

Bassem EL HADDAD

Julien OGER

SCRUM

De la théorie à la pratique

2^e édition



+ un memento de 14 pages !

● Éditions
EYROLLES

SCRUM

De la théorie à la pratique

Un livre-outil pour maîtriser Scrum

En génie logiciel, beaucoup de méthodes dites agiles sont apparues ces dernières années: Scrum, Lean, eXtreme Programming... Prenant le contre-pied des méthodes traditionnelles prédictives, elles permettent d'éviter une définition trop précoce et figée des besoins grâce à leur souplesse et leur adaptabilité. Parmi ces méthodes, Scrum s'est peu à peu imposée comme la méthode agile la plus populaire, bien loin maintenant devant les autres.

Nourrie par la grande expérience métier de ses auteurs, la deuxième édition de cet ouvrage, mise à jour et complétée, permettra au chef de projet informatique d'appréhender et d'utiliser Scrum au quotidien, avec à la clé beaucoup de conseils pratiques et d'exemples de cas concrets. Tous les rôles, événements et artefacts liés à ce cadre agile y sont détaillés pour qu'il puisse acquérir les bases de cette méthode ou consolider ses connaissances. Il découvrira également comment adapter Scrum à ses problématiques et l'associer à d'autres concepts tels que le Kanban ou le Lean Management.

Pour aider le lecteur dans son apprentissage et l'accompagner dans ses missions, ce livre est complété par un dépliant détachable de 14 pages qui regroupe l'ensemble des principes fondamentaux de Scrum.

À qui s'adresse cet ouvrage ?

- Aux chefs de projets informatiques, architectes logiciels, développeurs, analystes, testeurs...
- À ceux qui débutent avec Scrum comme aux plus expérimentés souhaitant se perfectionner.

Chef de projet et certifié Scrum Master, **Bassem El Haddad** accompagne ses clients dans la réalisation, le cadrage et le pilotage de projets agiles. Il s'intéresse ainsi notamment au concept de l'agilité et à la gestion de l'humain pour mener à bien ses projets, quels que soient leur taille et leur domaine de mise en œuvre.

Directeur de projet, **Julien Oger** est en charge de l'élaboration et de l'organisation de méthodes de travail, basées sur les préceptes agiles, pour de petits et grands comptes sur des sujets divers (recherche, santé, assurance, banque, associatif, formation...).

Au sommaire

L'agilité : une approche, plusieurs « méthodes ». Les méthodes agiles • Des méthodes ? Une philosophie ! • Agiles vs classiques • Une multitude de « méthodes » • DevOps, une évolution naturelle • Scrum, un framework agile • **Scrum au cœur de la gestion de projet.** Gérer un projet informatique • Organisation et étapes clés • Différents types d'acteurs • Et Scrum dans tout ça ? • **Les bases de Scrum.** Rôles mis en jeu • Scrum Master • Product Owner • Équipe de développement • Autres acteurs • Événements • Le sprint • La mêlée quotidienne • Artefacts • Les carnets • Incrément de produit • Un processus complet • Évolutions de Scrum • **Se préparer à Scrum.** Les clés pour se préparer • Les questions à se poser • Gestion humaine • Démarche d'amélioration continue • Organiser sa méthode • Mettre en place une stratégie • Organiser ses équipes et le changement • **Se lancer avec Scrum.** Processus global • Phase d'initialisation • Organiser et s'approprier le projet • Définir les attentes • Planifier • Démarrer la production • Phase de préparation • Planifier la release • Préciser les récits utilisateurs • Organiser le sprint • Phase de production • État des lieux récurrent • Phase de finalisation • Revue et livraison du sprint • Rétrospective et introspection • Mesure et suivi • La livraison, et après ? • **Optimiser sa méthode.** Différents niveaux d'adaptation • S'adapter à l'humain et au contexte du projet • Être agile avec Scrum • Scalabilité de Scrum • Étendre Scrum • Techniques liées à la production • DevOps et le futur de l'agilité • **Étude d'un cas pratique.** Contexte de mise en œuvre • Phase d'initialisation • Phase de préparation • Phase de production • Phase de finalisation • **Pour aller plus loin.** Outils : supports de la méthode • Processus de perfectionnement • Formations et certifications.

Scrum

De la théorie à la pratique

CHEZ LE MÊME ÉDITEUR

A. GAUNAND. – **Le leadership agile.**
N°56489, 2017, 224 pages.

V. MESSENGER. – **Gestion de projet agile.**
N°13666, 2013, 278 pages.

Y. CONSTANTINIDIS. – **Mémento Cahier des charges informatique (3^e édition).**
N°11880, 2016, 14 pages.

Y. CONSTANTINIDIS. – **Expression des besoins pour le SI (4^e édition).**
N°67577, 2018, 360 pages.

P. TACHÉ. – **Conduire un projet informatique.**
N°55915, 2014, 192 pages.

O. ENGLEDER, S. FERNANDES. – **Manager un projet informatique (4^e édition).**
N°56675, 2017, 340 pages.

H.-P. MADERS. – **Animer une équipe projet avec succès.**
N°55523, 2012, 264 pages.

A. FERNANDEZ-TORO. – **Sécurité opérationnelle (2^e édition).**
N°14460, 2016, 424 pages.

C. HOHMANN. – **Lean Management.**
N°55381, 2012, 424 pages.

R. HENNION, A. MAKHLOUF. – **Les fiches-outils – Focus du Lean Six Sigma.**
N°56372, 2016, 216 pages.

Bassem El Haddad
Julien Oger

Scrum

De la théorie à la pratique
Initiation • Perfectionnement • Agilité

2^e édition

● Éditions
EYROLLES

ÉDITIONS EYROLLES
61, bd Saint-Germain
75240 Paris Cedex 05
www.editions-eyrolles.com

En application de la loi du 11 mars 1957, il est interdit de reproduire intégralement ou partiellement le présent ouvrage, sur quelque support que ce soit, sans l'autorisation de l'Éditeur ou du Centre Français d'exploitation du droit de copie, 20, rue des Grands Augustins, 75006 Paris.

© Groupe Eyrolles, 2017

© Éditions Eyrolles, 2019, ISBN : 978-2-212-67862-8

Remerciements

Il y a un peu plus de deux ans paraissait la première édition de *Scrum, de la théorie à la pratique*, afin d'accompagner les plus novices comme les plus avertis dans l'utilisation de la méthode Scrum.

Face à l'excellent accueil qu'a reçu cet ouvrage, c'est avec un immense plaisir que nous avons décidé d'en proposer une deuxième édition. Nous souhaitons donc avant tout remercier l'ensemble de nos lecteurs et lectrices qui nous ont donné la motivation nécessaire pour « remettre le couvert ». Un très grand MERCI à vous !

Nous tenons aussi à remercier nos proches et nos familles pour leur soutien indéfectible depuis le début, car c'est grâce à eux que nous avons pu écrire sereinement et produire ce que nous souhaitions exactement.

Merci également à Arnel Fauveau et Brigitte Aubine pour leur aide sur la première édition.

Un grand merci encore à Eyrolles pour le renouvellement de leur confiance et pour l'accompagnement sans faille de toute l'équipe éditoriale.

