

SCRUM, LES CLES POUR DEMARRER AVEC L'AGILITE

G. Kurzawa*, G. Farges

Master Qualité et Performance dans les Organisations
Université de Technologie de Compiègne - CS 60319, 60203 Compiègne Cedex France, www.utc.fr/master-qualite

*correspondant : gaetankurzawa@gmail.com

RESUME :

« L'Agilité » est apparue il y a près de 20 ans pour les projets complexes notamment ceux liés aux Technologies de l'Information (IT).

Au vu des succès qu'elle rencontre dans les projets par rapport aux méthodes dites classiques, l'agilité et ses pratiques sont attractives et nombreux sont ceux à les expérimenter dans leurs projets.

Scrum (mêlée en français) est le cadre méthodologique agile le plus utilisé aujourd'hui (par plus de 58% des utilisateurs de l'Agilité), c'est également celui qui est le plus éprouvé et ayant le plus de retour d'expériences. Il permet à ses utilisateurs de gagner en transparence, inspection et adaptation.

Après avoir présenté la complexité des environnements, les approches classique et agile sont abordées et comparées.

L'approche agile y est ensuite présentée en partant de la théorie jusqu'à une pratique via le cadre méthodologique Scrum. Cette partie permet au lecteur d'avoir une vision et une application concrète d'une méthode Agile.

Mots-clefs : Agilité, Scrum, pratiques agiles, projets complexes

ABSTRACT:

Scrum, Keys to start with Agility

20 years ago, appears Agility a way to handle complex projects and especially Information Technology (IT) projects.

In the light of the success of Agility in project compared to traditional approaches, Agility and its practices are attractive and many experience these methods in their projects.

Scrum (mêlée in French) is the most used agile framework (by more than 58% of Agility users). Nowadays, this framework is the most proven and have the most feedbacks. It allows users to gain transparency, inspection and adaptation

After having shown complexity of environments, classical and agile approach are discussed and compared.

Then, agile approach is shown going from the theory to an agile practice by the Scrum framework. This part will allow the reader to have a vision and a concrete application of Agility methods.

Keywords: Agility, Scrum, agility practices, complex projects

I LA GESTION DE PROJET UNE CLE DE LA REUSSITE

I.1 ENJEUX DE LA GESTION DE PROJET

Projet : processus unique qui consiste en un ensemble d'activités coordonnées et maîtrisées comportant des dates de début et de fin, entrepris dans le but d'atteindre un objectif conforme à des exigences spécifiques, incluant les contraintes de délais, de coûts et de ressources [1].

D'après cette définition le but du projet est d'atteindre un objectif final qui est la production d'un livrable. Ce livrable peut être sous forme d'un produit (voiture, téléphone...) ou ouvrage (logiciel, route...).

Le projet fait intervenir des ressources pour atteindre le but et est encadré par des contraintes. Tout l'enjeu d'un projet et de sa gestion est d'arriver à atteindre l'objectif en respectant les contraintes.

La gestion de projet consiste à planifier le projet, suivre le budget, maîtriser les risques, respecter la qualité demandée. Elle est standardisée par la norme internationale ISO 21500 « Management de projet » [2].

Selon le **Chaos Report de 2015** seulement 29% des sondés déclaraient que leur projet était un succès. 52% ont déclaré que leur projet a été challengé, c'est-à-dire que l'une des contraintes du projet n'a pas été respectée. Les 19% restants représentent les projets qui n'ont jamais abouti [3]. Les raisons de ces échecs sont souvent une mauvaise gestion de projets, des contraintes qui sont inadaptées ou encore une approche non adéquate.

I.2 DES PROJETS QUI S'INSCRIVENT DANS DES ENVIRONNEMENTS PLUS OU MOINS COMPLEXES

La matrice de Stacey (figure 1) définit différents types d'environnements en fonction du niveau d'entente des besoins et de la certitude sur la technologie [4].

Si les besoins sont clairs (près de l'entente) et que la technologie est maîtrisée (près de la certitude), l'environnement est simple. En s'éloignant l'environnement est d'abord compliqué puis complexe pour atteindre finalement le chaos.

Compliqué est défini comme étant un système prévisible, ce système peut être découpé en sous-ensembles indépendants. Un projet compliqué peut par exemple être la production d'un avion. Le système en lui-même est d'une grande ampleur mais chaque sous système peut être découpé en petits éléments indépendants qu'il faudra assembler par la suite.

En environnement complexe le système est imprévisible et ne peut pas être découpé en sous-éléments indépendants à cause de nombreuses interactions entre les sous-systèmes. Dans ce type d'environnement le besoin n'est pas forcément bien défini et les moyens pour arriver au but peuvent être nombreux, c'est le cas par exemple des projets de développement informatique.

