesurable. Pourquoi donc ?

Dans l'espoir que cette résistance absurde diminue, je vous propose, page suivante, l'esquisse d'un espace de travail pour une équipe colocalisée. Les motivations sur la géographie de cet espace va au-delà de la portée de ce petit guide - n'hésitez pas à me contacter si vous voulez plus d'informations.





Salle d'une autre équipe


Design type d'une équipe colocalisée


Quelques éléments pour l'agencement d'équipe colocalisé:

- Les bureaux sont rectangulaires pour faciliter le jumelage
- Un bon format de bureau est 1.5 à 1.8 m x 0,8 m
- L'espace entre la paroi et le bureau est d'environ 1,2 m
- Le coin "conception" comporte des tableaux blancs de 3 x 3 m
- La taille typique de la pièce est de 7 x 8 m = 50 à 60 m²
- Le Scrum Master peut voir toute l'équipe
- Le Propriétaire du produit a un siège flottant (non-permanent) dans la salle
- Les cloisons avec des fenêtres internes créent des barrières contre le bruit et assez d'espace pour diffuser l'information sans réduire la lumière

 *Un espace comme celui-ci réduit considérablement le besoin de salles de réunion—les équipes peuvent faire la planification de Sprint, les réunions quotidiennes et les revues dans leur salle*

 *Soyez conscient que votre architecte ou agenceur de bureau ne va pas beaucoup vous aider. Il n'a probablement pas été formé à l'élaboration d'espaces collaboratifs*

 *Le coût des changements (cloisons et meubles) sera amorti sur une semaine à un mois ! Ça semble incroyable, n'est-ce pas?*

 *Étant donné que Scrum repose sur des équipes auto-organisées, vous pouvez proposer à votre management de "joindre le geste à la parole" et laisser l'équipe Scrum déterminer l'agencement de son propre espace de travail. Après tout, c'est juste une décision de décoration !*

Résumé du chapitre

- ◆ Les équipes colocalisées sont deux fois plus productives que les équipes qui ne le sont pas
- ◆ Il existe un schéma simple pour un espace de travail d'équipe colocalisée
- ◆ Le coût de construction d'espaces pour équipes colocalisées peut être amorti en quelques semaines

5. Mesures

Objectifs et points d'attention

Dès que l'équipe comprend les bases de Scrum et a une certaine idée de sa vélocité - habituellement après trois sprints - il est probablement temps de commencer à mesurer. Les managers de toutes les organisations ayant un soupçon de culture de contrôle, voudront sous peu mesurer les résultats d'une équipe, ou la transition de l'organisation vers les méthodes Agiles. Sinon, comment allez-vous savoir si votre voyage vers l'Agile porte ses fruits, et quels types de corrections de trajectoire sont nécessaires ? Plus tôt vous commencez, plus vite vous aurez des données pour étayer vos décisions ultérieures.

Cependant le fait de mesurer provoque un changement dans le comportement mesuré. La célèbre citation d'Eli Goldratt pose : "Dites-moi comment vous m'évaluez, et je vous dirais comment je me comporte" [Goldratt, 2006]. C'est ici que réside le défi : choisir les mesures qui participeront aux résultats que nous souhaitons et éviter celles dont nous ne voulons pas. Ces éléments sont différents d'une organisation à l'autre, d'une équipe à l'autre et d'un moment à l'autre. Comme dit le proverbe : "Le contexte est roi". Il s'ensuit que les mesures, comme tout le reste, doivent être inspectées régulièrement et adaptées aux évolutions de nos besoins.

En outre, le gourou du management Peter Drucker nous dit : "Les travailleurs du savoir doivent se gérer eux-mêmes. Ils doivent avoir leur autonomie." [Drucker, 2001]. En suivant ce principe, nous devons impliquer les membres de l'équipe dans la définition et la collecte de leurs propres métriques. Un bon point de départ serait de comprendre pourquoi nous avons besoin de métriques tout court.

Je dirais que la valeur réelle d'une métrique est son potentiel comme outil d'apprentissage. En prenant un point de vue systémique, nous entrons dans ce que Chris Argyris appelle un apprentissage en double boucle. Il y a une double boucle d'apprentissage lorsqu'une erreur est détectée et corrigée de manière à modifier les normes, politiques et objectifs de l'organisation [Argyris 1977].

Premières métriques

Avec tout cela à l'esprit j'hésite à proposer des métriques concrètes ici. Néanmoins, je vous propose quelques mesures "pour commencer". La plupart sont simples à mettre en œuvre, certaines exigeront plus d'efforts. Elles sont basées sur mes propres recherches et celles d'autres praticiens agiles, notamment Pete Behrens [Hundermark 2009]. Je vous encourage à regarder la vidéo et les diapositives pour obtenir à la fois une certaine compréhension de ces métriques, une explication concernant le choix de ces métriques spécifiques et quelques conseils sur la manière de les mettre en place.

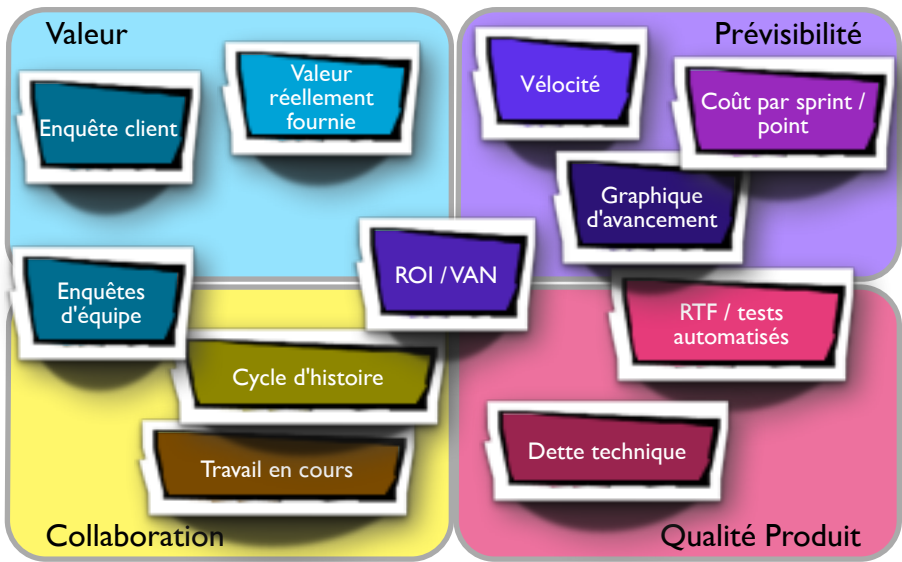
Les métriques simples:

- Sondages auprès des clients et de l'équipe
- Graphique de vélocité
- Tableau d'avancement
- Couverture en exécution de tests automatisés
- Dette technique
- Travail en cours (Work in Process / WIP) / Délai moyen de réalisation d'une histoire
- Coût par Sprint / Point / fonctionnalité

Les métriques qui exigent une capacité à mesurer la valeur:

- Valeur réelle livrée
- ROI ou VAN

Ces mesures sont représentées dans le diagramme selon quatre dimensions : valeur, prédictibilité, qualité et collaboration, qui sont décrits dans la vidéo.



Enfin, soyez vigilant sur la manière dont les mesures sont utilisées. Par exemple, un graphique de vélocité, dont le but est d'améliorer la prédictibilité, peut être utilisé abusivement pour mesurer la productivité, un but pour lequel il n'a pas été conçu et ne devrait donc pas être utilisé. Vous devez, pour chaque métrique, comparer soigneusement et régulièrement l'intention initiale avec l'utilisation réelle. Il s'agit d'une activité non négligeable.