Enfin, nous vous remercions vous, qui avez acheté cette nouvelle édition, et espérons que ce livre sera à la hauteur de vos attentes.

Avant-propos

C'est à la fin de l'année 2015 que nous avons décidé de nous lancer dans la réalisation d'un ouvrage dédié à Scrum. Comme c'est encore le cas aujourd'hui, cette méthode faisait partie de notre quotidien : nous la mettions en œuvre, la faisons vivre, l'adaptions aux contextes de nos projets, formions nos équipes... Il nous semblait donc légitime de partager nos nombreuses expériences avec Scrum, chose faite au printemps 2017 avec la première édition de cet ouvrage.

Les retours très positifs autour du livre et certaines adaptations que nous souhaitions apporter depuis quelques mois nous ont poussés à envisager une seconde édition, que vous avez aujourd'hui entre les mains. Avec cette mise à jour, nous avons voulu principalement améliorer l'expérience de lecture sans chambouler notre idée de départ : proposer un livre didactique et une approche progressive. Chaque chapitre a donc été réfléchi et travaillé pour une montée en compétence progressive et logique, en commençant par les bases de la gestion de projet jusqu'aux moyens de mise en œuvre, d'adaptation et de perfectionnement.

Nous espérons que ce livre vous apportera une meilleure lisibilité de Scrum et de l'écosystème agile. À vous de jouer !

Pourquoi choisir Scrum ?

Parmi les méthodes agiles, Scrum se démarque depuis plusieurs années grâce en particulier à un très bon support marketing. Mais il serait réducteur de s'arrêter à cet aspect car si l'utilisation de ce cadre de travail a pris une telle importance, c'est pour une raison toute simple : son efficacité. Mettre en place Scrum apporte en effet des atouts indéniables qui permettent d'améliorer l'organisation du projet et la qualité du produit réalisé. Ce cadre offre une dynamique positive à vos projets en proposant notamment une logique d'amélioration continue et une philosophie plaçant l'équipe au centre du projet.

Par ailleurs, ce qui séduit avant tout dans l'utilisation de Scrum, c'est sa capacité à pouvoir évoluer et s'adapter en fonction du contexte. Il ne s'agit pas là d'une méthode monolithique et totalement rigide mais, au contraire, d'une boîte à outils qui vous propose des mécanismes ajustables en fonction de vos besoins.

Une approche progressive

Mettre en place une méthode efficace basée sur Scrum ne se fait pas d'un claquement de doigts. Certes, Scrum apporte une base solide mais encore faut-il pouvoir la mettre en œuvre de manière pertinente. C'est là tout l'enjeu de cet ouvrage : vous apporter les clés permettant d'optimiser l'utilisation de Scrum.

Pour arriver à un niveau d'utilisation convenable de la méthode, cela demande généralement du temps et de l'expérience. Si cette prise d'expérience reste indispensable, ce livre vous permettra toutefois de structurer votre approche et vous apportera des bases solides pour une application efficace de Scrum dans votre organisation.

Pour cela, nous avons choisi de proposer un rythme de lecture progressif partant des bases de l'agilité et de la gestion de projet, jusqu'à la possibilité d'adaptation de Scrum. Notre objectif dès le démarrage de ce livre était de faire en sorte que le lecteur ressente cette progression chapitre après chapitre, et qu'il évolue dans l'apprentissage de Scrum.

En effet, si Scrum propose un processus complet, il vous laisse relativement libre sur le mode de mise en œuvre de votre méthode. C'est, entre autres, ce qui explique le minimum d'accompagnement que Scrum va exiger si vous souhaitez être efficace rapidement.

Ce livre vous propose de bénéficier de l'expérience issue de plusieurs dizaines de projets et d'éviter ainsi les erreurs les plus courantes. Par ailleurs, il permet de bénéficier d'une boîte à outils complète permettant de piocher dans différents concepts pour compléter Scrum.

Au final, *Scrum, de la théorie à la pratique* est un support d'accompagnement réunissant tout le nécessaire pour démarrer et/ou se perfectionner avec la méthode. Nous y avons intégré notre connaissance et surtout notre expérience, incluant notamment des retours clients et des exemples de mises en œuvre (concluantes ou non).

À qui s'adresse ce livre ?

Cet ouvrage, de par son approche pédagogique, s'adresse aussi bien à un public expérimenté qu'à de parfaits novices en la matière. Il conviendra à ceux qui sont déjà familiers de la méthode ou qui font déjà partie d'une équipe Scrum, mais il conviendra également à des profils plus juniors en gestion de projet ou totalement nouveau dans le domaine de l'agilité.

Le rythme de ce livre permet de démarrer doucement, en assimilant toutes les bases de Scrum ainsi que son vocabulaire qui lui est propre. Les débutants y trouveront ainsi des bases solides permettant de découvrir les bases de la gestion de projet agile avant de se plonger plus spécifiquement dans Scrum. Les lecteurs plus expérimentés quant à eux, vont revoir et approfondir des principes déjà acquis et appliqués. Ils pourront également trouver leur compte en basculant sur les chapitres consacrés à la mise en œuvre et illustrés de cas pratiques concrets.

Structure de l'ouvrage

Le livre est composé de huit chapitres qui proposent, dans leur globalité, une démarche progressive comme on pourrait la trouver dans la pratique d'un sport (au hasard, le rugby) :

- **Chapitre 1 - L'initiation (phase 1)** : le premier chapitre est une introduction à l'agilité et à Scrum. À travers les présentations et les historiques, ce chapitre permet de prendre le temps d'entrer dans le concept de « méthode agile » et des généralités sur Scrum.
- **Chapitre 2 - L'initiation (phase 2)** : cette seconde phase d'initiation porte cette fois sur les principes fondamentaux de la gestion de projet. Nous y abordons le fonctionnement général des projets informatiques et l'impact des méthodes (dont Scrum) sur l'organisation.
- **Chapitre 3 - L'échauffement** : le troisième chapitre permet de passer à la vitesse supérieure, en entrant dans une phase de prise en main de Scrum. On y retrouve des explications détaillées sur les rôles, les événements, les artefacts et le processus de production impliqués par l'utilisation de Scrum.
- **Chapitre 4 - Le briefing** : les bases sont posées, le vocabulaire et les éléments Scrum sont connus : il est temps de passer au « briefing d'avant-match » pour préparer au mieux la mise en place de sa méthode. C'est l'objectif de ce quatrième chapitre qui propose de consolider les bases de votre méthode avant de vous lancer dans l'aventure.
- **Chapitre 5 - Le coup d'envoi** : il est maintenant temps de passer à l'action ! Vous allez à présent entrer dans l'application concrète de Scrum en suivant l'ensemble des processus d'un projet standard. Chaque étape du projet est détaillée et propose une organisation type et les moyens à mettre en œuvre pour mettre toutes les chances de votre côté.
- **Chapitre 6 - La prolongation** : les projets types sont une chose mais la réalité impose parfois une certaine dose d'adaptation. Ce sixième chapitre propose des solutions pour adapter Scrum à vos problématiques, notamment métiers, organisationnelles, RH, etc. Il offre aussi une vision élargie permettant l'utilisation de votre méthode sur des projets d'envergure, mais aussi de l'associer à d'autres concepts (Lean, Kanban, DevOps, etc.)
- **Chapitre 7 - Le debrief** : le septième chapitre présente une étude de cas où Scrum est appliqué sur un projet de A à Z. Nous nous sommes basés sur notre expérience et certaines des problématiques les plus courantes pour proposer des solutions à des cas concrets.
- **Chapitre 8 - Le perfectionnement** : vous l'aurez compris, le match n'est pas terminé ! C'est vous qui récupérez le témoin et ce sera à vous désormais d'apprendre à utiliser les outils Scrum, de suivre des formations, de passer des certifications, et ainsi de décider du déroulement de ces prochains matchs. Ce dernier chapitre vous donne les pistes pour vous perfectionner et aider vos équipes à faire de même.