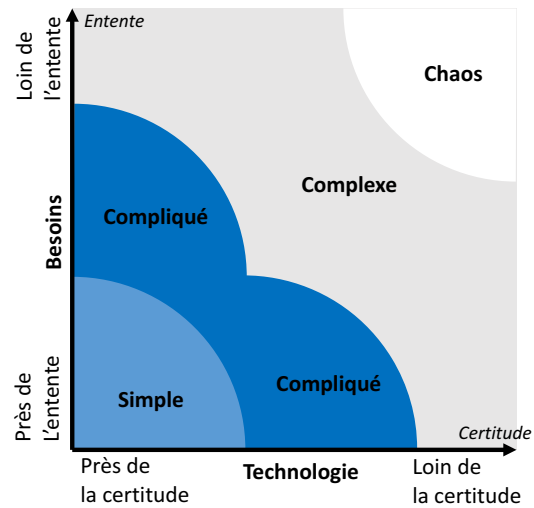


Figure 1 : Matrice de Stacey, des environnements plus ou moins complexes (adaptée de [4])

La manière de **gérer la complexité** va donc dépendre de l'environnement dans lequel le projet se situe. Les approches dites classiques (Waterfall [5] ou Cycle en V [6]) sont plus adaptées à des environnements simples et compliqués.

L'approche agile sera elle plus adaptée pour des environnements compliqués et complexes. Il faut cependant bien prendre en compte le contexte et l'environnement pour adapter l'approche.

I.3 LA GESTION DE PROJET PAR L'APPROCHE CLASSIQUE (OU WATERFALL)

Dans ce type d'approche le besoin et la vision du produit final sont clairement définis.

Les étapes sont faites en cascade, les unes à la suite des autres : tant qu'une étape n'est pas terminée, la suivante ne peut pas être commencée (figure 2).

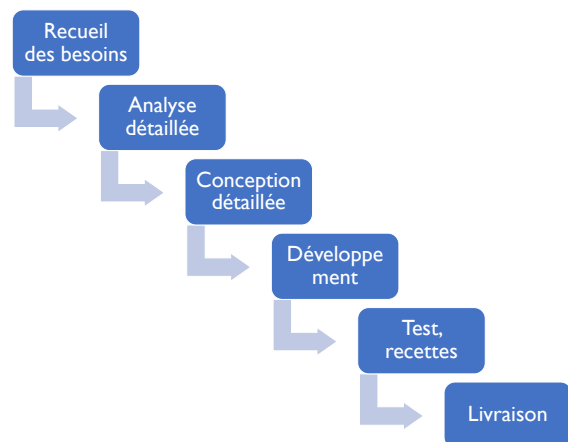


Figure 2 : Approche classique ou Waterfall [d'après [7]]

Le système est dépendant des étapes précédentes et si une erreur ou un problème est détecté en fin de processus, la correction doit repasser le processus au cours d'un nouveau cycle. En outre « l'industrialisation » de ces différentes étapes est souvent utilisée. Il s'agit de cloisonner chaque étape pour optimiser le processus et être plus efficace [8]. A minima l'industrialisation peut se faire au sein d'une équipe (chaque personne est dédiée à une tâche) mais elle peut se faire également via de la sous-traitance.

Comme mentionné dans la partie précédente cette approche est souvent adaptée pour les **environnements simples**. Ce sont ceux où le besoin, facilement définissable et spécifiable, n'évolue pas au cours du projet.

L'approche classique rencontre souvent un **effet tunnel** : l'idée de départ est claire mais le produit final est souvent différent de ce qui est attendu. Cela est dû au **cloisonnement** et au **manque de visibilité** entre les étapes.

En outre, le besoin évolue souvent au cours du projet contrairement au postulat de départ.

Tous ces éléments induisent qu'avec l'approche classique 29% des projets sont arrêtés en cours de route et 45% des fonctionnalités ne sont pas utilisées ou encore que 60% aboutissent mais après dépassement des délais et des budgets [3].

2 FAIRE DU CLASSIQUE EN MIEUX : L'APPROCHE AGILE

L'approche agile est apparue il y a 20 ans et a été définie en 2001 dans le « **Manifeste pour le développement Agile de logiciels** » [9]. Ce texte a été écrit sous la gouvernance de 17 experts du développement d'applications informatiques tels Ken Schwaber et Jeff Sutherland, fondateurs de la méthode Scrum (1996) qui est à ce jour la méthode agile la plus connue et utilisée [10].

L'agilité est née de la volonté d'être en rupture avec l'approche classique qui n'est pas la mieux adaptée notamment pour certains projets complexes comme les projets sur les Technologies de l'Information (IT).

La volonté est de rompre avec « l'industrialisation » et notamment le cloisonnement entre les différentes phases et le manque de visibilité qu'il entraîne.

En outre, à force de tout industrialiser, la performance, l'innovation et la motivation des acteurs du projet pourrait tendre à diminuer [11]. C'est ce que l'approche agile cherche à retrouver.

L'Agilité est une façon de penser (un « mindset ») et repose sur 4 valeurs et 12 principes [9] (figure 3).

L'Agilité est utilisée pour le développement de logiciels et la conduite de projet.

Un grand nombre de pratiques dites « agiles » participent de ces valeurs et principes.