Points clés

- ☑ *Le but des mesures est d'apprendre. Donc, commencez vite. Plus vous établissez un point de référence tôt, plus vous serez en mesure d'utiliser les données afin de comprendre vos résultats et ainsi nourrir vos systèmes pour prendre de meilleures décisions.*
- ☑ *Ne laissez pas un groupe de managers définir les métriques depuis leur «tour d'ivoire». Travaillez avec votre(vos) équipe(s) et les parties prenantes pour déterminer ce qui doit être mesuré et comment le mesurer.*
- ☑ *Ce que vous mesurez aujourd'hui n'est pas forcément la bonne chose à mesurer demain. Soyez à l'aise avec l'idée que les métriques de votre équipe sont dans un état plus ou moins constant d'incertitude. Marius de Beer, DSI et coach Agile, explique que la durée de vie médiane des métriques dans son organisation est de trois mois [De Beer 2013].*

Résumé du chapitre

- ◆ Les équipes doivent s'autoévaluer
- ◆ Les métriques sont des outils pour apprendre
- ◆ Les métriques doivent être adaptées au regard du temps et du contexte
- ◆ Quelques exemples que vos équipes pourraient vouloir essayer

6. Déployer Scrum à l'échelle de l'Organisation

Voici des dragons!

La première règle pour la mise à l'échelle de Scrum est "ne faites pas de mise à l'échelle". En d'autres termes, soyez sûr d'avoir besoin de plus d'une équipe Scrum pour faire le travail avant d'envisager l'utilisation de plusieurs équipes. Une étude montre que les équipes composées de quatre à huit membres sont 35 fois plus productive que les autres [Coplien 1994]. Jeff Sutherland, l'un des pères fondateurs de Scrum, a recensé de nombreux cas d'équipes "hyper-productives" [Sutherland 2003-2012].

Avant de tenter d'introduire Scrum dans plusieurs équipes, faites-le avec une seule équipe. Faites-le "à la lettre". Apprenez de vos erreurs. Marchez avant de courir. Faites-le bien avant d'essayer de changer les règles. Suivez le modèle d'apprentissage *Shu Ha Ri* (voir le chapitre 7 pour une explication de *Shu Ha Ri*).

Ma propre expérience de mise à l'échelle de Scrum me fait dire que commencer petit et ajouter le strict nécessaire au fil de l'eau est moins douloureux et plus fructueux que commencer avec une méthode "lourde" pour essayer de trouver les morceaux à supprimer. Toute tentative de mise à l'échelle de Scrum va amplifier le besoin d'une bonne conception et la mise en oeuvre de pratiques de développement. Soyez sûr de souligner le besoin de bien le faire. Chaque situation de mise à l'échelle est différente et nécessite sa solution spécifique. Comme pour toute chose avec l'Agile, le chemin de la transformation passe par des essais, de l'inspection et de l'adaptation.

Les plus grandes difficultés dans l'adoption d'un changement d'échelle de l'Agile sont organisationnelles. Par conséquent, toute mise en oeuvre d'une mise à l'échelle mérite une attention particulière sur les aspects de communication, de culture et de gestion du changement. Je suggère que vous acceptiez qu'il n'y ait pas de panacée pour réaliser la mise à l'échelle de Scrum (ou toute autre forme d'Agile). Avec cet état d'esprit, je vous suggère de lire attentivement ce que les autres ont fait et font. Je

recommande en particulier les livres Larman et Vodde (ci-dessus) ainsi que les articles et présentations de Henrik Kniberg. Le livre de Ken Schwaber "The Enterprise and Scrum" [Schwaber 2007] vaut le détour. "Succeeding with Agile" de Mark Cohn [Cohn 2009] propose également des conseils utiles pour appliquer Scrum à grande échelle.

Si vous êtes un novice et êtes motivé pour aider votre organisation à s'améliorer, trouvez et embauchez un coach Agile qui a une expérience significative dans le déploiement de l'Agilité au niveau que vous ciblez. Cela améliorera votre courbe d'apprentissage et atténuera la douleur !

Enfin, soyez préparé à un voyage de plusieurs années vers l'agilité organisationnelle. Si j'ai vu des améliorations considérables en quelques mois, les transformations vraiment importantes, surtout à grande échelle, prendront du temps : prévoyez cinq à dix ans. La bonne nouvelle est que vous prendrez beaucoup de plaisir avec des récompenses tout au long du chemin.

Méthodologies et modèles de changement d'échelle

Avec la popularisation de l'Agilité, il est évident que de grandes entreprises veulent l'appliquer à leur échelle. Ceci crée des défis et des opportunités pour les praticiens agiles.

Un certain nombre de contributions remarquables proviennent "des tranchées" de la pratique Agile. Le livre de Ken Schwaber *The Enterprise and Scrum* représente la première salve [Schwaber 2007]. Des contributions notables dans ce champs sont celles de Henrik Kniberg [Kniberg 2008] [Kniberg & Ivarsson 2012] et Mike Cohn [Cohn 2010]. J'accorde une place de choix au travail complet de Craig Larman & Bas Vodde, décrit dans deux livres *Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum* [Larman et Vodde 2008] et *Practices for Scaling Lean & Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum* [Larman et Vodde 2010].

Dans le même temps les méthodes plus anciennes telles que le RUP (Rational Unified Process) ont reçu un coup de peinture pour les remettre

aux couleurs de l'"Agile". Ces propositions doivent être appréhendées avec précaution et circonspection.

Le sujet de l'Agile dans l'entreprise est un champ intéressant et important, avec beaucoup de nouvelles pratiques émergentes au cours des dernières années. J'ai détaillé ce sujet de mise à l'échelle sur mon blog [Hundermark 2014].

Mon approche de mise à l'échelle

Un aphorisme utile à garder à l'esprit est "changer d'échelle pour les principes pas pour les pratiques".

Je présente ci-dessous quelques principes et pratiques connexes tirés du livre de Don Reinertsen *Principles of Product Development Flow* [Reinertsen 2009], de ceux de Larman et Vodde, de conversations avec d'autres coachs agiles, et de mes propres expériences de mise à l'échelle de Scrum avec mes propres clients dans le cadre de missions de conseil au cours des huit dernières années. Comprenez bien que ceci ne représente que le vernis de ce que signifie un changement d'échelle Agile.

Principes et pratiques pour le changement d'échelle de Scrum
Minimisez l'inter-dépendance entre les équipes, donc
Formez des équipes par fonctionnalités plutôt que par composants, et
Vérifiez que chaque équipe est multi-disciplinaire.
Tenez-vous en à la "bonne" taille d'équipe, peu importe le reste, donc
Fusionnez plusieurs produits en un seul carnet de produit pour une équipe, ou
Laissez un seul carnet de produit nourrir plusieurs équipes.
Réduisez les silos de compétences et les dépendances au sein des équipes, donc
Accompagnez la pluri-compétence (T-Shaped people).

Déployez des petits lots pour réduire le temps des cycles, la variabilité et pour augmenter l'efficacité, de sorte
Évitez les grands éléments de travail dans les sprints, et
Instituez un rythme régulier de réunions durant les sprints.
Raccourcissez le délai des travaux en attente, pour
Alimentez plusieurs équipes synchronisées à partir d'un seul carnet de produit.
Exploitez les économies d'échelle de plusieurs équipes, afin de
Synchroniser les sprints de plusieurs équipes.
Conservez du mou (slack) pour maintenir le flux (de travail), donc
Autorisez les équipes à prendre dans le carnet de produit, en fonction de leur capacité observée, et
Défiez les équipes de terminer plus tôt au moins aussi souvent qu'elles finissent plus tard.
Gardez des boucles de rétroaction courte, pour
Vérifier que les extrants des équipes sont testés et intégrés dans l'Incrément de chaque Sprint, et
Éliminer des artefacts comme «intégration» ou «consolidation» à la fin des sprints.
Optimisez l'ensemble, c'est-à-dire
Mesurez les résultats au plus haut niveau possible, et
Laissez les équipes trouver leurs propres solutions locales.
Faites attention à la qualité, pour
Vérifier que la «dette technique» est réduite, et n'augmente pas, et
Corriger les erreurs dès qu'elles sont détectées.
Faites attention à la communication, donc
Organisez des réunions formelles pour synchroniser les équipes.
Faites attention à l'apprentissage, donc
Formez des communautés de pratiques pour les différentes disciplines pour partager l'apprentissage, et

Organisez de grandes rétrospectives de groupes à une fréquence plus réduite (par exemple pour les versions).

