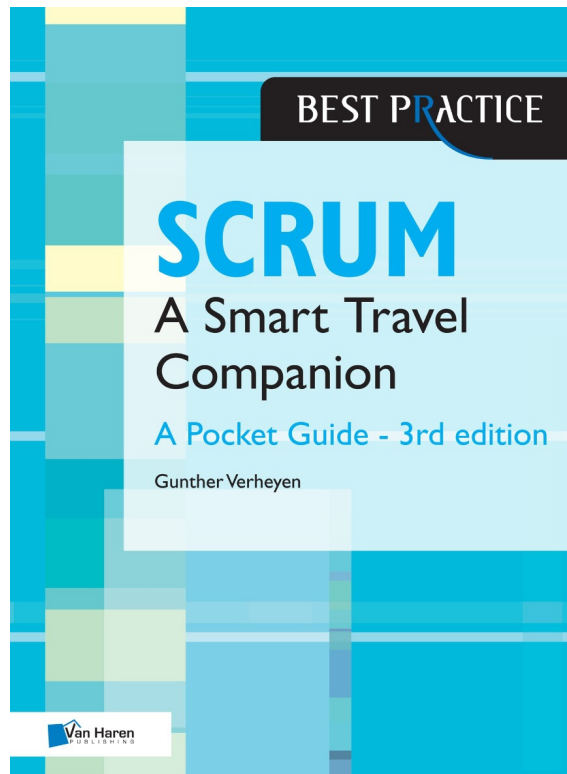


Scrum, un guide de poche¹

Gunther Verheyen



Scrum – A Pocket Guide – 3rd edition de Gunther Verheyen (2021)

La traduction française du best-seller de Gunther Verheyen sortira début 2022.

Traduction : Léo Davesne et François Bruneau.

Relecture : Christophe Gesché, Nedjma Saidani, Farouk Choulak, Guillaume Leone, Guillaume Deleplace et Sedera Randria.

Ci-après, en voici les trois premiers sous-chapitres pour le calendrier de l'avent d'Agile Christmas.

¹ Titre temporaire

1. Le Paradigme Agile

1.1 ÉVOLUER OU NE PAS ÉVOLUER

L'industrie du logiciel a été longtemps dominée par le paradigme des représentations et des croyances *industrielles* basées sur d'anciens processus et de vieilles théories de production. Un élément central dans cette étendue de connaissances, de points de vue et de pratiques, était la conviction taylorienne² que l'on ne peut pas faire confiance aux « travailleurs » pour accomplir leur travail de façon intelligente, autonome, et créative. On attend d'eux de réaliser uniquement des tâches pré-définies, qu'ils peuvent exécuter sans avoir à réfléchir. Leur travail doit être préparé, conçu et planifié par du personnel plus expérimenté. En plus de cela, des superviseurs hiérarchiques doivent surveiller scrupuleusement l'exécution de ces tâches méticuleusement préparées. La qualité est assurée en bout de chaîne par l'acceptation des bons lots de livrables et le rejet des mauvais. Les récompenses pécuniaires sont utilisées pour stimuler le comportement souhaité. Les comportements indésirables sont punis. *La bonne vieille stratégie de « la carotte et du bâton ».*