Parmi elles : **Rapid Application Development** (RAD, 1991) [12], **Scrum** (1990) [13] et **Extreme Programming** (XP, 1999) [14] sont les plus connues

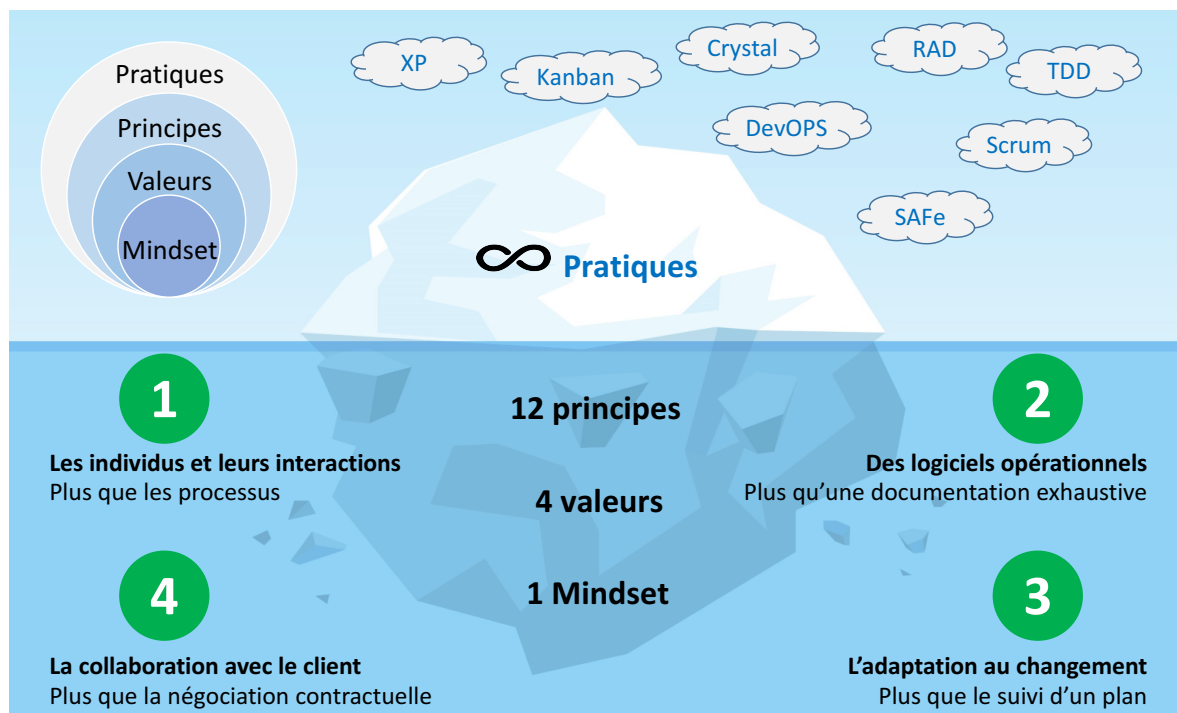


Figure 3 : L'approche Agile est constituée par 1 Mindset, 4 Valeurs, 12 Principes, une infinité de Pratiques (d'après [9])

3 LES APPORTS DE L'AGILITE PAR RAPPORT A L'APPROCHE CLASSIQUE

3.1 CLASSIQUE VS AGILE

Les contraintes d'un projet sont souvent représentées par l'équilibre Qualité, Coût, Délai (QCD), avec les fonctionnalités au centre.

Un projet classique est centré sur le suivi d'un « Plan » (figure 4), c'est le cahier des charges et les spécifications du produit qui vont guider l'ensemble du projet. C'est pour cette raison que les fonctionnalités sont figées. Les délais et le coût sont ensuite adaptés pour produire les fonctionnalités attendues. La qualité est souvent fixée dans un premier temps, puis ajustée pour s'adapter aux contraintes de coûts et de durée.

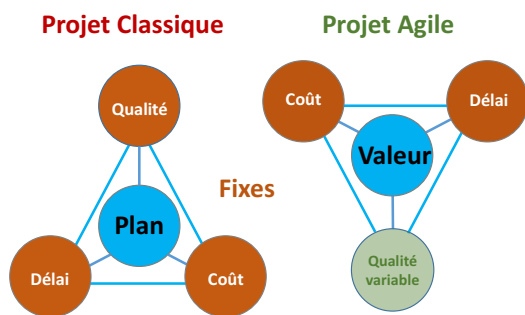


Figure 4 : Projets Classique / Agile (d'après [3])

A l'inverse, un projet agile est centré sur la « Valeur » (figure 4). La priorité est donnée à ce qui est important pour le client et à l'adaptation permanente du projet à son besoin. Par conséquent, vu que le besoin change généralement avec le temps, les fonctionnalités évoluent de concert.

Dans une **approche classique**, le coût, les délais et la qualité sont généralement fixés dès le départ et pris comme des objectifs mesurables à atteindre, générant d'emblée des contraintes dans la dynamique du projet. En **approche « agile »**, les tests se font dès le début du projet et à chaque sprint ce qui permet d'avoir un produit de qualité. Le délai est fixe, le client prévoit une date de livraison pour le produit, c'est le contenu qui sera adapté et non la date.

3.2 LES BÉNÉFICES DE L'AGILITÉ SELON SES UTILISATEURS

Selon le 10^{ème} rapport annuel sur l'état de l'Agilité de VersionOne [10], les bénéfices de l'Agilité sont (figure 5) :

- L'adaptation au changement
- La productivité de l'équipe
- La visibilité sur le projet
- L'augmentation du moral de l'équipe
- Une meilleure prédiction de la livraison du produit

Les projets agiles ne rencontrent actuellement que 9% d'échecs et environ 39% de réussites [3]. Les causes de ces échecs sont principalement dues à une culture d'entreprise qui n'est pas adaptée à l'Agilité, un manque d'expérience des pratiques ou une faible mobilisation du personnel de management.