À ces chapitres s'ajoute un complément : le memento. Cet aide-mémoire peut se lire indépendamment du livre et se trouve à la fin de cet eBook. Tous les principes fondamentaux de Scrum y sont contenus et vous accompagneront dans vos déplacements.

À propos des auteurs



Diplômé d'un titre d'ingénieur en architecture logicielle, **Bassem El Haddad** a connu très tôt le monde du Web et de l'entreprise.

Passé par différents types de structures, il débute en tant que développeur sur plusieurs projets avant de découvrir ses nouvelles passions après celle du code : la méthodologie et le management. Bassem se base sur l'hypothèse que la réussite d'un projet passe inévitablement par l'implication des individus et la bonne gestion des profils, tout en s'appuyant sur un cadre organisationnel en guise de fil conducteur.

Aujourd'hui chef de projet et certifié Scrum Master, Bassem El Haddad s'intéresse principalement aux méthodes agiles et à la gestion de l'humain pour mener à bien l'ensemble de ses projets, qu'ils soient de petite taille ou de plus grande envergure.

Vous pouvez le suivre sur Twitter via [@belhaddad75](#).



Julien Oger travaille depuis 15 ans dans le domaine du Web, commençant par du développement pour être aujourd'hui directeur de projet. Il s'occupe entre autres de l'élaboration et de l'organisation des méthodes de travail sur mesure, basées sur les préceptes agiles et empruntant notamment à Scrum et eXtreme Programming.

Ces années d'expérience ont permis de participer à des dizaines de projets allant de quelques jours à plusieurs mois de travail. Ceci pour de petits et grands comptes et sur des sujets aussi divers que la recherche, la santé, l'assurance, la banque, l'associatif, la formation, etc.

Il intervient sur de la gestion de projet mais également de l'assistance à maîtrise d'ouvrage et de l'audit organisationnel.

Vous pouvez le suivre sur Twitter via [@JulienOG](#).

Table des matières

CHAPITRE 1

L'agilité : une approche, plusieurs « méthodes »..... 1

Les méthodes agiles	2
Des méthodes ? Une philosophie !	2
Définir l'agilité	6
Agiles vs classiques	8
Une multitude de « méthodes »	12
DevOps, une évolution naturelle	14
Scrum, un framework agile	17
Présentation de Scrum	17
Ce que propose Scrum	22
Scrum dans la mouvance agile	24

CHAPITRE 2

Scrum au cœur de la gestion de projet..... 27

Gérer un projet informatique	27
Principes fondamentaux	28
Organisation et étapes clés	33
Différents types d'acteurs	36
Et Scrum dans tout ça ?	38
Un support pour la gestion de projet	39
Contexte d'utilisation de Scrum	41

CHAPITRE 3

Les bases de Scrum..... 47

Rôles mis en jeu	47
Scrum Master	48
Product Owner	51
L'équipe de développement	54

Les autres acteurs	56
--------------------------	----

Organisation des acteurs	57
--------------------------------	----

Événements

Le sprint	60
-----------------	----

La mêlée quotidienne	64
----------------------------	----

Artefacts

Pourquoi des artefacts ?	67
--------------------------------	----

Les carnets	69
-------------------	----

Incrément de produit	73
----------------------------	----

Un processus complet

Un produit en évolution constante	75
---	----

Processus type	77
----------------------	----

Évolutions de Scrum

CHAPITRE 4

Se préparer à Scrum..... 83

Les clés pour se préparer	83
---------------------------------	----

Appliquer Scrum : les questions à se poser	84
--	----

La gestion humaine au cœur	
----------------------------	--

de la préparation	88
-------------------------	----

Démarche d'amélioration continue	96
--	----

Organiser sa méthode

Mettre en place une stratégie	99
-------------------------------------	----

Organiser ses équipes	103
-----------------------------	-----

Organiser le changement	106
-------------------------------	-----

CHAPITRE 5

Se lancer avec Scrum..... 113

Processus global	114
------------------------	-----

Découpage du projet et granularité	114
--	-----

Un projet par étapes	117
----------------------------	-----

Phase d'initialisation	119	DevOps et le futur de l'agilité	204
Principe général	120		
Organiser et s'approprier le projet	122		
Définir les attentes	127		
Planifier	133		
Démarrage de la production	136		
Phase de préparation	138		
Principe général	138		
Planifier la release	139		
Préparer le sprint : préciser les récits utilisateurs	142		
Organiser le sprint	146		
Phase de production	150		
Principe général	150		
Transformation	152		
État des lieux récurrent	153		
Phase de finalisation	156		
Principe général	157		
Revue et livraison du sprint	158		
Rétrospective et introspection	160		
Mesure et suivi	161		
La livraison, et après ?	166		
CHAPITRE 6		CHAPITRE 7	
Optimiser sa méthode.....	169	Étude d'un cas pratique.....	211
Différents niveaux d'adaptation	169	Contexte de mise en œuvre	212
S'adapter à l'humain	170	Le projet : CoachOnTheWeb	212
S'adapter au contexte du projet	173	Un cadre simple	212
Être agile avec Scrum	179	Phase d'initialisation	213
Jouer avec l'agilité	179	Étape 1 – Organiser et s'appropriier le projet	213
Scalabilité de Scrum	183	Étape 2 – Définir le besoin	216
Étendre Scrum	191	Étape 3 – Planifier	218
Techniques liées à la production	191	Phase de préparation	220
Concepts complémentaires	196	Étape 1 – Planification de release	221
		Étape 2 – Préparer le sprint	224
		Étape 3 – Organiser le sprint	227
		Phase de production	229
		Étape 1 – État des lieux	230
		Étape 2 – Transformation	232
		Phase de finalisation	233
		Étape 1 – Revue de sprint	234
		Étape 2 – Rétrospective de sprint	236
		CHAPITRE 8	
		Pour aller plus loin.....	239
		Outils : support de la méthode	239
		Processus de perfectionnement	244
		Formations et certifications	245
		Sources	249
		Le mot de la fin	250
		Index.....	251

1

L'agilité : une approche, plusieurs « méthodes »