Résumé du chapitre

- ◆ La mise à l'échelle de Scrum à l'organisation est non-triviale
- ◆ Il existe quelques modèles et principes pour réaliser la mise à l'échelle
- ◆ Il y a quelques conseils et mises en garde concernant les approches spécifiques
- ◆ Quelques conseils sur la façon de procéder

7. S'améliorer

Shu Ha Ri

Shu Ha Ri est un terme japonais venant de l'Aïkido, l'art martial japonais, pour expliquer la manière d'apprendre. La description de ce concept est basée sur un article de Ron Fox [Fox 1995].

Shu, la première étape, signifie garder, protéger ou maintenir. Pendant cette étape, l'apprenant acquiert les bases et techniques du nouvel art qu'il étudie. Il copie méthodiquement les techniques montrées par le sensei (professeur) sans modification et sans nécessairement comprendre les logiques sous-jacentes. L'idée est que l'étudiant apprendra d'autant plus vite qu'il suit le chemin tracé par le sensei. C'est ce dernier qui déterminera quand l'étudiant est prêt à passer à l'étape suivante, *Ha*. Cela implique que l'étudiant *Shu* soit à proximité de son *sensei*. Il en résulte aussi que le *sensei* devrait être *Ri* pour transmettre les connaissances et être le mentor de ses élèves.

Ha, la deuxième étape, signifie se détacher. L'élève réfléchit sur le sens et le but de ce qu'il a appris. Il développe une compréhension plus profonde de l'art au-delà de la simple répétition. Les techniques ont été intégrées comme des réflexes. Il s'affranchit de la tradition dans une certaine mesure.

La troisième étape, *Ri*, signifie transcender ou aller au-delà. L'étudiant est maintenant un praticien qui pense par lui-même et confronte ses idées à la réalité. Au niveau *Ri* l'élève peut dépasser son *sensei*, faisant ainsi progresser l'art.

En quoi est-ce important ? Eh bien, je vois beaucoup d'équipes et de managers dans les organisations qui sont trop impatients et ne posent pas les bases solides dans la phase *Shu*. Certains veulent absolument atteindre *Ri* tout de suite ! Comme Ron Jeffries, agiliste de longue date, déclare : "Mon conseil est de commencer 'by the book' (à pratiquer en suivant la théorie - NdT), de devenir bons dans les pratiques, puis de faire ce que vous voulez. Beaucoup de gens veulent passer directement à l'étape trois. Comment le pourraient-ils ?"

Si le chapitre 3 – *Adopter Scrum* - se concentre surtout sur le fonctionnement au niveau Shu, ce chapitre est destiné au niveau Ha. En poursuivant votre voyage dans l'Agilité, vous verrez que Scrum offre un cadre minimal suffisant pour résoudre des problèmes complexes. L'amélioration continue peut être atteinte simplement en faisant bien l'essentiel. Et quand vous serez prêt, vous découvrirez qu'il existe aussi un ensemble d'outils utiles pour compléter ce que vous savez. Au niveau Ha les bases sont des "réflexes conditionnés" et vous pouvez commencer à expérimenter de nouvelles pratiques sans oublier les fondamentaux de l'Agilité.

Un modèle pour l'hyper-productivité

Jeff Sutherland définit une Équipe Scrum comme *hyper-productive* une fois qu'elle a augmenté sa vélocité de 400% par rapport à son état initial. Nous pourrions soutenir que le seul accroissement de la vélocité n'est pas une mesure suffisante, voire même que c'est un indicateur dangereux s'il est isolé. Nous voulons toujours plus : améliorer la qualité et livrer plus rapidement ; avoir une meilleure prédictibilité en plus de la vélocité ; construire des produits et fonctionnalités dont les clients ont besoin plutôt que simplement développer plus de fonctionnalités. Pourtant, pour le reste de cette section mettons de côté notre incrédulité et concentrons nous sur un ensemble de modèles d'amélioration de processus que Jeff et ses collaborateurs ont développé et dont ils pensent qu'ils aident toutes les équipes Scrum à obtenir beaucoup, plus rapidement [Sutherland 2013].

Ma propre conviction, construite en expérimentant avec de nombreuses équipes, est que ces modèles sont fondamentalement sains et conduiront non seulement à l'augmentation de la productivité, mais aussi à beaucoup d'autres améliorations mentionnées ci-dessus. Essayez-les. Qu'avez-vous à perdre?

Par souci de concision, je mentionne simplement chacun des neuf modèles qui composent *Les Patterns pour les équipes Hyper-productives*. J'ai ajouté un bref commentaire lorsque cela semblait nécessaire. Si vous souhaitez mettre en œuvre ces modèles, je vous invite à télécharger gratuitement le PDF avec les descriptions complètes des modèles par l'intermédiaire du lien sur le site Web de Jeff [Sutherland 2013].

Notez que les modèles décrits ici sont des modèles génératifs. Cela signifie qu'ils fonctionnent indirectement en s'attaquant à la structure sous-jacente du problème qui peut ne pas être manifeste. Les modèles génératifs conduisent à des comportements émergents, ce qui correspond tout à fait aux catégories de problèmes complexes auxquels Scrum cherche à répondre.

Les équipes stables

Conservez des équipes stables et évitez de déplacer les personnes entre les équipes. Une équipe stable apprend à connaître sa capacité, ce qui permet à l'entreprise de définir une certaine prédictibilité.

La météo d'hier

Dans la plupart des cas, le nombre de points terminés durant le dernier Sprint est le facteur prédictif le plus fiable pour estimer le nombre de points du prochain Sprint.

Les points d'estimation décrits ici sont souvent appelés Points d'Histoire (d'après les Histoires Utilisateur / User Stories - NdT). Dans de nombreux cas compter les histoires est suffisant (les éléments du carnet de produit).

Mode essaim

Concentrez au maximum les efforts de l'équipe sur un élément du Carnet de Sprint pour le terminer dès que possible. Celui qui prend un élément devient le Capitaine de l'équipe. Tout le monde doit l'aider dans la mesure du possible et personne ne doit l'interrompre. Dès que le Capitaine a Terminé (Done), celui qui prend la responsabilité de l'élément suivant du carnet est le nouveau Capitaine.

Les membres de l'équipe et les managers peuvent considérer que ce n'est pas efficace. C'est peut-être vrai. N'oubliez pas que notre objectif n'est plus *l'efficacité*, mais *l'effectivité*.

Pattern (Motif, comme dans design pattern - NdT) d'interruption

Prévoyez du temps pour les interruptions (et les urgences - NdT) et ne le dépassez pas. Mettez en place ces trois règles simples qui vont conduire l'entreprise à s'auto-organiser pour éviter de perturber la production:

- 1. L'équipe crée un tampon (de temps / charge - NdT) basée sur des données historiques*
- 2. Toutes les demandes doivent passer par le propriétaire du produit pour arbitrage*
- 3. Si le tampon commence à déborder, l'équipe doit automatiquement interrompre le Sprint automatically abort the Sprint*

Abandonnez un Sprint a des répercussions sur les dates de livraison. La direction prend conscience rapidement du coût et travaille pour résoudre la cause sous-jacente.

Du Code Propre tous les soirs

Corrigez tous les bugs dans la journée. Le but est d'avoir une base de code totalement propre à la fin de chaque journée.

Bien que Scrum soit silencieux sur les pratiques de développement, la discipline autour du code et de la qualité de conception est la clé de voûte de l'agilité.