L'Agilité est particulièrement adaptée à des **projets complexes**, là où le besoin n'est pas clairement défini et où l'adaptation et la visibilité sont importantes pour réussir le projet.

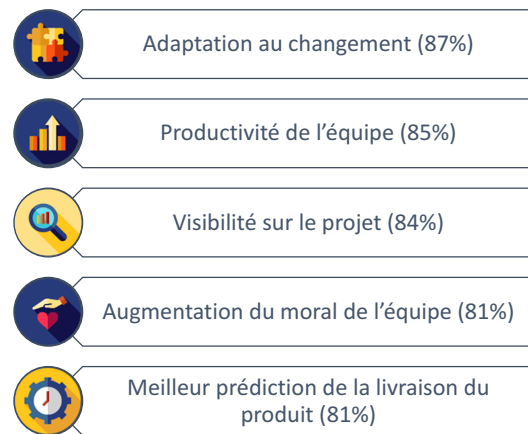


Figure 5 : Les bénéfices de l'Agilité selon ses utilisateurs.

Pourcentages des sondés déclarant une amélioration (d'après [4])

4 UNE PRATIQUE DE L'AGILITE : LE CADRE METHODOLOGIQUE SCRUM

Scrum (mêlée) : méthodologie incrémentale, organisée en sprints pendant lesquels l'équipe scrum va produire un incrément potentiellement livrable du produit [13]. La durée des sprints est définie en amont du projet, elle est généralement de 2 ou 3 semaines pour permettre à l'équipe de produire l'incrément du produit.

Le cadre est organisé autour de trois rôles, d'artéfacts et de cérémonies :

- Les **rôles** sont le « Product Owner », le « Scrum Master » et l'équipe de réalisation.
- Les **artéfacts** sont les outils utilisés aidant l'organisation et le pilotage.
- Les **cérémonies** sont les différentes réunions cadrant le sprint et ayant lieu de manière fixe à chaque sprint.

L'intérêt de ces rôles, artéfacts et cérémonies est de donner un cadre fixe et facilement compréhensible.

L'organisation en sprints permet de faire de nombreuses itérations. Il s'agit d'une méthode empirique qui se base sur l'expérience.

La méthode fixe un cadre méthodologique, la manière de faire varie ensuite en fonction des équipes et des expériences.

Il n'y a pas une seule manière mais des manières de faire Scrum. La méthode ne dit pas comment appliquer

mais dit ce qui doit être fait via les rôles, artefacts et cérémonies.

Le cadre méthodologique SCRUM est particulièrement adapté pour des projets complexes, comme par exemple ceux de développements informatiques.

4.1 LES CÉRÉMONIES SCRUM

Dans Scrum tout est placé dans une « **boîte temporelle** » (**Time box**) : du temps est alloué pour chaque « cérémonie » afin de cadrer l'organisation pour la laisser produire pendant la durée du sprint. Les « cérémonies » sont :

Sprint Planning : au cours de cette réunion l'équipe Scrum estime la complexité et sélectionne les récits qu'elle pense pouvoir démontrer à la fin du sprint et décide d'un but pour le sprint (figure 6).

L'équipe de réalisation découpe ensuite les récits en tâche. Les récits sont présentés à l'équipe par le « Product Owner » et le choix est fait par consensus, rien n'est imposé, le dialogue au sein de l'équipe est vraiment important pour lever toutes les zones d'ombres et gagner en visibilité.



Figure 6 : L'équipe Scrum, et le Product Owner définissent le Sprint Planning (source : auteurs)

Daily Scrum, également appelé "mêlée journalière", c'est de là que vient le nom de la méthode.

C'est une réunion **généralement debout** qui a lieu tous les jours au même endroit (figure 7).

Elle est délimitée dans le temps (Time-boxing) comme toutes les réunions Scrum et ne doit pas excéder **15 minutes**. Si les 15 minutes sont dépassées c'est que des sujets qui ne concernent pas toute l'équipe sont abordés et il faut préparer des réunions plus ciblées.

Durant cette réunion l'équipe présente ce qu'elle a fait la veille, ce qu'elle compte faire aujourd'hui et les éventuelles difficultés qu'elle a rencontrées pour atteindre le but du sprint ou dans l'avancement.

C'est une réunion d'information qui permet de donner de la visibilité et de la transparence sur ce que chacun fait, elle permet de se focaliser sur le but.

La réunion se fait devant le tableau de l'équipe basé sur le **management visuel** (tableau Kanban) afin d'aider à la visualisation des flux de travail.

En outre, il peut être une base de motivation pour l'équipe : il y a un **sentiment de satisfaction** lorsqu'on accomplit une tâche.



Figure 7 : La « Daily Scrum » génère la dynamique opérationnelle de l'équipe Scrum (source : auteurs)

Sprint Review, aussi appelée « Revue de Sprint », qui permet à l'équipe de présenter ce qu'elle a réalisé pendant le sprint (figure 8).

Le « Product Owner » valide ensuite en fonction des critères d'acceptation si le récit est validé ou non.

C'est une réunion où l'équipe est mise en valeur et où elle tire les fruits de son travail.



Figure 8 : La « Sprint Review » favorise la communication interne de l'équipe Scrum (source : auteurs)

Retrospective, cette réunion clôture le sprint et permet à l'équipe Scrum de faire le bilan et le retour d'expérience sur le sprint (figure 9).