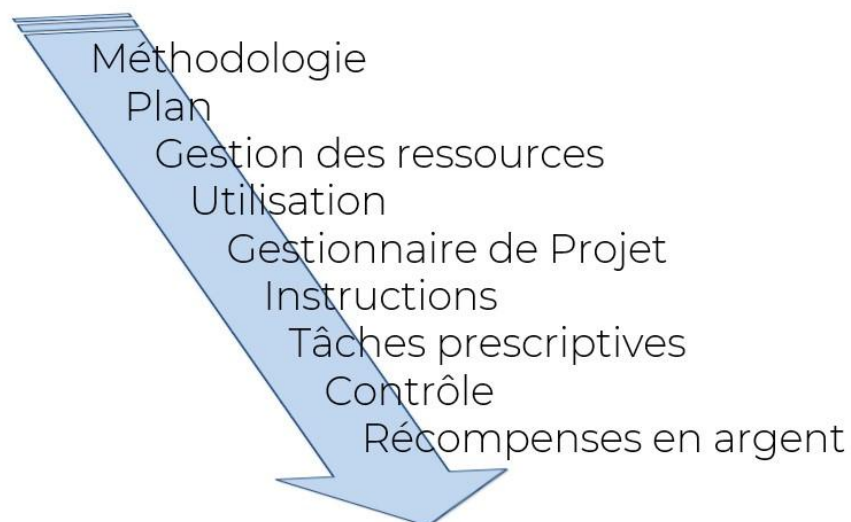


Figure 1.1 : Le paradigme industriel

² Frederick Taylor (1856-1915) était un ingénieur américain principalement connu pour ses travaux sur les moyens de maximiser la productivité et l'efficacité du travail tout en minimisant son coût. Il défendait la standardisation imposée et l'adoption forcée de méthodes et de pratiques systématiques. Selon lui, le contrôle appartient exclusivement au management, et les travailleurs sont présents uniquement pour exécuter le travail.

Les failles considérables de l'ancien paradigme appliqué au développement de logiciels sont connues et bien documentées. En particulier, les « Chaos reports » du Standish Group [Standish, 2011 ; Standish, 2013] ont à maintes reprises révélé les faibles taux de succès d'une approche traditionnelle dans le développement de logiciels. Bon nombre de lacunes et d'erreurs qui en résultent vont bien au-delà de niveaux de tolérance acceptables. La réaction générale, fort regrettable, semble avoir été de réviser les attentes à la baisse. La définition du « succès » dans le paradigme industriel est faite de la combinaison du respect des délais et du budget tout en incluant l'ensemble du périmètre. On a fini par accepter que seulement 10 à 20 % des projets informatiques soient couronnés de succès. *Bien que ces critères de succès puissent être contestés, c'est la promesse de ce paradigme.* Il est devenu communément admis que le résultat soit de mauvaise qualité et que plus de 50 % des fonctionnalités des applications logicielles développées de manière traditionnelle ne soient jamais utilisées [Standish, 2002 ; Standish, 2013].

Bien que ce ne soit pas consciemment et largement admis, le paradigme industriel a plongé le monde du logiciel dans une crise très grave. Beaucoup ont tenté de dépasser cette crise en renforçant la démarche industrielle. On a cherché à rendre le travail préparatoire encore plus exhaustif. On a créé davantage de plans, prévu davantage de phases, produit davantage de schémas, on a réalisé davantage de travail en amont, en espérant que le travail lui-même serait exécuté de façon plus efficace. *Comme les taux de réussite n'augmentaient pas, la pensée industrielle en a déduit que les instructions n'étaient pas assez claires et détaillées.* Mais l'idée de base selon laquelle les « travailleurs » avaient besoin d'être dirigés est restée. La supervision fut augmentée et intensifiée. Des instructions encore plus détaillées furent transmises.

Pourtant, peu d'améliorations ont suivi. Beaucoup de défauts, d'imperfections ainsi qu'une piètre qualité ont subsisté et ont dû être tolérés.

Cela a pris du temps mais, inévitablement, en observant les anomalies majeures du paradigme industriel, de nouvelles idées et de nouvelles perspectives ont commencé à se former.

Les graines d'un nouveau monde étaient déjà semées dans les années 1990. Mais c'est en 2001 qu'elles amenèrent à la formalisation du mot « Agile », un moment charnière dans l'histoire du développement informatique. Un nouveau paradigme était né dans le monde du logiciel. C'est un paradigme qui tire parti des heuristiques et de la créativité, ainsi que du respect (rétabli) envers la nature créative du travail et envers l'intelligence des « travailleurs ». Depuis, celui-ci s'étend à de nombreux autres domaines de la société.