Procédure d'urgence

Lorsque vous êtes trop haut sur votre Courbe d'Avancement (Burndown chart) essayez cette technique utilisée couramment par les pilotes. Quand les choses tournent mal, exécutez la procédure d'urgence conçue spécifiquement pour le problème. N'en retardez pas l'exécution en essayant de comprendre ce qui ne va pas ou ce qu'il faut faire. Dans un avion de chasse vous pourriez être mort en moins de temps qu'il n'en faut pour comprendre ce qui se passe. C'est la responsabilité du Scrum Master de s'assurer que l'équipe exécute immédiatement la procédure d'urgence Scrum, de préférence à la mi-sprint, quand l'avancement fait du hors-piste. Les étapes de la procédure d'urgence sont (faire seulement ce qui est nécessaire):

- 1. Changez la façon dont le travail est effectué. Faites quelque chose différemment*
- 2. Obtenez de l'aide, le plus souvent en déléguant un morceau du Carnet de Sprint à quelqu'un d'autre*
- 3. Réduisez le périmètre du Sprint (c'est-à-dire, enlevez des éléments du Carnet de Sprint - NdT)*
- 4. Abandonnez le Sprint et replanifiez-le*
- 5. Informez la direction de l'impact sur les dates de sortie*

Scrum ne signifie pas faire n'importe quoi pour atteindre l'Objectif du Sprint!

Scrumer le Scrum

Identifiez le principal obstacle lors de la rétrospective de Sprint et supprimez le avant la fin du prochain Sprint. Pour supprimer l'obstacle majeur, mettez le dans le carnet de Sprint avec des tests d'acceptation qui permettent de déterminer quand il est Terminé (Done). Ensuite, évaluez l'état de l'histoire lors de la revue de sprint comme tous les autres éléments.

Trop d'équipes de développement attendent "une permission" pour réaliser les améliorations continues de leur mode de travail. C'est tout simplement absurde.

Indicateur de bonheur

Le bonheur est l'un des meilleurs indicateurs car il est prédictif. Quand les gens pensent à leur niveau de bonheur, ils se projettent vraiment dans l'avenir et ce qu'ils ressentent. S'ils sentent que l'entreprise est en difficulté ou qu'ils font des choses inutiles, ils sont malheureux. De même s'il ya un obstacle majeur ou s'ils subissent des fonctionnements frustrants, ils sont malheureux.

Avant de rejeter ce Pattern comme un "doux rêve" ou une absurdité idéaliste, je vous mets au défi de l'essayer ! Commencez par lire l'article de Ben Linders sur InfoQ "How to Measure and Analyse Happiness" [Linders 2013].

Les équipes qui terminent tôt, accélèrent plus vite

*Les équipes prennent souvent trop de travail dans un Sprint et ne le finissent pas. L'échec empêche l'équipe de s'améliorer. Prenez donc moins d'éléments dans le Sprint. Maximisez vos chances de succès en utilisant le modèle de la **météo d'hier**. Ensuite, mettez en œuvre les quatre modèles pour réduire les obstacles au sein du Sprint, qui traiteront systématiquement toutes les interruptions et vous aideront à terminer tôt. Si vous finissez plus tôt, prenez le prochain élément du carnet de produit, ce qui améliorera la Météo d'hier pour les sprints à venir. Pour accroître la probabilité d'accélération, appliquez "**Scrumer le Scrum**" pour identifier votre kaizen durant la rétrospective de Sprint. Mettez le kaizen dans le carnet de Sprint avec des tests d'acceptation pour le prochain Sprint et assignez lui une priorité absolue.*

Plus sur les obstacles

Le Scrum Master a le devoir, parmi d'autres, de faire disparaître tout ce qui se trouve sur le chemin de l'équipe de développement et l'empêchent d'être à son plein potentiel. Il s'agit d'une quête sans fin. Les obstacles couvrent un large spectre allant de "mon ordinateur compile lentement" jusqu'à "notre PDG ne comprend rien à l'Agile".

Comme Scrum Master, si vous entendez des mots comme : *néanmoins, supposer, attendre, conjecture, penser, avec un peu de chance, heureusement, pas disponible, probablement, souhaiter, essayer*, vous avez probablement découvert un obstacle. De nombreuses équipes de développement ne se rendent même pas compte qu'elles sont ralenties. Elles considèrent souvent que "c'est la manière dont les choses fonctionnent ici". Votre travail dans cette situation est d'une part de les aider à prendre conscience de ces faiblesses (blind spot - NdT) et d'autre part de réaliser qu'ils sont habilités à procéder régulièrement à des transformations. J'enseigne ce vieil adage : "il est plus facile de demander pardon que de demander la permission".

Les équipes novices (se niveau *Shu*) ont tendance à croire qu'elles sont impuissantes à changer le statu quo. Ceci est souvent frustrant pour les managers qui ont appuyé sur le bouton magique Scrum et attendent que la magie de l'auto-organisation apparaisse. Cela reflète un manque de compréhension des dynamiques organisationnelles. Ces managers et leurs équipes bénéficieront du recrutement d'un coach agile expérimenté. La vérité crue est que l'auto-organisation requiert du temps (en combinaison avec un comportement de leadership approprié).

En allant plus loin, les équipes sont aveugles à leurs propres obstacles. Un dicton dit : "vous ne pouvez pas observer ce qui se passe dans la boîte si vous êtes à l'intérieur de la boîte". Le Scrum Master doit être le coach, le mentor de l'équipe et le leader-serviteur (servant leader), en les aidant à identifier leurs propres faiblesses et en les autonomisant et leur permettant de les surmonter un par un.

Etant donné que notre cerveau répond mieux aux stimuli positifs que négatifs, je préfère mettre l'accent sur les améliorations que l'équipe veut faire plutôt que sur les obstacles. La rétrospective de Sprint est l'outil de l'équipe pour faire émerger les éléments de blocage dans une démarche

d'amélioration continue. Je couple cette vision avec le pattern *Scrum le scrum* [Sutherland 2013] pour à la fois mettre l'accent sur l'amélioration continue et obtenir une rétroaction rapide qui nourrit les processus empiriques.

Cartographie des histoires (story mapping)

Jeff Patton [Patton 2008a] a développé son concept de cartographie des Histoires (Story Mapping) en partant des idées de Larry Constantine et Lucy Lockwood sur la conception centrée utilisation (Usage-Centred Design), sur les cas d'utilisation (Use Case) d'Alistair Cockburn et sur les Histoires Utilisateurs (User Stories) décrites par Kent Beck dans l'Extreme Programming. Une Cartographie des Histoires est une représentation en deux dimensions du Carnet de Produit (Product Backlog) avec les activités utilisateur sous la forme d'un flux narratif allant de gauche à droite et les versions successives du produit s'étendant verticalement. Les principaux éléments constitutifs d'un Cartographie d'histoire sont les tâches de l'utilisateur [Patton 2008b]. La figure ci-dessous montre un exemple d'une Cartographie d'Histoire. J'enseigne cet outil dans mes formations de Propriétaires du Produit (Product Owner) et l'utilise dans le coaching des équipes qui sont au-delà du niveau novice.



Copyright © 2004-2009 Marc Evers

Planification des livraisons avec une courbe d'avancement (burn up chart)

Le concept de vélocité

Nous observons le nombre de fonctionnalités réalisées et testées (en fait Done - NdT) qu'une équipe fournit au propriétaire du produit (Product Owner) à la fin de chaque Sprint. Nous appelons cela la vélocité. Nous disons que la vélocité d'une équipe est de "15 à 20 points" quand elle est en mesure de fournir à la fin de chaque Sprint un ensemble d'Histoires Terminées dont la taille estimée ne descend pas sous les 15 points et dépassent rarement 20 points.

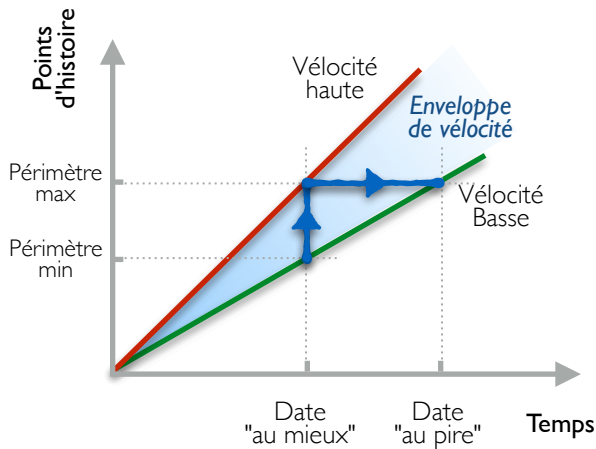
Cette plage de vélocité peut alors être utilisée (avec précaution) par le propriétaire du produit pour fournir à ses parties prenantes des limites supérieures et inférieures pour les futures livraisons de fonctionnalités.