C'est une étape essentielle qui permet à l'équipe de s'améliorer de manière continue.

Le but de cette réunion de **capitaliser ce qui a marché** et de proposer un plan d'action pour améliorer le processus.

L'équipe **apprend de ses erreurs** pour faire encore mieux au prochain sprint. Elle est souvent organisée sous forme d'atelier pour permettre à l'équipe de parler et d'échanger.

Le Scrum Master anime cette réunion et ses talents de facilitateur sont essentiels, il doit faire preuve d'écoute active, de reformulation et d'un bon esprit de synthèse pour permettre à chacun de s'exprimer et pour récolter le ressenti de chacun.



Figure 9 : L'équipe Scrum capitalise sur ses retours d'expérience et génère de l'amélioration continue de ses pratiques (source : auteurs)

4.2 LES RÔLES DANS UNE ÉQUIPE SCRUM

Le « **Product Owner** » (**PO**) est le garant du produit, il a la vision de ce qu'est le produit et ce qui doit être fait en priorité pour maximiser la valeur. C'est un rôle tenu par une seule personne.

C'est lui qui se charge de récolter les besoins des utilisateurs et de les retranscrire dans le « **product backlog** », qu'il doit également entretenir et prioriser. C'est lui qui valide si une fonctionnalité a été correctement réalisée.

C'est le référent de l'équipe de réalisation, si une question se pose sur le produit c'est à lui qu'il faut s'adresser en priorité.

Ce rôle nécessite un investissement élevé (c'est-à-dire d'être sur un seul projet) et une grande disponibilité pour pouvoir répondre aux questions de l'équipe de réalisation.

Un PO qui n'assume pas toutes ses fonctions, notamment de clairement identifier le besoin utilisateur ou d'être peu disponible peuvent être des causes d'échecs d'un projet agile.



Le « Scrum Master » veille au respect de la méthodologie et veille à ce que rien n'entrave l'équipe pendant le sprint. Il anime les cérémonies de la méthode.

Il est nécessaire qu'il ait une bonne connaissance de la méthode et de l'agilité en générale. Il est souvent l'intermédiaire entre le PO et l'équipe de réalisation.

Son but est **d'aider l'équipe à travailler de façon autonome** et à s'améliorer.

Il n'impose rien à l'équipe, ce n'est pas un chef de projet. Il est plus un coach et un facilitateur pour l'équipe. Il aide l'équipe à s'améliorer et à gagner en autonomie.



L'équipe de réalisation réalise le produit (Figure 10). Elle est auto-organisée et autonome, c'est-à-dire qu'elle définit son organisation pour atteindre ses objectifs.

Cela signifie que c'est l'équipe qui définit son organisation, **rien ne lui est imposé**, il doit toujours y avoir un dialogue et un consensus entre elle et le PO.

Toutes les compétences sont réunies au sein de l'équipe pour produire. Sa taille ne doit pas dépasser 9 membres afin de faciliter l'échange et la circulation de l'information. Elle doit être composée d'individus motivés qui participent au projet par choix.



Figure 10 : Découpage en tâche par l'équipe de réalisation
(source : auteurs)

L'intérêt de définir ces trois rôles sert à clairement définir qui fait quoi dans le projet et qui est porteur de l'information en cas de besoin, il s'agit de limiter les intermédiaires. En outre l'équipe dans sa globalité est restreinte à 11 membres maximum pour permettre à l'information de circuler plus facilement (transparence) et pour que chacun est sa place et son importance dans l'équipe. Au-delà de ce nombre, il sera nécessaire de former plusieurs équipes agiles.

4.3 LES ARTÉFACTS SCRUM

Un artéfact est un objet renfermant une information. Il peut s'agir d'une feuille, d'un fichier Excel ou encore d'un gestionnaire de tâches qui vont renfermer de l'information une fois intégrée dans un environnement de travail.

Dans le cadre de Scrum, les artéfacts vont désigner tous les outils qui vont aider à l'organisation et au pilotage du projet [15].

Le « **Product Backlog** » est une liste de fonctionnalités, besoins, exigences qui va permettre de donner la vision produit. C'est le « Product Owner (PO) » qui est responsable du « backlog », notamment de son organisation et de sa priorisation. Il va le prioriser en fonction de la valeur.

Le « backlog » est vivant et évolue en fonction des besoins du « PO », ce n'est donc pas un cahier des charges fixe. Il est généralement organisé en récits utilisateurs (« **User Story** » ou US) pour maximiser la valeur et se focaliser sur le besoin des utilisateurs (Figure 11).

Le niveau le plus fin de granularité est la « User Story ». Plusieurs « User Story » sont nécessaires pour réaliser **une fonctionnalité**. Enfin un ensemble de fonctionnalités permet de mettre en place **une épique ou thème**. Un exemple concret peut être celui de la rédaction d'un article, c'est une épique. Le niveau en dessous est constitué des différentes sections de l'article. Enfin le niveau le plus fin peut être les différents paragraphes.

Ces récits permettent de répondre aux questions qui, quoi et pourquoi de la fonction. Ces questions peuvent être reformulées par « En tant que (qui), je veux que (quoi), pour que (pourquoi) ».