Il vous est sans doute arrivé de participer à des discussions ou d'effectuer des recherches sur Scrum. À cette occasion, vous avez dû remarquer qu'un autre sujet lui était systématiquement associé : celui des méthodes agiles. Que ce soit lors de formations que vous avez pu suivre, de présentations clients, d'un point d'équipe ou d'une recherche sur Internet, ces deux notions semblent indissociables. Et pourtant, pour être totalement factuel, on peut dès à présent revenir sur deux idées reçues concernant ce lien entre Scrum et les méthodes agiles :

- *Idée reçue n°1 : Scrum n'est pas une méthode au sens propre du terme (et non !).*
- *Idée reçue n°2 : Scrum ne découle pas des méthodes agiles puisque sa conception est antérieure à la création du Manifeste agile fondateur.*

Alors pourquoi dédier ce premier chapitre à la notion d'agilité et aux méthodes associées ? Tout simplement parce que si les méthodes agiles n'ont pas engendré Scrum, celui-ci fait bien partie des premières « méthodes » associées au thème de l'agilité. Par ailleurs, si nous parlons abusivement de « méthode » pour qualifier Scrum, nous verrons par la suite qu'il s'agit essentiellement d'un « cadre de travail » plus que d'une méthode. C'est d'ailleurs selon nous l'un des points les plus intéressants concernant Scrum.

Les méthodes agiles

Dans le monde du développement logiciel, parler de méthodes agiles, et surtout de leur application dans un cadre opérationnel et professionnel, est presque une évidence. Et pourtant, cela n'a pas toujours été le cas. Il y a une dizaine d'années, les méthodes dites « classiques » étaient encore bien ancrées dans les formations universitaires et les pratiques professionnelles. Loin de l'innovation qu'a pu apporter l'agilité par la suite, elle était au début des années 2000 considérée comme expérimentale. Beaucoup la jugeaient inadaptée ou considéraient que ce type de méthode ne pouvait pas fournir de solution viable aux problématiques d'organisation projet.

Aujourd'hui, l'apport des concepts d'agilité est indiscutable et favorise grandement la réussite des projets. Il repose sur un principe fondamental : une philosophie orientée communication.

Des méthodes ? Une philosophie !

Une histoire de pragmatisme

Imaginons le processus suivant : rédiger un cahier des charges de 250 pages, le faire valider au client, briefer les équipes et entrer dans un tunnel de production jusqu'à la livraison finale aux utilisateurs... Tout ceci est extrêmement séquentiel mais certains reconnaîtront sûrement des situations vécues et des expériences malheureuses ! Avec ce mode de fonctionnement, l'application développée répondra vraisemblablement aux exigences initiales du client, mais plus à son besoin *in fine*. Et c'est en somme le schéma que l'on souhaite ne plus produire lorsqu'on met en place une part d'agilité dans sa méthode de travail.

L'élaboration du concept « agile » est justement liée à une réflexion sur le processus de développement logiciel menée au début des années 2000. Le constat est alors clair : le nombre de projets en souffrance est trop important, il est donc essentiel d'élaborer une approche plus pragmatique et adaptée. Cette difficulté d'adaptation des méthodes dites « classiques » pour les projets informatiques, est essentiellement due au fait qu'il s'agit d'un domaine en évolution rapide et constante. Vous pouvez faire ce constat facilement en prenant l'exemple d'Internet et en comparant votre utilisation de la Toile il y a 5 ou 10 ans et celle d'aujourd'hui.

La période semble courte et pourtant, retourner 10 ans en arrière à l'ère du numérique reviendrait presque à faire un bond à la Préhistoire ! Les idées pullulent, les innovations se multiplient et les usages se diversifient, se modifient, puis s'adaptent en conséquence. Il est alors facile de comprendre que lancer des projets de plusieurs mois sans prendre un minimum de recul sur la méthode à mettre en œuvre est un pari très risqué. Un risque toutefois limité par la mise en œuvre des méthodes agiles.

Au final, on peut voir ces méthodes agiles comme un catalogue regroupant des cadres d'organisation projet – comprenant des méthodes, mais pas seulement – qui vont répondre à une exigence : rationaliser les processus de production logiciels afin d'améliorer la satisfaction liée au produit obtenu. Pour cela, il existe quelques techniques et règles simples qui s'articulent essentiellement autour de la communication. Il s'agit principalement de faciliter les interactions au

sein de l'équipe et de s'autoriser des marges de manœuvre pour ajuster le logiciel en fonction de l'évolution des besoins client et du contexte projet.

Naissance d'une culture agile

Les méthodes agiles ne sont pas nées d'un éclair de lucidité pour trouver LA solution permettant d'optimiser l'organisation des projets informatiques. La ratification du Manifeste agile (qui fait référence) date de 2001, mais il découle en réalité de nombreux travaux réalisés en amont. On peut d'ailleurs remonter au milieu du xx^e siècle pour trouver les premières recherches sur les processus itératifs et incrémentaux destinés à l'industrie.

1930-1950 – Amélioration continue et processus industriels

Loin de toute considération informatique, les premières études sur le développement itératif et incrémental (*IID, Iterative and Incremental Development*) ainsi que sur l'amélioration continue ont été réalisées entre les années 1930-1950. Les études de Walter Shewhart et Edwards Deming, sur l'amélioration de la production et de la qualité, vont alors dans une direction totalement différente de ce que propose le taylorisme.

On passe d'une division du travail en tâches simples à une approche systémique centrée sur les interactions (très proche de ce que l'on peut retrouver dans le concept agile) : toute action sur l'entreprise se répercute au niveau de l'individu et inversement. Parmi les enseignements à tirer de ces différentes études, on peut citer les 7 maladies mortelles de l'entreprise, les 14 points de Deming, et le cycle de Shewhart (PDCA, *Plan-Do-Check-Act*) ou roue de Deming dont nous verrons l'intérêt dans la suite de cet ouvrage.

Au niveau informatique, on commence dès les années 1950 à mettre en pratique ces théories organisationnelles sur des projets militaires tels que l'avion expérimental X-15 ou le programme spatial Mercury dans les années 1950. Nous pouvons d'ailleurs faire une petite parenthèse pour parler brièvement de la méthode PERT (*Program Evaluation and Review Technic*), qui consiste à ordonnancer des tâches dépendantes dans le but d'atteindre un objectif. Cette méthode est née suite au désir de la marine américaine de coordonner les efforts de milliers de personnes pour la réalisation de missiles Polaris pendant la guerre froide.

1960-1990 – Du PDCA au toyotisme

Loin de rencontrer le succès escompté aux États-Unis (où l'approche de Taylor était très encrée jusque dans les années 1980), les théories de Deming et Shewhart s'exportent pourtant très bien dans le Japon de l'après-guerre. Les années 1950-1960 voient notamment apparaître les nouvelles organisations du travail de conception japonaise, notamment le toyotisme mis en place par Toyota dès le début des années 1960. Il regroupe plusieurs concepts pour améliorer la production tels que la production en « juste-à-temps » ou le « zéro défaut ».