Utilisation d'une enveloppe de vélocité pour la planification des livraisons

Le graphique d'avancement du produit ou de livraison ci-dessous montre la manière d'utiliser la plage de vélocité, construites avec les sprints précédents, pour fournir une enveloppe d'estimation crédible pour l'avenir. La ligne verte représente la limite inférieure de l'historique de vélocité de l'équipe à date. La ligne rouge indique la limite supérieure. La zone située entre ces lignes est la plage de vélocités probables, c'est à dire "l'enveloppe" de vélocité. En utilisant les données historiques, on peut estimer une date proche et une date tardive pour la livraison d'un périmètre spécifique, comme le montre la ligne bleue horizontale. Ou à l'inverse, nous pouvons estimer le périmètre livrable à une date précise, comme le montre la ligne bleue verticale. Dans la pratique, je préfère utiliser une date fixe avec un périmètre incertain. Dans ce cas, je livre une certaine valeur à mon client sur laquelle il peut compter. Et je peux en fournir plus, plus tard si nécessaire. Comparez cela avec le choix d'un périmètre donné : ce mode de fonctionnement empêche le client de choisir la valeur qu'il veut.



Nous devons toujours aider les parties prenantes à comprendre que le but est la transparence, afin qu'elles connaissent la réalité et puissent faire les choix

appropriés. Et nous ne devons jamais oublier le principe de la planification adaptative qui sous-tend Scrum et l'Agile.



Utilisation d'un Tableau d'avancement pour la planification des livraisons

- En théorie, plus la variance de la vélocité de l'équipe au cours des sprints successifs est faible, plus le Propriétaire du Produit peut prédire de manière fiable les futures livraisons. Cependant, il est dangereux d'encourager l'équipe à atteindre la perfection. Un résultat probable serait soit de jouer sur la taille (des éléments estimés - NdT), ce qui conduit à une perte de transparence, soit de passer un temps excessif à estimer, entraînant du gaspillage. Comme pour toute chose, il faut trouver le juste milieu. Une variation de $\pm 20\%$ autour de la moyenne est acceptable*
- Les tailles estimées ne sont généralement pas comparables entre les équipes. Cela n'a pas d'importance tant qu'il n'y a qu'une équipe travaillant sur le Carnet de Produit. C'est une des problématiques de la mise à l'échelle de Scrum, est assez facile à traiter, et sort du sujet de ce petit guide*

-  Avec de nombreuses équipes que j'ai coaché il s'avère que la vélocité n'augmente pas après les premiers sprints. J'ai trouvé cela plutôt surprenant jusqu'à ce que je réalise ce qui se passait. Avec la maturité et l'amélioration des pratiques, le travail devient plus facile et l'équipe diminue progressivement la valeur des estimations. La conséquence naturelle est l'augmentation de la valeur fournie par point d'histoire. Il importe peu que l'effort pour un point d'histoire change au fil du temps, à condition que ce changement soit progressif et soit complété par une sorte de moyenne mobile pour la planification. Je dis cela pour vous faire prendre conscience que les points d'histoire et la vélocité ne permettent pas de mesurer les intrants ou extrants sur le long terme ; ils ne sont qu'une aide à la planification
-  En outre, j'espère qu'un propriétaire du produit mature comprendra que ni les intrants ni les extrants ne sont des mesures utiles ; les équipes agiles devraient se concentrer sur la maximisation des résultats (la valeur - NdT) avec le moins de production possible (de code - NdT). Et il doit sensibiliser les parties prenantes à cela

Résumé du chapitre

- ◆ Présentation du modèle *Shu Ha Ri*
- ◆ Patterns pour améliorer la pratique de Scrum
- ◆ La nature des blocages
- ◆ Une introduction à la Cartographie des Histoires
- ◆ La Courbe d'Avancement pour la planification des livraisons

8. De l'aide

Ce guide commence par Scrum est simple . Faire du Scrum c'est difficile. Les formateurs et coachs agiles du monde entier utilisent la métaphore suivante : "Scrum est une lampe de poche qui permet de découvrir ce qui est caché dans les coins les plus sombres de votre organisation". Ils soutiendront que Scrum ne résoudra pas un seul problème à votre place.

Alors, que faire avec Scrum (ou toute autre méthode) ? Eh bien, vous pouvez lire des livres comme celui-ci et essayer d'appliquer ce que vous avez trouvé. Et j'espère que vous le ferez. Cependant, vous rencontrerez bientôt des difficultés que vous ne parviendrez pas à surmonter seul. Votre contexte ne se prête pas à ce que vous avez lu. Vos collègues, managers et même vous, êtes enclins à poursuivre dans vos habitudes, même si vous savez que cela ne marche pas. Changer est difficile.

Si j'ai appris une chose durant mon voyage dans l'Agilité, c'est à chercher de l'aide. Ce furent les actions les plus stimulantes pour moi. Et je tiens à vous encourager à faire de même. Pour citer Karl E. Weick et Kathleen M. Sutcliffe dans *Managing the Unexpected* : "C'est un signe de maturité et de confiance de savoir quand vous avez atteint vos limites et d'en savoir suffisamment pour trouver une aide extérieure" [Wieck et Sutcliffe , 2001].

Dans ce chapitre, mon but est de vous aider à reconnaître quand vous avez besoin d'aide, comment obtenir l'aide qui convient à vos besoins. Je suis conscient que c'est un vaste sujet et je ne peux que l'effleurer. Comme d'habitude je propose un ensemble de références que j'ai trouvé utiles.

La culture et l'éléphant

L'Agile n'est pas un mode d'action ; il s'agit plutôt d'un état d'esprit. Par conséquent les gens doivent changer leurs attitudes et comportements de façon plus ou moins importante. D'après mon expérience, c'est de loin la partie la plus difficile du voyage vers l'agilité.

«L'éléphant» qui se dresse sur le chemin de ces changements est la *culture* de l'organisation.

Edgar H Schein, expert en culture organisationnelle et en leadership, définit formellement la culture d'un groupe comme «un ensemble d'hypothèses fondamentales partagées, assimilées par un groupe car ayant résolu ses problèmes d'adaptation externe et d'intégration interne, qui ont suffisamment réussi pour être considéré comme valides et, par conséquent, est enseigné aux nouveaux membres comme la bonne façon de percevoir, de penser et d'appréhender les problèmes» [Schein 2010].

Comme la formule est un peu compliqué (sic), ces mots de Schein entreront peut-être plus facilement en résonance : «La clé pour saisir la "résistance au changement" est de comprendre que certains comportements qui pour nous sont devenus dysfonctionnels, peuvent néanmoins être difficiles à abandonner parce qu'ils servent d'autres fonctions positives».

Alors, que faire ? Mon propre parcours de coach s'est enrichi des échanges avec d'autres consultants ayant beaucoup plus d'expérience et d'expertise que moi, pour comprendre et apprécier que la culture est un moyen de réussir un changement organisationnel. Et, si vous vous en sentez les capacités, Schein propose une recette pour réaliser dans votre organisation un atelier rapide d'évaluation de la culture [Schein 2010, la page 315].

Qu'est-ce que le coaching?

Il est difficile de trouver une définition partagée du coaching. Une que j'apprécie est : «l'art de faciliter la réalisation, l'apprentissage et le développement d'un autre» [Downey , 2003].