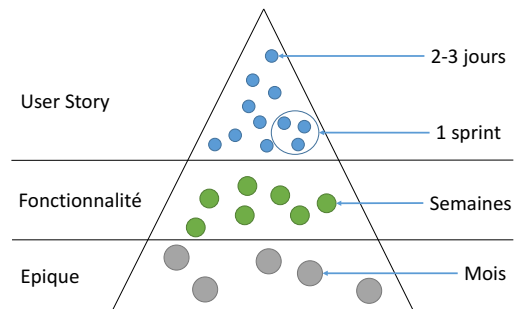


Figure 11 : Organisation du backlog produit [16]

Le simple fait d'écrire le besoin sous forme de récit utilisateur n'est pas suffisant pour permettre à l'équipe de produire. Une pratique pour avoir de bons récits utilisateurs est d'utiliser l'approche « **INVEST** » qui est l'acronyme de « Indépendante », « Négociable », « Valorisable », « Estimable », « Small (petite) », « Testable » (Figure 12).

L'indépendance signifie que l'ordre de réalisation n'a pas d'importance. Le caractère **négociable** du récit vient du fait que ce n'est pas un cahier des charges fixe, la meilleure solution de réalisation est obtenue par un consensus. Le récit doit pouvoir être ordonné et priorisé selon son importance ou sa **valeur**. Lorsqu'on lui présente le récit, l'équipe doit pouvoir **estimer** sa complexité. Le récit doit aussi être **petit** pour le réaliser en un seul sprint et **testable** d'un point de vue fonctionnel, c'est-à-dire que le travail réalisé doit pouvoir être vérifié.

L'usage de l'approche « INVEST » va permettre de faire mûrir le besoin et de le clarifier, ce qui est essentiel pour que l'équipe de réalisation comprenne le besoin et fabrique un produit au plus près de ce qu'attend le « PO ». La constitution du backlog est une étape préalable d'un projet agile, l'entretien de cet outil se fait tout au long du projet via des ateliers et réunions avec l'équipe de réalisation.

I	Indépendante
N	Négociable
V	Valorisable
E	Estimable
S	Small (petite)
T	Testable

Figure 12 : Une bonne pratique définissant un bon récit utilisateur : INVEST [16]

Le « **Sprint Backlog** » est un ensemble de récits sélectionnés par l'équipe pour le sprint. L'équipe **s'engage sur les récits qu'elle pourra terminer** et démontrer à la fin du sprint.

Les récits sélectionnés sont ensuite découpés en tâches pour aider l'équipe à les réaliser. Comme vu précédemment les récits permettent de répondre aux questions qui, quoi, pourquoi. Le découpage en tâches va permettre de répondre à la question « Comment ? ». Ce découpage permet à l'équipe de se projeter et de visualiser le travail à faire au cours du sprint.

Une autre bonne pratique pour le découpage en tâche est d'utiliser « **SMART** », acronyme de « **Simple** », « **Mesurable** », « **Achievable** (réalisable) », « **Relevant** (pertinent) » et « **Time-boxed** » (figure 13).

S	Simple
M	Mesurable
A	Achievable (réalisables)
R	Relevant (Pertinentes)
T	Time-boxed

Figure 13 : SMART est une bonne pratique pour le découpage en tâches [16]

Le découpage en tâches « SMART » va permettre de se focaliser sur l'objectif et de **gagner du temps** pour réaliser le récit.

La tâche doit être simple, c'est à dire définir clairement l'objectif avec un verbe d'action. L'objectif être mesurable, c'est-à-dire qu'on peut identifier quand la tâche est terminée. Un membre de l'équipe à la compétence pour être sûr que la tâche soit réalisable. Enfin la tâche doit être pertinente pour atteindre l'objectif et planifiée (time-boxed) pour ne pas prendre plus de 2 ou 3 jours.

Les **rapports ou indicateurs** permettent d'aider au pilotage du projet et sont essentiels pour mesurer l'avancement du projet, connaître les risques ou faire remonter des alertes. Ce sont des outils de visualisation et de prise de décisions.

Le **pilotage** peut être réalisé au niveau du sprint via le comptage et le suivi des tâches restant à réaliser pendant pour le sprint. Le pilotage se fait également à l'échelle du projet via les estimations des récits utilisateurs : connaissant l'estimation de chaque récit et ce que peut réaliser l'équipe par sprint, le « Product Owner » peut faire des projections à plus ou moins long terme.

L'estimation de la complexité des récits par l'équipe est un très bon outil d'aide pour la prise de décision en permettant de distinguer si les récits sont clairs, complexes et suffisamment fins. Si ce n'est pas le cas, les récits peuvent être découpés plus finement toujours selon la méthode « INVEST ».

Conclusion

L'agilité est utilisable dans de nombreux environnements mais plus particulièrement adaptée aux environnements complexes.

L'agilité offre une alternative à l'approche classique de « gestion de projet » en proposant **une meilleure adaptabilité et une plus grande transparence**.

« **Scrum** » (mêlée) est la méthode agile la plus utilisée, car elle permet grâce à ses **rôles, artefacts et cérémonies** de mettre en place un cadre simple, fluide et efficace pour la gestion de projet (figure 14).

Le recueil du besoin et sa formulation sous forme de récits utilisateurs est la première étape pour constituer un « backlog produit » et commencer un projet agile.