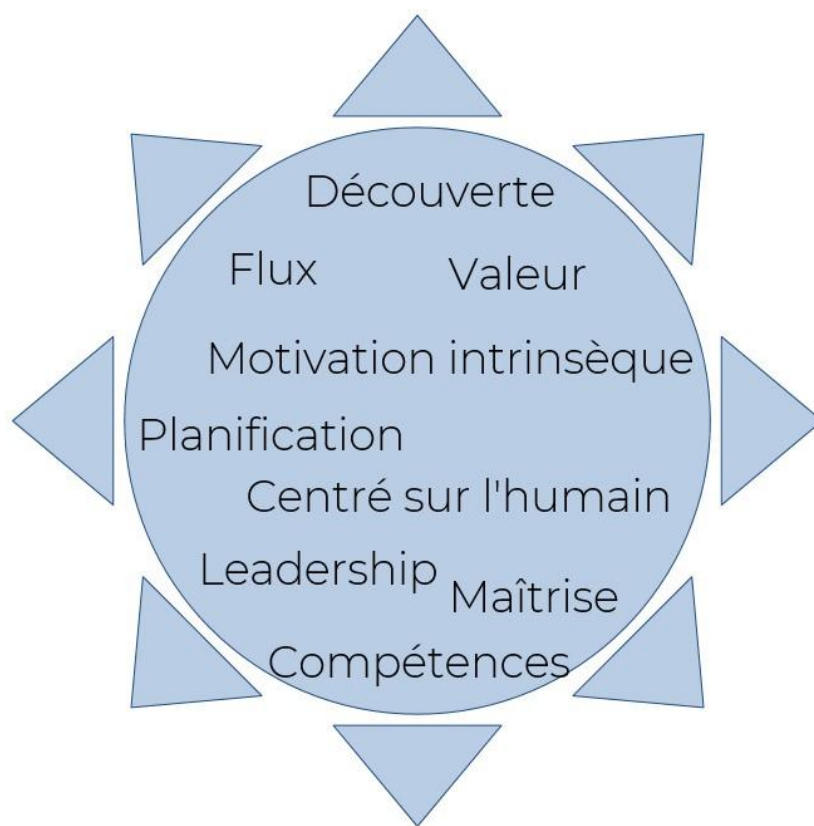


Figure 1.2 : Le paradigme Agile

L'industrie du logiciel a de bonnes raisons de continuer sa mutation vers le nouveau paradigme. Les failles de l'approche existante sont significatives et bien connues alors que la présence de logiciels dans notre société augmente exponentiellement, et que ceux-ci sont devenus essentiels au bon fonctionnement du monde moderne. Toutefois, par définition, le passage à un nouveau paradigme prend du temps. Et l'ancien paradigme semble enraciné très profondément et doué d'une persistance considérable. L'approche industrielle du développement logiciel continue d'être enseignée et défendue comme étant la plus appropriée.

Beaucoup disent que l'Agilité est trop radicale et préconisent en conséquence une introduction graduelle des pratiques Agiles dans le cadre traditionnel existant. Pourtant, il y a des raisons d'être très sceptique quant à une telle évolution, une lente progression de l'ancien vers le nouveau paradigme, du cycle en cascade dit « Waterfall » à l'Agilité.

Il y a de fortes chances qu'une évolution graduelle n'ira jamais plus loin que la surface, et ne fera rien de plus que d'effleurer celle-ci. De nouveaux noms seront utilisés, de nouveaux termes et de nouvelles pratiques seront imposés, mais les fondements de la pensée et du comportement resteront les mêmes. Les failles essentielles resteront intactes ; en particulier le mépris de l'individu, ce qui mène à

continuer de traiter les personnes créatives et intelligentes comme des « travailleurs » stupides, comme des « ressources ».

Préserver les fondements traditionnels signifie que l'on conserve les données existantes, les indicateurs et les normes en place. Le nouveau paradigme sera donc jugé par rapport à ces derniers. Toutefois, des paradigmes distincts de par leur nature sont composés de concepts et d'idées fondamentalement différents, le plus souvent antinomiques. Il n'est pas sensé de comparer les paradigmes industriel et Agile. Il faut de l'honnêteté intellectuelle pour accepter les graves défauts des anciennes pratiques. Il faut du leadership, de la vision, de l'esprit d'entreprise et de la persévérance pour adopter les nouvelles pratiques, abandonnant ainsi la pensée traditionnelle.