Les caractéristiques suivantes sont communes à la plupart des approches:

- Il s'agit d'un dialogue structuré entre le coach et le coaché* , avec des attentes claires, des retours (feedback - NdT) et de la transparence ;
- Le coach facilite l'apprentissage et le développement du coaché ;
- Le coaching tend à améliorer le bien-être et la performance du coaché.

**Certaines personnes ont une aversion pour le terme coaché. Cependant, je n'ai pas trouvé de meilleur mot générique. Notez également que le coaché peut être un individu ou un groupe.*

Une autre dimension du coaching dans notre monde, est que le coach Agile peut endosser des rôles de professeur, conseiller-expert et mentor. Les deux premiers sont généralement basés sur l'expertise dans le cadre de processus agiles (comme Scrum). Le troisième est basé sur l'expérience personnelle du coach, par exemple dans le rôle de Scrum Master ou Propriétaire de Produit (Product Owner). Un bon coach Agile explicitera aux coachés les moments où il prend l'un de ces rôles.

Comment puis-je savoir quand j'ai besoin d'aide?

Nous avons tous besoin d'aide. Demandez à n'importe quel leader ayant réussi, et il vous parlera des gens qui l'ont aidé formellement et informellement durant son voyage.

Alors une meilleure question est peut-être quel type d'aide ai-je besoin suivant les contextes ?

Le rôle de Scrum Master est d'assurer la fluidité du processus pour le reste de l'équipe Scrum ou de l'organisation. La formation en classe, telle que la Certification de Scrum Master (Certified Scrum Master), est insuffisante pour acquérir les connaissances tacites nécessaires pour maîtriser les pratiques agiles et le cadre Scrum. L'équipe et l'organisation doivent acquérir les compétences d'un praticien expérimenté pour les guider dans le dédale des défis auxquels ils devront faire face pendant une transition Agile. Un coach Agile sera probablement en mesure d'aider ici.

Quand il s'agit de la transformation organisationnelle, nous avons besoin de l'aide d'un coach ayant des compétences plus formelles. Étant donné que l'Agile (et son proche cousin le Lean) se concentre sur l'ensemble du système, il s'ensuit qu'un coach avec une vision systémique sera plus pertinent. Pour une meilleure compréhension, je recommande l'article *Agile and Systemic Coaching* de Sigi Kaltenecker et Bent Myllerup [Kaltenecker et Myllerup 2011].

Un autre truisme est que nous avons des faiblesses dont nous sommes inconscients ainsi que des zones d'ombres (blind spots - NdT). Ceci n'a rien à voir avec de la bêtise ou de l'incompétence, mais provient de notre nature humaine. Un tiers de confiance, externe à notre environnement

immédiat - par exemple un coach systémique - est mieux placé pour observer et nous informer sur nos propres zones d'ombre.

Au cours de ses recherches sur l'adoption réussie de l'Agilité, Marcus de Beer, coach Agile et DSI, a observé que les meilleures organisations associent des coachs internes et externes [De Beer 2012].

Quels sont les aspects économiques du coaching?

Engager un coach n'est qu'une composante dans l'adoption de l'Agile. Il y a des coûts associés à la formation des équipes aux méthodes agiles, à l'affectation de temps pour pratiquer, à la restructuration de l'organisation, à l'accompagnement des managers pour s'orienter vers de nouveaux rôles et bien plus encore.

Mes observations, basées sur des douzaines d'équipes dans diverses organisations, est que les avantages d'une transition Agile surpassent les coûts à chaque fois. En aucun cas, les personnes concernées n'ont cherché à revenir à leur ancien mode de fonctionnement ; en fait leur exclamation est «plutôt partir que faire marche arrière».

Néanmoins, une énigme subsiste : pourquoi certaines organisations montrent-elles des améliorations spectaculaires et continues, tandis que d'autres ont tendance à stagner après une petite avancée ? J'ai vu certaines équipes tripler leur retour sur investissement en quelques sprints et j'en ai vu d'autres satisfaites avec une petite amélioration. Je crois que la réponse réside dans les aspects culturels dont j'ai parlé au début de ce chapitre.

Donc, je crois que le coût du coaching n'est pas la bonne question à poser. Je préfère demander : "Comment pouvons-nous améliorer X?", où X peut être la satisfaction du client, le bonheur des employés, le ROI, etc. Et après «Quels investissements sont nécessaires et comment peut-on mesurer la valeur?» Votre coach devrait être prêt et capable de collaborer avec vous pour comprendre vos besoins et répondre à ces questions.

Lorsqu'on demande aux managers qui ont réalisé une transition vers l'Agile et Scrum, ce qu'ils auraient fait différemment, la réponse la plus fréquente à travers le monde est «plus de coaching» !

Résumé du chapitre

- ◆ Prendre en compte l'importance de la culture organisationnelle
- ◆ Qu'est ce que le coaching ?
- ◆ Savoir quand on a besoin d'aide
- ◆ Aspects économiques du coaching

9. Un peu plus sur l'Agile et le Lean

Le Manifeste Agile

En Février 2001, dix-sept spécialistes indépendants et penseurs des «méthodes agiles» dans le monde du logiciel se sont réunis pour échanger et trouver des dénominateurs communs. Parmi eux se trouvaient, les fondateurs de Scrum, Jeff Sutherland et Ken Schwaber, ainsi que Mike Beedle, qui ont travaillé sur les modèles initiaux de Scrum et co-écrit le premier livre sur le sujet. Le groupe s'est baptisé "Agile Alliance" (Alliance Agile) et s'est accordé sur un Manifeste pour le Développement logiciel Agile (Agile Manifesto - NdT). Ils définissent en outre douze principes sous-jacents au manifeste, reproduit dans son intégralité sur la page ci-dessous [Highsmith, 2001].

Commentaire

Le Manifeste Agile et ses douze principes sont toujours là après plus d'une décennie. Ces deux aspects sont le critère décisif pour évaluer toutes les méthodes agiles et les praticiens de l'Agile. La critique la plus forte reste l'orientation vers le développement de logiciels, alors que les méthodes Agiles sont en fait plus largement applicables.

Une idée fausse communément véhiculée est que l'Agile vise à éliminer tous les processus, la documentation, les contrats et la planification. La lecture du Manifeste devrait suffire à prouver que tel n'en est pas l'objet.

Pour en savoir plus sur l'histoire du Manifeste, et avoir une analyse des quatre valeurs et douze principes, je vous suggère de lire l'article fondateur de Martin Fowler et Jim Highsmith dans le Dr. Dobbs Journal [Fowler et Highsmith, 2001].

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Principles behind the Agile Manifesto

We follow these principles:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Développement de produits Agile et Lean

Scrum est un exemple de méthode Agile. L'adhésion au Manifeste Agile et à tous ses principes ne signifie pas nécessairement qu'une équipe ou une organisation utilise Scrum. Ken Schwaber et Jeff Sutherland le disent clairement : «Les rôles, les objets, les événements et les règles de Scrum sont immuables et si la mise en oeuvre partielle de pratiques est possible, le résultat n'est pas Scrum. Scrum n'existe que dans son intégralité et sert de conteneur pour d'autres techniques, méthodes et pratiques.» [Schwaber 2013]. Cela peut sembler compliqué pour le novice. Pour ceux qui cheminent avec Scrum et sont capable d'une introspection sur la structure Scrum, les raisons en sont plus évidentes. Scrum fournit un modèle minimal pour établir l'ordre.

Les développements Agile et Lean de produit (par opposition à la production Lean) sont de proches cousins. Les deux se concentrent sur la fourniture d'un flux de valeur aux clients et tous deux font du respect des individus une valeur fondamentale. L'Agile s'appuie sur une planification adaptative avec des boucles de rétroaction rapides, tandis que le Lean se concentre sur la mise en oeuvre du flux. Tous deux partagent l'idée d'un apprentissage rapide et de l'amélioration continue.