La méthode « **INVEST** » sert ici pour écrire de bons récits utilisateur. Le découpage en tâches et le management visuel sont des pratiques de plus en plus répandues pour être performant. L'approche « **SMART** » est une pratique agile qui pourra être utile pour un découpage en tâches plus efficace.

Avec Scrum, et plus généralement avec les approches agiles, les entreprises ont l'opportunité de **développer rapidement de nouveaux produits** et services capables de suivre les évolutions des attentes clients.

Il est donc utile d'encourager leur emploi et de faciliter leur mise en œuvre afin de contribuer à la compétitivité des entreprises et au développement économique et sociétal.

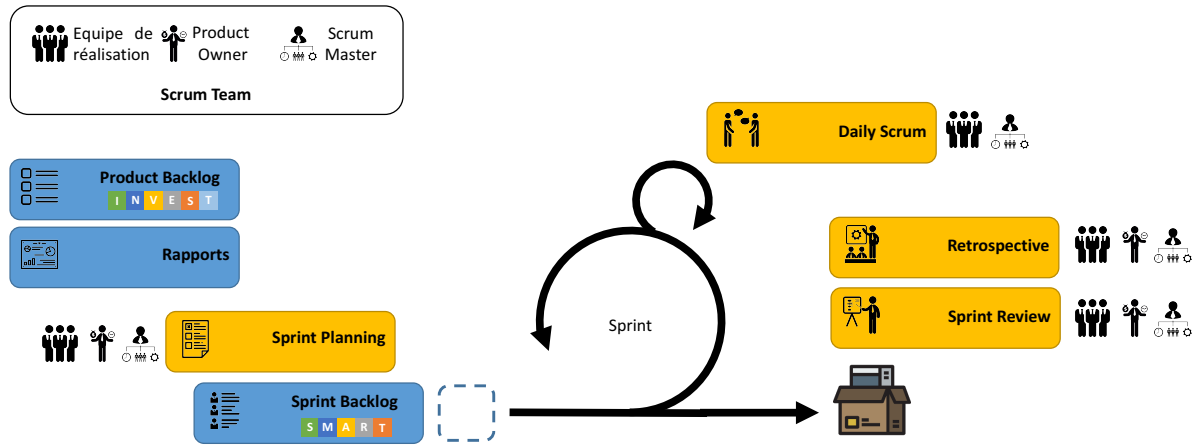


Figure 14 : Le cadre méthodologique Scrum en un coup d'œil : cérémonies, rôles et artéfacts [16]