Une évolution graduelle est de fait une situation de statu quo qui maintient le paradigme industriel intact.

Il existe des preuves accablantes que l'ancien paradigme ne fonctionne pas. L'essentiel des retours d'expérience sur les meilleurs résultats de l'Agilité ont longtemps été anecdotiques, isolés ou relativement mineurs. En 2011, le « Chaos report » du Standish Group [Standish, 2011] a marqué un tournant, en apportant pour la première fois des résultats de recherche clairs qui furent confirmés dans tous les « Chaos reports » suivants. Des recherches approfondies ont été menées pour comparer des projets traditionnels avec des projets utilisant des méthodes Agiles. Le rapport montre qu'une approche Agile a un rendement plus élevé, y compris selon les attentes classiques de livraison dans les temps, dans le budget initial et avec toutes les fonctionnalités promises. Le rapport montre que les projets Agiles ont réussi trois fois plus souvent et qu'il y avait trois fois moins d'échecs pour les projets Agiles que pour les projets traditionnels. Pour les très gros projets, les résultats ont été moins criants, ce qui est probablement dû à des attentes erronées au départ, c'est-à-dire la combinaison délais + budget + périmètre. Si on le mesurait par rapport à des attentes justes, qui mettent l'accent sur une collaboration active avec le client et une livraison fréquente de valeur, le nouveau paradigme serait encore plus performant, notamment avec la livraison régulière de tranches verticales de valeur, qui permet de dépasser le problème de volume inhérent aux très gros projets.

Néanmoins, l'Agilité est un choix, pas une obligation. C'est une manière d'améliorer l'industrie du logiciel. La recherche montre que l'Agilité offre de meilleurs résultats.

! Scrum soutient l'Agilité.

Scrum est une manière tangible d'adopter et d'enraciner l'Agilité. Les règles claires de Scrum aident à comprendre le nouveau paradigme. Le nombre restreint de prescriptions invite à l'action immédiate et entraîne une assimilation plus fructueuse et plus durable du nouveau paradigme. En utilisant Scrum, les personnes développent de nouvelles manières de travailler ; par la découverte, l'apprentissage basé sur l'expérimentation et la collaboration. Ils développent une nouvelle manière d'être, et entrent dans un état d'*agilité*. Ce processus aide également leurs organisations à se transformer vers un tel *état* d'agilité ; un état de changement constant, de fluidité, d'évolution et d'adaptation. Il permet de libérer du temps, des personnes et de l'énergie pour être (de nouveau) innovant.

Cependant, malgré son minimalisme, l'expérience montre qu'adopter Scrum représente souvent un bond de géant. C'est peut-être à cause des incertitudes qui vont avec l'abandon des vieilles certitudes, même si ces dernières ne se sont avérées ni particulièrement fiables ni particulièrement... certaines. C'est peut-être à cause du temps nécessaire pour opérer un vrai changement. C'est peut-être à cause de la détermination et du travail que cela implique. On observe, encore et toujours, que Scrum est simple, mais pas facile.

1.2 LES ORIGINES DE L'AGILITÉ

Malgré la prépondérance de concepts industriels basés sur la planification, une approche évolutive du développement de logiciels n'est pas nouvelle. Craig Larman a décrit en détail les prédécesseurs historiques de l'Agilité dans son livre « Agile & Iterative Development, A Manager's Guide » [Larman, 2004].