Même s'il reste éloigné de notre sujet, on peut vraiment prendre en considération les apports du toyotisme sur l'organisation du travail en flux tendu (qui apporte alors les prémices du Lean Management). Il donnera naissance par la suite à différentes notions que l'on retrouve de près ou de loin dans le monde agile. On peut prendre pour exemple :

- La prise en considération de l'individu, qui participe et donne son avis.
- La qualité, qui doit rester un maître-mot sur toute la chaîne de production.
- L'amélioration continue du processus de production, notamment par l'application du concept de KaiZen. Celui-ci est composé des deux idéogrammes japonais *Kai* (« changement ») et *Zen* (« meilleur »). Il prône les petits changements simples et fréquents. Il met notamment en avant les interactions et l'implication des individus.

1991 – RAD, la première méthode agile

Parallèlement à ces dispositifs industriels, l'informatique voit également émerger de nombreuses réflexions sur l'organisation, la méthodologie et le développement logiciel. La première véritable méthode itérative et incrémentale liée à la gestion de projet informatique remonte à 1991 : il s'agit de la méthode RAD (*Rapid Application Development*).

Cette méthode de « développement rapide d'application » peut être considérée comme la première méthode agile. Elle servira d'ailleurs de base à quelques autres méthodes comme RAD 2 ou DSDM.

1991-1999 – Naissance d'un écosystème

Bien avant l'officialisation du concept agile et dans la foulée de la méthode RAD, de nombreuses méthodes ou cadres de travail voient le jour (nous reviendrons sur leur différence par la suite). Ils présentent un point commun, à savoir qu'ils partagent un certain nombre de valeurs et de pratiques axées entre autres sur la communication, l'humain, et la livraison régulière « par morceaux » de l'application.

La plupart de ces méthodes sont développées depuis les années 1980 mais elles sont progressivement officialisées à partir du milieu des années 1990 : 1994 pour la méthode DSDM, 1995 pour Scrum, 1999 pour RAD 2 et eXtreme Programming, 2004 pour Crystal.

2001 – Le Manifeste agile

Après une phase de diversification des méthodes, des travaux communs entre certains des grands acteurs du domaine sont mis en place. L'objectif est d'aboutir à une cohérence sur le concept et les objectifs des méthodes agiles.

C'est ainsi qu'est publié en 2001 l'Agile Manifesto, ou « Manifeste pour le développement agile de logiciels », considéré comme le document de référence sur les méthodes agiles. Il décrit de manière très formelle les grands principes du développement agile.

Les 17 signataires de ce manifeste sont pour la plupart des personnalités importantes du monde de l'informatique et des méthodes de gestion/organisation de projets. Parmi eux, Alistair Cockburn (initiateur de la méthode Crystal), Ward Cunningham (inventeur du concept de Wiki) ou encore Kent Beck (créateur d'eXtreme Programming ou XP).

Bien entendu, Scrum est également représenté par ses deux créateurs, Ken Schwaber et Jeff Sutherland, qui font partie des signataires à l'origine du manifeste.

2016 – Le futur de l'agilité

L'histoire des méthodes agiles ne s'arrête pas en 2001 avec la ratification du Manifeste agile. Il est logique (et indispensable) de voir émerger des adaptations, ajustements et évolutions de ces méthodes. Même si le manifeste original n'a pas subi la moindre modification, il est intéressant de noter deux choses sur les méthodes qui en découlent.

Premièrement, même si le noyau historique des méthodes agiles reste largement utilisé (Scrum et XP en particulier), on note tout de même l'apparition de nouveaux modèles organisationnels : c'est notamment le cas du cadre organisationnel DAD (*Disciplined Agile Delivery*) mis en place par IBM à partir de 2006. Par ailleurs, de nouvelles méthodes sont couramment mises en place dans un grand nombre de structures.

Deuxièmement, on note la montée en puissance de la mouvance DevOps au cours des cinq dernières années. Elle tend à améliorer l'intégration de l'ensemble des équipes techniques et de leur savoir-faire sur les projets. Même si ce mouvement date de 2009, il prend aujourd'hui une importance non négligeable.

Une philosophie agile

Philosophie : manière de voir, de comprendre, d'interpréter le monde, les choses de la vie, qui guide le comportement. Conception de quelque chose qui repose sur un ensemble de principes.

Dictionnaire Larousse

L'un des objectifs de ce livre est évidemment de vous accompagner lors de l'application des techniques d'organisation agiles et plus spécifiquement Scrum. Mais au-delà de l'aspect purement technique, de l'application des bonnes pratiques ou encore des définitions des préceptes de bases, il convient de garder à l'esprit un point essentiel : appliquer les méthodes agiles, c'est également adopter la philosophie qui va avec.

Cet ouvrage met à disposition des méthodes, des outils, et un certain nombre de connaissances qui vous permettront de mettre en application Scrum. Toutefois, se lancer dans le changement et l'amélioration des méthodes sans adopter un état d'esprit adéquat ne garantit pas de bons résultats. C'est d'autant plus vrai pour Scrum, et c'est d'ailleurs la raison pour laquelle nous parlerons ici de l'adoption d'une véritable « philosophie agile ».

Ce sujet n'est pas nouveau et ce thème de philosophie agile est fréquemment mis en avant (*The art of agile development* par James Shore). Mais si nous insistons sur le sujet, c'est qu'il s'agit d'une première étape primordiale dans la mise en place de votre méthodologie : vous mettre en condition pour avoir cette « vision agile » et la faire partager.

Comme nous le verrons tout au long de cet ouvrage, le partage, la confiance, l'ouverture, l'écoute et la transparence sont absolument essentiels pour la réussite d'un projet agile. D'ailleurs, pour résumer cela, on pourrait simplement dire que l'individu est au cœur des méthodes agiles : elles sont appliquées par des humains, au service de l'humain et du projet.

Nous traiterons ces aspects sociologiques un peu plus tard dans ce livre, notamment la nécessité du « savoir travailler ensemble » dans ce qui pourrait s'apparenter à une communauté où chacun a son importance, ses intérêts liés au projet et son rôle au sein de l'équipe.

L'informatique et au-delà...

Pourquoi le développement d'applications aurait-il le monopole de l'agilité ?

La mise en place des méthodes agiles commence aujourd'hui à dépasser le cadre purement informatique. Bien que pensées initialement pour les développements logiciels, d'autres branches s'adaptent progressivement à une approche de gestion calquée sur les méthodes agiles.

Nous sommes bien évidemment aux balbutiements de ces techniques et, pour rappel, le contexte de l'informatique est assez particulier et totalement adapté à l'agilité (avec des cycles de développements courts et des technologies qui évoluent très rapidement). Cependant, les cas existent et sont de moins en moins marginaux.

Airbus est un bon exemple de cette nouvelle approche avec la mise en place de ses « sprints innovation » dont l'objectif est de développer des projets d'innovation en 3 à 4 mois seulement. D'autres projets de moins grande envergure et plus industriels sont également sur un modèle de construction agile, comme WikiSpeed qui adapte l'agilité à la construction automobile.

Ces différents exemples montrent qu'il est possible d'appliquer l'agilité à d'autres domaines. Et il n'y a rien d'étonnant à cela dans la mesure où ces méthodes agiles ont pour origine des préceptes relativement génériques.

Définir l'agilité

Imaginons un instant que vous cherchiez à créer votre propre méthode agile ou à simplement vérifier que Scrum répond bien au « cahier des charges » imposé par le concept d'agilité. Quelle base fiable pouvons-nous utiliser pour confronter la définition et la mise en œuvre de ces méthodes ?