Autres méthodes Agile et Lean

La méthode «légère» la plus populaire au moment de la rédaction du Manifeste Agile était l'Extreme Programming (XP). Une distinction fondamentale entre Scrum et XP est que le premier se concentre exclusivement sur un modèle de gestion du projet alors que le second met l'accent sur les pratiques et techniques de développement de logiciels. En fait, ils partagent beaucoup et chaque bonne équipe Scrum que j'ai observé utilise bon nombre des principes et des pratiques XP. Pour une description complète de XP je recommande le livre *Extreme Programming Explained* de Kent Beck [Beck 2005].

Une autre méthode apparue plus récemment est appelé la méthode Kanban. Ce mouvement est initié par David Anderson et d'autres. Elle est sans doute à classer dans le Lean que dans l'Agile, mais cette distinction est sans doute ésotérique pour la plupart des gens.

D'après mon expérience, les deux méthodes présentent de solides avantages et permettent une meilleure organisation du travail et de la connaissance. Pourtant, les deux sont soumises aux gémonies à cause de mauvaises implémentations. Encore un cas de "Tirer sur le messager" ?

Kanban et Scrum ne sont pas mutuellement exclusives et des équipes expérimentées sauront mélanger les idées des deux modèles. Comme l'objet de ce livre est Scrum, il n'est pas question de rentrer dans les détails, mais le tableau suivant met en évidence quelques similitudes et différences.

Kanban	Scrum
Gère un flux de travail en le visualisant, ce qui limite la quantité de travail en cours, et supprime les goulets d'étranglement	Gère la conception itérative et la livraison incrémentale de la valeur à partir d'une liste dynamique de travail qui est alimenté par une évaluation continue des besoins des utilisateurs
Pas de rôles, quatre principes, six pratiques	Rôles, événements et objets définis
Perçue comme simple ; en réalité très sophistiqué	Simple à apprendre ; difficile à bien faire
Le changement progressif et continu	(Plus) changement révolutionnaire
Orienté pour traiter les problèmes compliqués*	Orienté pour traiter les problèmes complexes*

- La méthode Kanban gère un flux de travail donné en le visualisant, ce qui limite la quantité de travail en cours, et supprime les goulets d'étranglement
- Kanban impose peu de règles. Cela crée une impression de simplicité. En pratique, il s'agit d'une méthode sophistiquée et nécessite de l'accompagnement
- Les changements avec Kanban sont évolutifs. Cela peut rendre le choix de Kanban plus acceptable pour les organisations dont la culture ne permet pas d'introduire des changements rapides

- Scrum, en revanche, attaque la complexité de front et nécessite un changement plus radical dans l'attitude. Scrum provoque un changement fondamental. Cela est difficile pour les organisations et les personnes, mais les résultats peuvent être stupéfiants
- Il serait judicieux de vous familiariser avec les deux méthodes, ou d'obtenir des conseils d'experts, pour faire un choix éclairé. Vous pourriez aussi utiliser ou hybrider les deux suivant les domaines d'application
- Arne Roock a écrit un guide excellent et accessible sur Kanban [Roock 2012]. Les plus éclairés devraient lire l'ouvrage complet de David Anderson sur le sujet [Anderson 2010]

*Je considère que la méthode Kanban est probablement le meilleur outil pour résoudre des problèmes dans le domaine du compliqué. Scrum, en revanche, est parfaitement adapté à la résolution de problèmes complexes. Certains lecteurs trouveront que cette déclaration porte à controverse. Vous pouvez trouver une référence pour la compréhension de la complexité dans l'article de Snowden et Boone dans la HBR [Snowden 2007].

Résumé du chapitre

- ◆ Qu'est ce que le Manifeste Agile ?
- ◆ Quelle relation entre Agile et le développement de produit lean
- ◆ Une introduction à d'autres méthodes: XP et Kanban

Remerciements

La genèse de la première version de ce petit livre a été la nécessité, en tant que sponsor, de fournir un petit plus dans la pochette cadeau du premier jour du Scrum Days Afrique du Sud en 2009. Depuis, un certain nombre de personnes ont eu la gentillesse de le recommander et aussi de suggérer des améliorations. Les praticiens agiles que j'ai eu la chance de rencontrer et avec qui j'ai interagis, ont dans une certaine mesure influencé ce que vous trouvez écrit ici.

Au risque d'en oublier, je voudrais mentionner ceux qui ont eu une influence significative durant mon voyage agile : Boris Gloger, mon premier formateur / coach Scrum et mentor ; Jim Cundiff, qui en tant que PDG de la Scrum Alliance m'a encouragé et aidé à fonder le Groupe d'utilisateurs Scrum Afrique du Sud (SUGSA) ; les nombreuses personnes qui ont généreusement donné de leur temps pour servir SUGSA au cours de mon mandat : Neil, Sue, Steve, Mike, Carlo, Charl, Evan, Karen G, Karen V, Kevin, Neill et Alwyn ; Tobias Mayer, qui m'a encouragé et a promu cette brochure ; Pete Behrens et Roger Brown, qui a créé le programme du SCC ; Andreas Schliep et Peter Beck, qui ont collaboré fréquemment dans la formation et le coaching ; Mike Freislich, mon premier partenaire à ScrumSense ; Carlo Kruger et Marius de Beer, mes partenaires dans le business pendant deux ans et avec qui j'ai beaucoup appris ; David Anderson, qui m'a présenté Kanban ; Karen Greaves, avec qui nous avons fait plus de cession de formations que je ne peux m'en rappeler ; et Sigi Kaltenecker, qui m'a encadré dans les pratiques de développement organisationnel au cours des quatre dernières années.

En outre, j'ai appris beaucoup avec les autres CST (Formateur Certifié Scrum) durant ma formation, avec les représentants Lean / Agile durant les conférences où nous avons eu des discussions poussées, avec les participants à mes formations et les clients avec qui j'ai travaillé, que j'ai formés et encadrés durant les huit dernières années. Vous êtes trop nombreux pour que je vous cite tous.

Et je tiens à remercier Ken Schwaber, Jeff Sutherland et les autres pionniers Scrum qui nous ont ouvert la voie.

Mes sincères remerciements vont à tous les bénévoles qui ont généreusement pris le temps pour traduire la deuxième version en plusieurs langues :

- ◆ Alan Cyment (espagnol—Un Mejor Scrum)
- ◆ Peter Beck, Sebastian Lauber, Andreas Schliep et Henning Wolf (allemand—Scrum besser machen)
- ◆ Samuel Gonsales (portugais brésilien—Melhor Scrum)
- ◆ Stéphane Wojewoda Matthieu NGuyen et Yann Le Moal (français—Du Meilleur Scrum)

Vous trouverez des liens vers la copie électronique de cette brochure et toutes les traductions disponibles sur www.scrumsense.com/dobetterscrums.

N'hésitez pas à envoyer des propositions de traduction à info@scrumsense.com.

J'ai essayé de citer et donner les sources quand j'ai emprunté le travail d'autre. Si je vous ai oublié, n'hésitez pas à me le dire pour que je corrige cela.