Cependant, le terme officiel « Agile » date de février 2001 quand 17 leaders du développement de logiciels se sont rassemblés à Snowbird, une station de ski en Utah aux États-Unis. Leurs discussions tournaient autour de leurs perspectives sur le développement de logiciels à une époque où les approches en cascade défaillantes étaient remplacées par des implémentations lourdes de RUP (« Rational Unified Process »), lesquelles n'avaient pas mené à de meilleurs résultats que les processus traditionnels. Ces pontes du développement suivaient différentes voies et méthodes, chacune étant une expression distincte de ce qui deviendrait le nouveau paradigme Agile ; Scrum, eXtreme Programming, Adaptive Software Development, Crystal, Feature Driven Development, etc.

Ce rassemblement a mené à donner le label « Agile » aux principes, croyances et raisonnements que ces leaders et leurs méthodes avaient en commun. Ils ont été publiés en tant que « Manifeste pour le développement Agile de logiciels » [Beck, et.al., 2001].

On entend souvent parler du désir de « faire de l'Agilité » ou de « se mettre à l'Agilité ». Et bien trop souvent, c'est le désir d'une solution magique, un remède miracle qui résoudrait tous les problèmes. Cela me fait souvent dire que « *l'Agilité n'existe pas* ». L'Agilité n'est pas un processus, une méthode ou une pratique uniques et définitives. L'Agilité est l'ensemble de principes que les méthodes pour le développement Agile de logiciels ont en commun. L'Agilité se réfère à l'état d'esprit, aux convictions et aux préférences exprimées dans le Manifeste pour le développement Agile de logiciels.

Nous découvrons comment mieux développer des logiciels par la pratique et en aidant les autres à le faire.

Ces expériences nous ont amenés à valoriser :

Les individus et leurs interactions plus que les processus et les outils

Des logiciels opérationnels plus qu'une documentation exhaustive

La collaboration avec les clients plus que la négociation contractuelle

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

Nous reconnaissons la valeur des seconds éléments, mais privilégions les premiers.

Figure 1.3 : Le Manifeste pour le développement Agile de logiciels

Le Manifeste aide à comprendre les idées sous-jacentes de l'Agilité. Si vous l'utilisez comme référence afin d'en acquérir une compréhension plus profonde, alors je vous recommande fortement de lire les douze principes qui se situent derrière les quatre énoncés de valeurs, voir <http://agilemanifesto.org/iso/fr/principles.html>.

1.3 DÉFINITION DE L'AGILITÉ

En l'absence d'une définition concise et précise, je préfère décrire « l'Agilité » en des termes qui correspondent à trois caractéristiques clés. Il s'agit des aspects qui sont communs au portfolio des méthodes Agiles et sont typiques d'une manière Agile de travailler :

- ❖ Centré sur l'humain ;
- ❖ Un processus itératif et incrémental ;
- ❖ La valeur comme mesure de succès.

1.3.1 Centré sur l'humain

L'Agilité n'est pas guidée par un plan prédictif de la façon d'implémenter des exigences qui auraient été exhaustivement analysées, conçues et architecturées au préalable. L'Agilité reconnaît que les spécifications ne peuvent pas être prédites en amont dans leurs moindres détails.

L'Agilité n'est pas non plus un processus qui consiste à confier toutes sortes de livrables intermédiaires à différents départements spécialisés où chaque département accomplirait son travail de son côté.

L'Agilité est axée sur la *collaboration* continue de personnes issues de tous les départements qui doivent être impliqués ; qu'il s'agisse du business, de l'IT, du marketing, des ventes, des RH, du service client, de l'opérationnel ou de la direction. L'Agilité ne reconnaît certainement pas la bonne vieille mésentente entre le business et l'IT. Les deux sont nécessaires pour répondre avec succès au besoin de créer des produits à la fois utilisables *et* utiles — des logiciels qui apportent de la valeur.

L'accent mis sur la collaboration, les interactions et les conversations nécessitent un style de management différent. Les équipes Agiles sont *facilitées* grâce au leadership serviteur. Un cadre et un contexte pour l'auto-gestion sont donnés, sur la base desquels les équipes reçoivent des objectifs et une direction. Un contrôle subtil émerge de ce cadre, contrôle qui ne concerne pas les individus, les tâches ou les estimations. La collaboration et la facilitation remplacent les mécanismes traditionnels qui consistent à diriger et à contrôler en donnant chaque jour aux individus une liste de micro-tâches à effectuer, sur un mode d'autorité totalitaire et de contrôle invasif.