Références bibliographiques

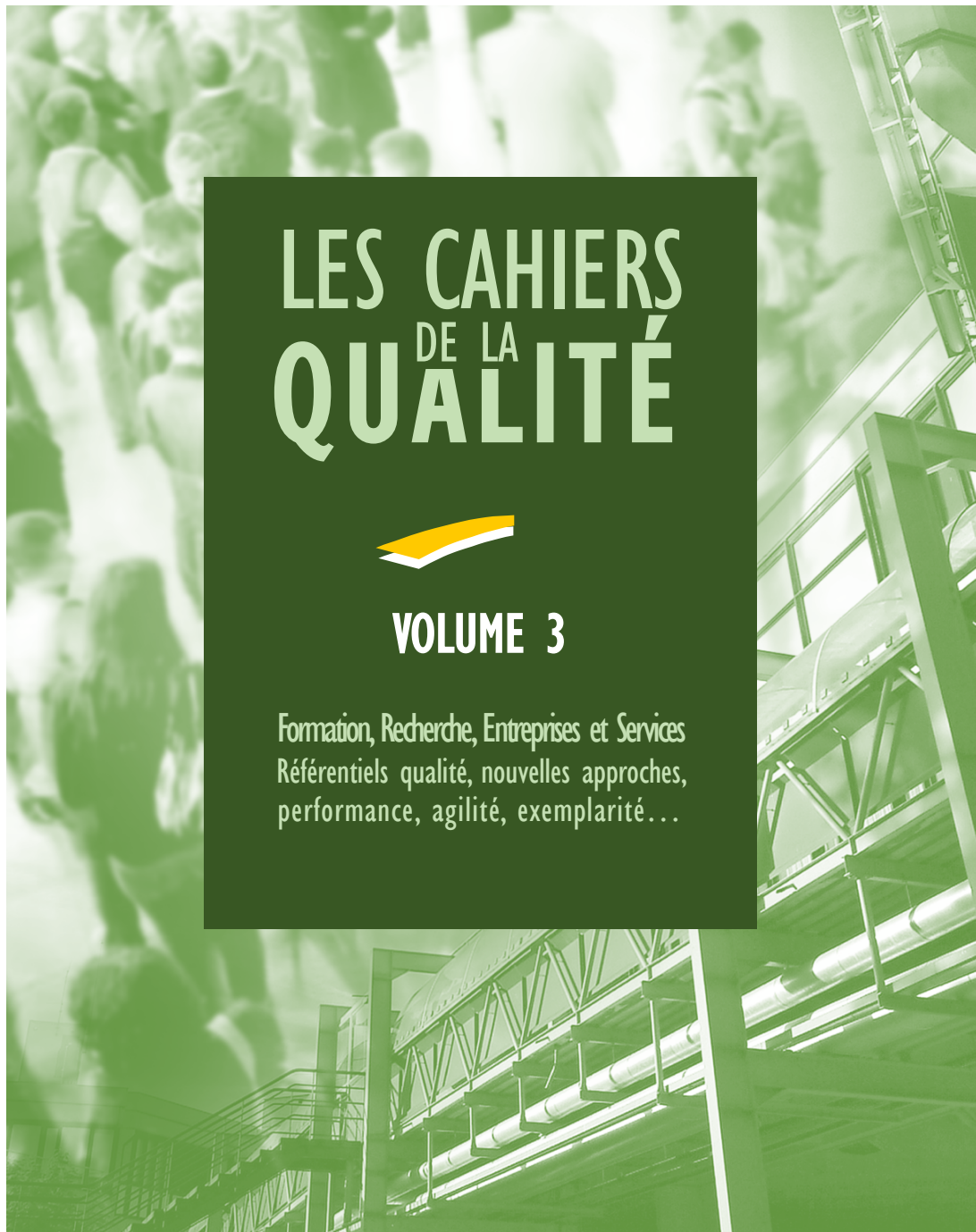
- [1] "FD ISO 10006 - Systèmes de management de la qualité - Lignes directrices pour le management de la qualité dans les projets - Tirage 2." Editions Afnor, Paris, www.afnor.org, 01-Jan-2005.
- [2] "NF ISO 21500 - Lignes directrices sur le management de projet." Edition Afnor, www.afnor.org, 01-Oct-2012.
- [3] S. Hastie and S. Wojewoda, "Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch," 04-Oct-2015. [Online]. Available: <https://www.infoq.com/articles/standish-chaos-2015>. [Accessed: 03-Jun-2017].
- [4] R. D. Stacey, *Strategic management and organisational dynamics: the challenge of complexity to ways of thinking about organisations*, 5th ed. Harlow: Financial Times Prentice Hall, 2007.
- [5] "waterfall model." [Online]. Available: <http://www.riff.org/node/13502>. [Accessed: 03-Jun-2017].
- [6] "Cycle en V," *Wikipédia*. 10-Nov-2016.
- [7] Access Dev - Agence de Communication Digitale à Montpellier, "La gestion de projet : méthode classique VS méthodes agiles." [Online]. Available: <http://www.access-dev.com/access-dev/la-gestion-de-projet-methodes-classiques-vs-methodes-agiles/>. [Accessed: 03-Jun-2017].
- [8] F. Thomas, "L'industrialisation informatique : l'alternative aux réductions budgétaires," 27-Aug-2010. [Online]. Available: <http://www.journaldunet.com/solutions/expert/48198/l-industrialisation-informatique--l-alternative-aux-reductions-budgetaires.shtml>. [Accessed: 03-Jun-2017].
- [9] Kent Beck et al., "Manifeste pour le développement Agile de logiciels," 02-Nov-2001. [Online]. Available: <http://agilemanifesto.org/iso/fr/manifesto.html>. [Accessed: 17-May-2017].
- [10] "10th Annual State of Agile Report," Nov-2015. [Online]. Available: <https://www.versionone.com/about/press-releases/versionone-releases-10th-annual-state-of-agile-report/>. [Accessed: 03-Jun-2017].
- [11] N. Hennion, "Cycle en V : pourquoi est-il si méchant ?," *Au coude à coude*. [Online]. Available: <http://aucoudeacoude.typepad.com/aucoudeacoude/2011/04/ce-qui-ne-marche-pas-dans-le-cycle-en-v.html>. [Accessed: 05-Jun-2017].
- [12] J. Martin, *Rapid Application Development*. Indianapolis, IN, USA: Macmillan Publishing Co., Inc., 1991.
- [13] Jeff Sutherland and Ken Schwaber, "The Scrum Guide (TM)." Scrum.Org and ScrumInc, www.scrumguides.org, Jul-2013.
- [14] K. Beck and C. Andres, *Extreme Programming Explained: Embrace Change*, Second Edition. Boston: Addison Wesley, www.awprofessional.com, 2004.
- [15] James Macanufo, Sunni Brown, and Dave Gray, *Games-torming : Jouer pour innover : pour les innovateurs, les visionnaires et les pionniers*. Editions Diateino, 2014.
- [16] G. Kurzawa, "AGILITE EN PRATIQUE : SCRUM, Un cadre propice à la transparence et à l'amélioration continue," Master Qualité et Performance dans les Organisations (QPO), Mémoire d'Intelligence Méthodologique du stage professionnel de fin d'études, www.utc.fr/master-qualite, puis "Travaux" "Qualité-Management", réf n°398, Université de Technologie de Compiègne, Juin 2017.

Bonus

Téléchargeables gratuitement sur le site UTC :

- www.utc.fr/master-qualite puis « Travaux » « Qualité-Management », réf n°398.
- [Mémoire](#)
- [Poster](#)

Cet article est publié dans :



Commande sur :

<https://www.amazon.fr/Cahiers-Qualit%C3%A9-Entreprises-R%C3%A9f%C3%A9rentiels-performance/dp/1973283026>

© Université de Technologie de Compiègne, Editeur Indépendant
Les Cahiers de la Qualité – Volume 3

ISBN-10 : 1973283026 - ISBN-13 : 978-1973283027- ASIN : 1973283026 - Dépôt légal : 7 janvier 2018
UTC - rue du docteur Schweitzer - CS 60319 - 60203 COMPIEGNE Cedex – France – www.utc.fr