Références

- Adkins, Lyssa (2010). *Coaching Agile Teams: A Companion for ScrumMasters, Agile Coaches and Project Managers in Transition*. Addison-Wesley Professional.
- Ambler, Scott M. and Lines, Mark (2012). *Disciplined Agile Delivery: A Practitioners' Guide to Agile Software Delivery in the Enterprise*. IBM Press.
- Anderson, David J. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press.
- Argyris, Chris (1977). *Double Loop Learning in Organizations*.
<http://hbr.org/1977/09/double-loop-learning-in-organizations/ar/1>
- Beck, Kent with Cynthia Andres (2005). *Extreme Programming Explained (second edition)*. Addison-Wesley Professional.
- Beaumont, Serge (2009). *The Definition of Ready*.
<http://blog.xebia.com/2009/06/19/the-definition-of-ready/>
- Cockburn, Alistair (2007). *Agile Software Development: The Cooperative Game (2nd edition)*. Addison-Wesley Professional.
- Cohn, Mike (2004). *User Stories Applied*. Addison-Wesley Professional.
- Cohn, Mike (2006). *Agile Estimating and Planning*. Prentice Hall.
- Cohn, Mike (2009). *Succeeding with Agile: Software Development Using Scrum*. Addison-Wesley Professional.
- Conway, Melvin (1968). *Conway's Law*.
http://www.melconway.com/Home/Conways_Law.html
- Coplien, James O. (1994). *Borland Software Craftsmanship: A New Look at Process, Quality and Productivity*. <https://sites.google.com/a/gertrudandcope.com/info/Publications/Patterns/Process/QPW>.
- De Beer, Marius (2012). *Data-driven Agility: An analysis of Agile adoption in North American Organisations*. <http://www.scrumsense.com/downloads/data-driven-agility.pdf>.
- De Beer, Marius (2013). Personal correspondence and conversations during 2012 & 2013.
- Drucker Peter F. (2001). *Management Challenges for the 21st Century*. HarperBusiness.
- Fowler, Martin and Highsmith, Jim (2001). *The Agile Manifesto*.
<http://www.drdoobs.com/open-source/the-agile-manifesto/184414755>.
- Fowler, Martin (2005). *The New Methodology*.
<http://www.martinfowler.com/articles/newMethodology.html>.
- Fox, Ron (1995). *Shu Ha Ri*. In THE IAIDO NEWSLETTER Volume 7 number 2 #54 FEB 1995. <http://www.aikidofaq.com/essays/tin/shuhari.html>.

- Gartner (2012). *The End of the Waterfall as We Know It*.
<http://www.gartner.com/id=2127715>. (Not available for individual purchase.)
- Gloger, Boris (2008). *Heartbeat Retrospectives*.
<http://borisgloger.com/2008/04/24/heart-beat-retrospectives-1-introduction/>.
- Goldratt, Eliyahu M. (2006). *The Haystack Syndrome: Sifting Information Out of the Data Ocean*. North Riven Press.
- Greenleaf, Robert K. (2012). *The Servant as Leader*. The Greenleaf Center for Servant Leadership.
- Highsmith, Jim, et al (2001). *Manifesto for Agile Software Development*.
<http://www.agilemanifesto.org/>.
- Hundermark, Peter (2009). *Measuring for Results*.
<http://www.scrumsense.com/coaching/measuring-for-results>.
- Hundermark, Peter (2014). *Scaling Scrum to the Organisation*.
<http://www.scrumsense.com/blog/scaling-scrum-organisation>.
- James, Michael (2006). *An Example Checklist for ScrumMasters*.
<http://www.scrummasterchecklist.org/>.
- Jeffries, Ron (2001). *Card, Conversation, Confirmation*.
<http://www.xprogramming.com/xpmag/expCardConversationConfirmation>.
- Jeffries, Ron, et al (2013). *Agile Atlas*. <http://agileatlas.org>.
- Kaltenecker, Siegfried and Myllerup, Bent (2011). *Agile and Systemic Coaching*.
<http://www.scrumalliance.org/community/articles/2011/may/agile-systemic-coaching>.
- Katzenbach, Jon R. and Smith, Douglas K. (2002). *The Wisdom of Teams*. Collins.
- Kniberg, Henrik (2007). *Scrum and XP from the Trenches*.
<http://www.infoq.com/minibooks/scrum-xp-from-the-trenches>.
- Kniberg, Henrik (2008). *Multi-Team Sprint Planning*. http://www.scrumalliance.org/system/resource_files/0000/0871/Multi-Team-Sprint-Planning.pdf.
- Kniberg, Henrik and Ivarsson, Anders (2012). *Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds*.
<http://blog.crisp.se/2012/11/14/henrikkniberg/scaling-agile-at-spotify>.
- Larman, Craig and Vodde, Bas (2008). *Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum*. Addison-Wesley Professional.
- Larman, Craig and Vodde, Bas (2010). *Practices for Scaling Lean & Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum*. Addison-Wesley Professional.
- Larman, Craig (2013). *Larman's Laws of Organizational Behavior*.
http://www.craiglarman.com/wiki/index.php?title=Larman's_Laws_of_Organizational_Behavior
- Leffingwell, Dean (2013). *Scaled Agile Framework*.
<http://scaledagileframework.com/>.

- Linders, Ben (2013). *How to Measure and Analyse Happiness*.
<http://www.infoq.com/news/2013/01/measure-analyze-happiness>
- Mayer, Tobias (2009). *Scrum Roles—an abstraction*.
<http://agileanarchy.wordpress.com/2009/10/08/scrum-roles-an-abstraction/>.
- Nonaka, Ikujiro and Takeuchi, Hirotaka (1986). *The New, New Product Development Game*. Harvard Business Review, Jan-Feb 1986.
- Patton, Jeff (2008a). *The new user story backlog is a map*.
http://www.agileproductdesign.com/blog/the_new_backlog.html
- Patton, Jeff (2008b). *User Story Mapping*. http://www.agileproductdesign.com/presentations/user_story_mapping/index.html
- Pichler, Roman (2010). *Agile Product Management with Scrum: Creating Products that Customers Love*. Addison-Wesley Professional.
- Pink, Daniel (2009). *The Puzzle of Motivation*.
http://www.ted.com/talks/dan_pink_on_motivation.html.
- Pink, Daniel (2011). *Drive: The Surprising Truth About What Motivates Us*. Riverhead Books.
- Reinertsen, Donald G. (2009). *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing.
- Roock, Arne (2012). *Stop Starting, Start Finishing!*
<http://www.leankanbanuniversity.com/justin>.
- Rubin, Kenneth S. (2012). *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley Professional.
- Sliger, Michele and Broderick, Stacia (2008). *The Software Project Manager's Bridge to Agility*. Addison-Wesley Professional.
- Schein, Edgar H. (2010). *Organisational Culture and Leadership, Fourth Edition*. Jossey-Bass.
- Schwaber, Ken and Beedle, Mike (2001). *Agile Software Development with Scrum*. Prentice Hall.
- Schwaber, Ken (2007). *The Enterprise and Scrum*. Microsoft Press.
- Schwaber, Ken and Sutherland, Jeff (2013). *Scrum Guide*.
<http://www.scrum.org/Scrum-Guides>.
- Snowden, David J. and Boone, Mary E. (2007). *A Leader's Framework for Decision Making*. Harvard Business Review, Nov 2007.
- Sutherland, Jeff (2003–2012). *Scrum Log Jeff Sutherland > Productivity*.
<http://scrum.jeffsutherland.com/search?q=productivity>.
- Sutherland, Jeff (2007). *Origins of Scrum*.
<http://scrum.jeffsutherland.com/2007/07/origins-of-scrum.html>.
- Sutherland, Jeff (2009). *Sprint Burndown: by hours or by story points?*
<http://scrum.jeffsutherland.com/2009/04/Sprint-burndown-by-hours-or-by-story.html>.

- Sutherland, Jeff (2011). *Scrum: Inside the Sprint Retrospective*.
<http://youtu.be/MFLvQXMNrO8>.
- Sutherland, Jeff (2013). *Finish Early, Accelerate Faster*. http://scrumlab.scruminc.com/articles.html/_/open/finish-early-accelerate-faster-r56.
- Tabaka, Jean (2006). *Collaboration Explained | Facilitation Skills for Software Project Leaders*. Addison-Wesley Professional.
- Teasley, Covi, Krishnan, & Olson (2000). *How Does Radical Collocation Help a Team Succeed?* Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work (pp. 339 - 346). New York: ACM.
- VersionOne (2013): *The State of Agile Development | 7th Annual Survey*. <http://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf>
- Wake, William (2003). *INVEST in Good Stories, and SMART Tasks*.
<http://xp123.com/xplor/xp0308/index.shtml>.
- Weick, Karl E. and Sutcliffe, Kathleen M. (2001). *Managing the Unexpected*. Jossey-Bass.
- Yip, Jason (2006). *It's Not Just Standing Up: Patterns of Daily Stand-up Meetings*.
<http://martinfowler.com/articles/itsNotJustStandingUp.html>.

@peterhundermark
scrumsense.com
Cape Town • Johannesburg