Et bien en définitive, c'est très simple. L'agilité fait globalement consensus et il n'existe pas de mouvance ou de vague particulière, uniquement le manifeste fondateur, le Manifeste agile. Celui-ci est disponible sur le site <http://agilemanifesto.org/> et définit les quatre préceptes de base ainsi que les douze principes à suivre pour être agile.

Les quatre préceptes...

Le Manifeste agile oriente la gestion de projet autour de quatre préceptes simples. Chose intéressante à noter, ces préceptes ne sont en aucun cas directifs. Ils sont présentés sous forme de préconisations, sans jamais remettre en cause ce qui peut se faire habituellement.

1 – Les individus et leurs interactions, plus que les processus et les outils

Il s'agit ici de mettre la communication et l'humain au centre de la démarche. La création d'un logiciel n'est pas à la charge d'équipes cloisonnées par des expertises, elle se nourrit des échanges constructifs entre tous les acteurs du projet.

2 – Des logiciels opérationnels, plus qu'une documentation exhaustive

Il s'agit de focaliser la plus grande partie du temps sur la production et l'outil. Attention, il n'est pas question de remettre en cause la documentation et son utilité, juste de la rendre plus rationnelle, plus légère, moins chronophage et plus simple à faire évoluer.

3 – La collaboration avec les clients, plus que la négociation contractuelle

L'implication du client est un prérequis fort qui assure la réussite du projet. En effet, le client fait partie de l'organisation de manière active et il est impliqué dans toutes les phases de création de son produit. La relation contractuelle ne doit pas être remise en cause mais simplement adaptée à ce dispositif : redistribution des responsabilités, nécessité d'investissement de l'ensemble des acteurs (client compris), etc.

4 – L'adaptation au changement, plus que le suivi d'un plan

C'est dans l'adaptation d'un logiciel au fil de l'eau que l'agilité prend tout son sens. Cependant, il est important de comprendre que c'est également l'un des points les plus complexes. Être agile ne signifie pas appliquer tous les changements sans discernement dès que le client ou les utilisateurs le souhaitent. Le plan initial doit servir de ligne directrice, l'agilité permettant ensuite d'ajuster le projet au fur et à mesure des développements et d'être proactif sur les futurs besoins de l'application.

... déclinés en douze principes

Les quatre préceptes de base du Manifeste agile sont déclinés en douze grands principes qui apportent des précisions sur la manière de produire agile. Le tableau 1-1 reprend ces principes, auxquels nous avons ajouté ce qui, selon nous, est essentiel dans l'application de la démarche.

Tableau 1-1 Les douze principes de l'agilité

	Principe	À retenir
1	<i>Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.</i>	Satisfaire l'utilisateur
2	<i>Accueillez positivement les changements de besoins, même tard dans le projet. Les processus agiles exploitent le changement pour donner un avantage compétitif au client.</i>	Accepter le changement
3	<i>Livrez fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.</i>	Livrer fréquemment

Tableau 1-1 Les douze principes de l'agilité (suite)

Principe	À retenir
4 <i>Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.</i>	Travailler en synergie
5 <i>Réalisez les projets avec des personnes motivées. Fournissez-leur l'environnement et le soutien dont ils ont besoin et faites-leur confiance pour atteindre les objectifs fixés.</i>	Stimuler la motivation
6 <i>La méthode la plus simple et la plus efficace pour transmettre de l'information à l'équipe de développement et à l'intérieur de celle-ci est le dialogue en face-à-face.</i>	Communiquer en direct avec les opérationnels
7 <i>Un logiciel opérationnel est la principale mesure d'avancement.</i>	Un seul indicateur : les fonctionnalités implémentées
8 <i>Les processus agiles encouragent un rythme de développement soutenable. Ensemble, les commanditaires, les développeurs et les utilisateurs devraient être capables de maintenir indéfiniment un rythme constant.</i>	Bannir les <i>rushs</i> de production
9 <i>Une attention continue à l'excellence technique et à une bonne conception renforce l'agilité.</i>	Ne pas négliger la qualité de production
10 <i>La simplicité – c'est-à-dire l'art de minimiser la quantité de travail inutile – est essentielle.</i>	Rester concentré sur l'essentiel : mettre en place un produit
11 <i>Les meilleures architectures, spécifications et conceptions émergent d'équipes auto-organisées.</i>	Favoriser une certaine autonomie des équipes
12 <i>À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence.</i>	Intégrer la notion d'amélioration continue tout au long du projet

En prenant connaissance du contenu de ce manifeste, certains s'étonneront sûrement de l'absence d'un guide ou d'outils de mise en œuvre. En réalité, comme nous l'avons évoqué précédemment, l'essence même de ce manifeste n'est pas de proposer un « mode opératoire ».

L'agilité propose une orientation, une philosophie de travail organisée sur la base de règles simples. Ces règles offrent un cadre suivi partiellement ou entièrement par l'ensemble des méthodes qui gravitent autour de ces valeurs. Il serait ainsi légitime de se demander si, en dehors de l'agilité, les méthodes classiques ont ou non des valeurs partagées.

Agiles vs classiques

Deux approches opposées

Il est fréquent de mettre en opposition les méthodes agiles à celles qui sont plus communément appelées « méthodes classiques » ou « traditionnelles ». Derrière ces termes se cache en réalité un immense fourre-tout dans lequel on regroupe les méthodes qui ne répondent pas aux exigences de l'agilité.

Cette approche plus traditionnelle de la gestion de projet se cale directement sur les modèles « industriels ». Elle s'impose avec le développement de la production informatique dans les années 1960-1970 et notamment SDM (*Software Development Methodology*). La création d'applications est alors inscrite dans un processus séquentiel et linéaire. Elle part d'un besoin ferme et aboutit à la mise en place de LA solution technique correspondante, sans possibilité de modification en cours de développement.

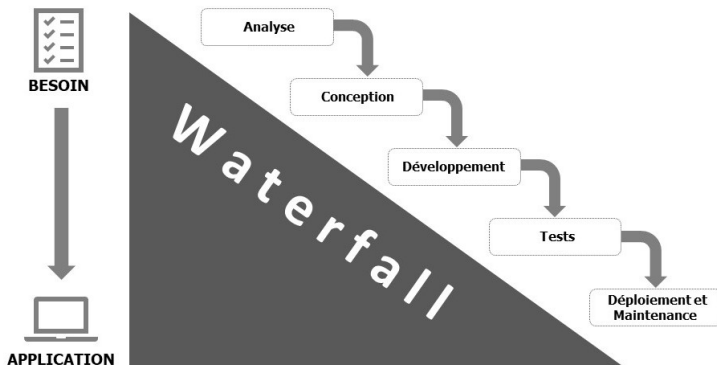
C'est principalement cette rigidité du processus de production qui entre en opposition avec les concepts agiles. On parle plus couramment d'une opposition de style entre deux approches :

- **Approche prédictive** des méthodes classiques qui se basent sur une expression figée du besoin. Elle aboutit à la création d'un logiciel développé d'un seul jet. Comme son nom l'indique, tout est prévu, rien n'est laissé au hasard !
- **Approche adaptative** des méthodes agiles qui acceptent l'évolution du besoin en cours de développement et propose la mise en place de l'application par briques incrémentales et itératives.