Cela montre à quel point les personnes sont respectées pour leur créativité, leur intelligence et leur faculté à s'auto-organiser. Les personnes sont respectées pour leur capacité à comprendre et traiter un problème sans être submergées de

cérémonies et de bureaucratie. Une telle surcharge ne fait que tuer la réflexion collaborative, l'innovation et les responsabilités de chacun, en les remplaçant par de la bureaucratie, de la paperasse, des rapports intermédiaires, ou des justifications administratives.

Les individus sont respectés dans le temps qu'ils sont à même d'investir dans leur travail au travers du concept de *Rythme Soutenable*. Le travail est organisé de façon à ce que le tempo puisse être maintenu, et ce, indéfiniment.

1.3.2 Un processus itératif et incrémental

Les processus Agiles ne sont pas des approches dites « à l'arrache », libres de toutes contraintes. Les processus Agiles sont *définis* et requièrent beaucoup de *discipline*.

Les produits sont créés morceau par morceau (« incrémentalement »), chaque morceau étant fait d'ajouts, d'améliorations, de suppressions ou de modifications à ce qui a été élaboré précédemment. Les éléments ainsi réalisés et le produit résultant dans son entièreté sont régulièrement revisités (« itérativement ») afin d'en assurer l'intégrité globale.

L'Agilité demande clairement une attention de tous les acteurs à la qualité et à l'excellence. L'Agilité remplace l'idée qu'il suffise d'incarner ces dernières dans des documents et des descriptions papier — lesquelles ont des propriétés bien différentes que le résultat final espéré, à savoir le produit ou le service livrable en lui-même.

La nécessité d'un processus itératif et incrémental est renforcée par le constat que quelles que soient la quantité de temps, d'énergie et d'argent investie pour les prédire et les planifier en amont, les besoins et leur implémentation sont amenés à changer. Les marchés et la compétition évoluent, les utilisateurs ne savent ce qu'ils veulent qu'une fois le produit dans leurs mains, les stratégies d'entreprises changent, pour ne citer que quelques exemples. Tout cela appelle à une attention et ouverture extrêmes au changement.

Contrairement à un processus prédictif, le changement n'est pas exclu du processus Agile ni repoussé aux formalités tardives du développement. Les nouveaux éclairages, l'évolution des opinions et les priorités fluctuantes forment le cœur vivant de l'Agilité. L'Agilité prospère sur *l'émergence* ; l'apparition et l'évolution graduelle et continue des demandes, des plans, des idées, des architectures et des designs. Le changement n'est plus disruptif parce qu'il fait maintenant partie de la façon naturelle de travailler. L'Agilité *favorise* même le changement comme une source d'innovation et d'amélioration.

1.3.3 La valeur comme mesure de succès

Dans les environnements qui opèrent sur un mode d'émergence, l'avancement ne peut pas être mesuré ni garanti sur la base de la simple conformité à des plans et des jalons définis en amont, des documents, des rapports, des signatures, des approbations ou toutes autres obligations formelles comme dans le paradigme industriel. L'Agilité instaure de nouvelles façons de mesurer l'avancement et le succès.

L'Agilité insiste sur le fait que la progression et la réussite ne peuvent être déterminées que par une fréquente inspection de *versions fonctionnelles* du produit (et non pas de descriptions intermédiaires de celui-ci) et de la *valeur* réelle que le produit génère.

Lors du développement d'un produit, il est naturel que les gens amenés à l'utiliser ne puissent évaluer son utilité et sa praticité que lorsqu'ils l'ont véritablement dans les mains. Aucune documentation papier ni procédé virtuel ne peut remplacer cela. C'est une invitation à un processus itératif et incrémental, le but étant de fermer régulièrement la boucle de rétroaction avec les utilisateurs et de mesurer l'impact et l'appréciation, sources d'information importantes pour les évolutions futures du produit.