En outre, on utilise fréquemment deux représentations lorsqu'il s'agit d'illustrer les démarches liées aux méthodes classiques :

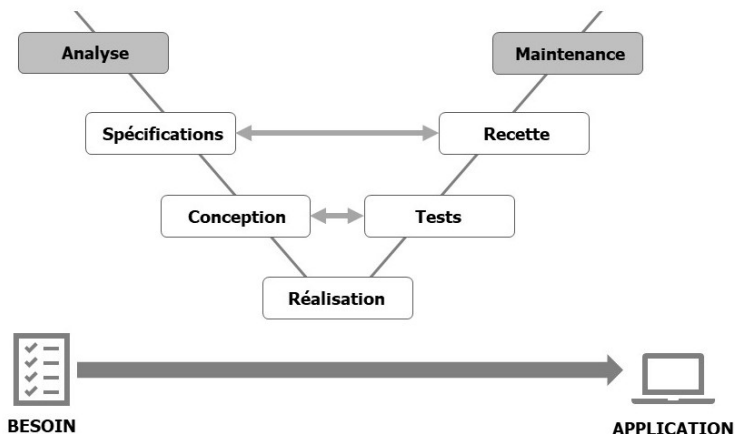
- Le **waterfall** (cascade) qui met bien en avant le processus linéaire de création logicielle traditionnel. Il propose une succession d'étapes simples allant du recueil des besoins jusqu'au déploiement et à la maintenance du produit.

Figure 1-1
Le waterfall



- Le cycle en V, plus élaboré, peut être considéré comme le véritable standard d'organisation pour les méthodes classiques. Sur cette représentation, les étapes clés sont mises en relation 2 par 2. Ces relations marquent un lien de causalité entre les étapes : par exemple, l'étape de recette se base nécessairement sur les spécifications.

Figure 1-2
Le cycle en V



Principales différences

Nous avons vu que les méthodes agiles étaient régies par un manifeste proposant douze grands principes. Il est intéressant de mettre en opposition ces principes avec ceux des méthodes classiques.

Le tableau 1-2 présente les principales différences entre les approches classiques et les approches traditionnelles. Ces différences sont organisées sur la base des 12 thèmes du manifeste (principes agiles) décrits précédemment.

Tableau 1-2 Méthodes : agiles vs classiques

	MÉTHODES AGILES	MÉTHODES CLASSIQUES
Objectif	Satisfaire l'utilisateur	Respecter le besoin initial et les engagements
Changement	Accepter le changement	Opposé au changement ou, en tout cas, moins enclin à l'accepter compte tenu des livraisons tardives et des processus de gestion lourds
Livraison	Livrer fréquemment	Livrer en une seule fois une application « finalisée »
Équipe	Travailler en synergie	Travailler de façon segmentée (chacun voit sa partie du travail)
Moteur	Stimuler la motivation	Stimuler la productivité
Communication	Communiquer en direct avec les opérationnels	Communiquer de façon verticale en passant par des relais hiérarchiques (par exemple, le chef de projet MOE est relayé entre la MOA et ses développeurs)
Indicateurs	Un seul indicateur : les fonctionnalités implémentées	Justifier par les indicateurs (sans livraison intermédiaire, les indicateurs sont les seuls justificatifs de l'avancement, des écarts, etc.)

Tableau 1-2 Méthodes : agiles vs classiques (suite)

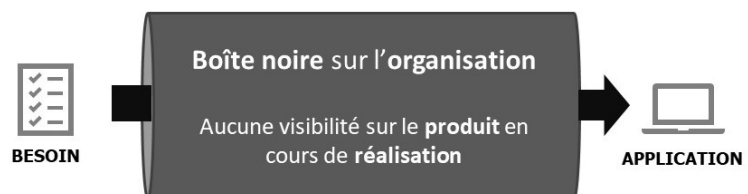
	MÉTHODES AGILES	MÉTHODES CLASSIQUES
Rythme	Bannir les <i>rushs</i> de production	Adapter la production aux contraintes projets
Qualité	<i>Que ce soit en méthode agile ou classique, l'excellence technique est essentielle. Ce neuvième principe sert principalement à justifier le fait que les méthodes agiles prônent la qualité des livrables techniques (contrairement à ce que peuvent en dire un certain nombre de détracteurs)</i>	
Livrables	Rester concentré sur l'essentiel : la production	Une documentation précise est essentielle pour assurer les échanges et la validation autour des besoins client
Autonomie	Favoriser une certaine autonomie des équipes	Encadrer scrupuleusement le travail des équipes
Amélioration	Intégrer la notion d'amélioration continue tout au long du projet	Introspection possible mais uniquement en fin de projet

Comme on peut le voir, les deux types de méthodes diffèrent principalement par leur approche : une approche orientée besoin pour les méthodes classiques (prédictif) et orientée satisfaction des acteurs du projet pour l'agilité (adaptatif). Néanmoins, il est important de noter qu'un des thèmes fait consensus : la qualité. Il est logique pour n'importe quelle méthode de proposer une approche qualité indiscutable. Les approches plus traditionnelles ne sont pas mauvaises, soyons clair sur ce point. Elles proposent simplement un paradigme organisationnel plus rigide adapté à certains types de projets.

Pourquoi être si sectaire ?

Au final, ce qui est le plus fréquemment reproché à l'utilisation des méthodes classiques tient en deux mots : effet tunnel. Pour résumer ce phénomène, l'équipe de réalisation échange avec le métier uniquement au moment des spécifications initiales, puis démarre ensuite une période plus ou moins longue de développements. Pendant cette période, le commanditaire et les utilisateurs ont peu de visibilité sur le produit et le découvrent une fois l'ensemble des développements terminés. Il n'est alors pas rare d'entendre ce commentaire : « C'est effectivement ce que j'ai demandé, mais ce n'est plus exactement ce que je veux. »

Figure 1-3
L'effet tunnel



De leur côté, les détracteurs des méthodes agiles ont des arguments qui, à l'inverse, vont tendre vers le manque de rigueur et le côté « aléatoire » qui se dégagent d'une telle organisation.

Avec le recul (et une bonne utilisation), il est évident aujourd'hui que les méthodes issues de l'agilité n'ont rien d'une bouillie organisationnelle sans queue ni tête. Cependant, les méthodes classiques ne sont pas forcément toutes mauvaises. L'approche prédictive a pour avantage d'apporter une grande rigueur scientifique, indispensable à la mise en place de systèmes hautement critiques. C'est le cas par exemple des dispositifs complexes liés au nucléaire, l'aéronautique, l'aérospatial, le médical, etc.

Bien évidemment, les projets informatiques directement liés à ces métiers pourraient suivre une logique agile. Seulement, il semble évident que remettre en question un besoin sur ce type de projet représente un risque très important au niveau de la sécurité et du coût. Une modification du cahier des charges peut dans certains cas entraîner des vérifications longues, complexes et très chères. En effet, le risque n'est pas le même si on compare un site de rencontres et un logiciel d'autopilote destiné à l'aéronautique.

En d'autres termes, mettre en opposition ces deux types de méthodes sans prendre en compte le contexte n'a pas de véritable intérêt. Aujourd'hui, les critiques observées à l'encontre des méthodes classiques concernent surtout leur utilisation : elles s'avèrent bien souvent inadaptées aux projets sur lesquels elles sont appliquées. Ainsi, et comme souvent, ce n'est pas l'outil qui est mauvais mais plus l'usage qui en est fait.

Une multitude de « méthodes »

Méthodes ou cadres de travail

Si le Manifeste agile met à disposition des préceptes et principes à respecter, il constitue aujourd'hui le terreau d'un très grand nombre de méthodes et de cadres de travail. Il convient de bien distinguer les deux, dans le sens où même Scrum ne se définit pas comme une méthode à part entière.

Ainsi, une méthode va spécifier un processus global à appliquer de façon stricte et structurée, de l'expression de besoin à la mise en service de l'application. Le cadre de travail (*framework*), quant à lui, propose une boîte à outils sur tout ou partie des composantes d'un projet : l'équipe, les événements, les outils et artefacts, etc. Elle nécessite la mise en place d'une méthodologie adaptée (et adaptable, d'où l'avantage en termes d'agilité).

Les plus populaires

Lorsqu'on parle d'agilité, on fait allusion le plus souvent aux méthodes et cadres de travail qui en découlent. On retrouve parmi ces méthodes RAD, DSDM, FDD, Crystal, eXtreme Programming et bien sûr Scrum.

Comme nous l'avons précisé, beaucoup de ces méthodes sont nées avant la ratification du Manifeste agile. Ce dernier a permis de formaliser un tronc commun qui a ensuite servi de base pour la création de nouvelles méthodes et l'amélioration de méthodes existantes.

Si ces méthodes sont basées sur un socle commun, elles n'en restent pas moins différentes sur bien des aspects : degré de précision, approche, documents supports, types d'acteurs, etc. Dans les sections suivantes, nous vous présentons un récapitulatif des spécificités des principales méthodes agiles. Scrum sera quant à lui détaillé par la suite.

RAD

La méthode RAD (*Rapid Application Development*) est considérée comme étant la première méthode agile. Elle est publiée par James Martin en 1991 et reprend de nombreux travaux orientés autour des cycles en spirales et du développement incrémental.

RAD a posé les bases des techniques de réalisation propres aux méthodes agiles. Elle propose un mode de développement dit « semi-itératif » : seule la phase de développement comprend des « itérations » permettant des ajustements à la marge, l'expression de besoin et la conception étant déterminés en début de projet.

DSDM

La méthode DSDM (*Dynamic Systems Development Method*) a été mise en place au milieu des années 1990 en Grande-Bretagne. L'idée proposée par un regroupement de sociétés est alors de professionnaliser et de structurer les pratiques liées à la méthode RAD.

Par rapport à RAD, l'apport principal de DSDM porte sur deux concepts complémentaires : les neuf principes du DSDM et un processus de réalisation spécifique.

FDD

La méthode FDD (*Feature Driven Development*) est spécifiée en 1997 par Jeff De Luca et propose une gestion de projet pilotée par les fonctionnalités à forte valeur ajoutée pour les utilisateurs (d'où son nom). Il s'agit donc de mettre en place des cycles itératifs relativement courts et permettant de proposer des fonctionnalités tangibles.

FDD définit pour chaque itération un processus en cinq étapes allant de la conception à la réalisation. On parle ici d'itération (et non de semi-itération) puisque chacune d'elles comporte sa propre étape de conception, accompagnée de la réalisation d'un ensemble de fonctionnalités.

Crystal

Crystal n'est pas à proprement parler une méthode mais plutôt un ensemble de méthodes agiles. Le concept est développé par Alistair Cockburn en 2004 sur le principe suivant : chaque projet est différent, la méthode doit pouvoir s'adapter simplement à leurs spécificités.

On peut dire que la famille de méthodes Crystal est plutôt permissive (en tout cas très souple) et principalement basée sur sept principes : les « sept propriétés de sécurité ». Cockburn propose également un outil permettant de sélectionner la méthode Crystal la plus adaptée en fonction de deux critères : la taille de l'équipe et la criticité du projet.

Quelques exemples :

- la méthode spécifique Crystal Clear est principalement adaptée à des projets de six personnes ou moins et impose des cycles d'itérations de trois mois au maximum ;

- la méthode Crystal Orange, quant à elle, implique une équipe de 21 à 40 personnes et définit également sa propre série de livrables à produire.

eXtreme Programming

eXtreme Programming, ou XP, fait partie des méthodes les plus populaires depuis sa sortie en 1996. Ses créateurs Kent Beck, Ward Cunningham et Ron Jeffries sont partis d'un principe simple : pousser à l'extrême les bonnes pratiques de réalisation d'un projet. Parmi celles-ci, la plus populaire concerne le *pair programming* (un poste pour deux développeurs) permettant de pousser au paroxysme la pratique de revue de code. Cependant, on trouve également d'autres concepts comme celui de toujours choisir la solution la plus simple ou encore celui de faire des livraisons plusieurs fois par jour si nécessaire.

En plus de cette approche « extrême », XP se base sur cinq valeurs : la communication, la simplicité, les retours d'information, le courage et le respect.

Être agile avec les méthodes agiles

Les précédentes sections ne présentent pas les méthodes agiles de manière exhaustive, et pour cause. L'agilité regroupant essentiellement un ensemble de bonnes pratiques, beaucoup de professionnels ont mis en place leurs propres organisations projet basées sur le manifeste. On trouve généralement deux modes d'action de mise en place de ces méthodologies « personnalisées » : un mode « Big Bang » et un mode « adaptation/combinaison ».

Le premier est assez simple à comprendre. On prend les préceptes, et à partir de là, on entre dans une création totale en partant de 0. Il est alors possible d'adapter et d'optimiser finement sa méthode à ses impératifs de production, son métier, son organisation, etc. Dans la réalité, ce mode de fonctionnement est assez rarement appliqué et plutôt contraignant (en particulier pour une première expérience).

Cependant, beaucoup mettent en place un choix qui selon nous est plus pragmatique : l'adaptation de méthodes et/ou cadres existants. L'idée est de prendre une méthode de base et de l'ajuster à ses propres impératifs de production : on ajoute, on modifie, on supprime. En résumé, on adapte. Il est également intéressant de mixer les idées provenant d'autres méthodes pour éviter de réinventer la roue sur des concepts qui font leur preuve – ce sujet est un élément clé de notre approche dans la mise en place de Scrum.

DevOps, une évolution naturelle

Dev + Ops = DevOps

Deux mondes que tout oppose ?

Le terme DevOps a été utilisé pour la première fois en 2009 par Patrick Debois lors des premiers *devopsdays*. Cette série d'événements propose des conférences axées sur le thème de la synergie entre les développements et l'exploitation (ingénieurs système). Et c'est là tout le principe de DevOps, qui a pour vocation de réunir ces deux mondes.