***Introduction***

Dès les années 1990, des équipes de développeurs ont commencé à travailler avec des méthodes qui s’apparentaient déjà à ce que l’on appelle aujourd’hui « développement agile ». Jusqu’à la fin du 20e siècle, différents développeurs et équipes de développeurs se sont attachés à alléger le travail de programmation ; c’est pourquoi la méthode était décrite principalement par la **notion de légèreté (« lightweight »)**. C’est aussi à cette époque que sont apparues les méthodes Scrum et [Kanban](https://www.ionos.fr/digitalguide/sites-internet/developpement-web/kanban/), qu’on ne désignait pas encore comme développement de produit agile car l’expression n’avait pas encore été créée.

C’est en 2001 qu’eut lieu un tournant : lors d’une réunion aujourd’hui connue sous le nom de « Snowbird meeting », du nom d’une station de ski dans l’Utah où elle se tenait, dix-sept développeurs ont rédigé **le Manifeste agile**. Ils y rassemblaient leur expérience dans le développement de logiciels et la gestion d’équipes, proposaient des solutions, établissaient des principes, et rassemblaient le tout sous la notion de « développement agile », qui est devenue depuis emblématique des méthodes de travail modernes.

***Différence entre processus agile et planifiés***

Le développement agile, au contraire des méthodes plus classiques, se préoccupe bien plus du produit à réaliser que des principes et théories pour le réaliser. Le produit, l'application informatique le plus courant, voit très rapidement le jour. Puis elle est améliorée, perfectionnée selon les spécifications, attentes et les exigences qualité.  
La règle : " plutôt que chercher à prévoir ce que doit être le produit, on l'adapte et on s'adapte en permanence..."

**- la rigidité**

Les méthodes agiles se sont dotées de la capacité de délivrer très rapidement et très régulièrement des "produits" suffisamment aboutis pour que l'on puisse les évaluer. Fondées sur un dialogue permanent avec le client, les méthodes agiles visent la conformité quasi parfaite de la réalisation selon les attentes actuelles du client. Les réunions assurant les rencontres entre les clients, les utilisateurs et les prestataires sont au cœur même du projet.

* **Planifier à volonté**

Ainsi, la modification des spécifications en cours de réalisation, véritable tourment pour les développeurs des projets classiquement conduits, est le principe fondateur des démarches agiles. Les projets conduits selon une méthode de type agile ne sont pas étroitement encadrés par un planning rigide aux stricts impératifs. Au contraire, les méthodes agiles se dotent de la capacité non pas de chambouler les plans mais bien de re-planifier à volonté. Un seul objectif : délivrer très rapidement (des "sprints" de 2 semaines en moyenne) un nouveau produit "fini" prêt à être évalué et amélioré. Le produit livré sera la nouvelle référence des échanges et du projet.

* **Conditions de réussite**

Règle fondamentale : la méthode ne peut pas fonctionner sans une véritable relation de confiance entre le client et l'équipe de réalisation. La règle du jeu doit être parfaitement connue et les informations circuleront en toute transparence. Ce sont deux principes incontournables. Du point de vue du prestataire, il n'est pas si aisé que cela d'assurer la mutation des méthodes classiques vers les méthodes agiles. Le mode de management est totalement différent. Il ne s'agit plus de se contenter de la maîtrise des référentiels de la profession ([CMMI, ITIL, ISO...](https://www.piloter.org/gouvernance/methode-gouvernance-si.htm)), et de déléguer des personnels rapidement formés.

Les démarches agiles nécessitent des développeurs à la hauteur et un chef de projet professionnel suffisamment averti pour appliquer les pratiques et assurer le "rythme" adéquat

***Les principales méthodes agiles***

* **Scrum**

Aujourd’hui « **Scrum**» est la méthode agile la plus populaire. Ce terme signifie « mêlée » au rugby. La méthode scrum s’appuie sur des « sprints » qui sont des espaces temps assez courts pouvant aller de quelques heures jusqu’à un mois. Généralement et de préférence un sprint s’étend sur deux semaines. À la fin de chaque sprint, l’équipe présente ce qu’elle a ajouté au produit. Scrum regroupe trois acteurs :

* **Le Product Owner**(ou « Directeur de produit ») : il communique les objectifs premiers des clients et utilisateurs finaux, coordonne l’implication des utilisateurs et des parties prenantes, et se coordonne lui-même avec les autres product owners pour assurer une cohérence.
* **Le Scrum Master**: membre de l’équipe, il a pour but d’optimiser la capacité de production de l’équipe. Pour se faire, le scrum master aide l’équipe à travailler de façon autonome tout en s’améliorant d’avantage.
* **L’équipe opérationnelle** (qui regroupe idéalement moins de dix personnes) : la particularité d’une équipe scrum est qu’elle est dépourvue de toute hiérarchie interne. Une équipe scrum est auto-organisée.

D’autres termes sont à connaître pour comprendre la méthode scrum:

* **Le product backlog**(carnet du produit) : ce document contient les exigences initiales dressées puis hiérarchisées avec le client en début de projet. Néanmoins il va évoluer tout au long de la durée du projet, en fonction des divers besoins du client.
* **Le sprint backlog**(carnet de sprint) : en chaque début de sprint, l’équipe définit un but. Puis lors de la réunion de sprint, l’équipe de développement choisit les éléments du carnet à réaliser. L’ensemble de ces éléments constitue alors le sprint backlog.
* **User story :** ce terme désigne les fonctionnalités décrites par le client.
* **La mêlée (scrum)**: c’est une réunion d’avancement organisée de manière quotidienne durant le sprint.
* **EXtreme Programming (XP)**

Cette méthode très réactive destinée à des petits ou moyens projets, permet de réduire les coûts du changement. Dans cette méthode, le client pilote le projet grâce à des cycles itératifs d’une à deux semaines. C’est lui qui choisit les fonctionnalités à implémenter. Il transmet ses exigences à l’équipe sous forme de scénario susceptible d’être implémenté en une itération.  
**Dans la méthode XP l’équipe se décompose en binômes** qui revoient régulièrement les codes. On parle de **responsabilité collective du code**, le but étant que chaque développeur soit capable d’intervenir n’importe où dans la structure interne du logiciel. Auquel s’ajoutent des**tests automatisés** mis en place afin de vérifier chacune des fonctionnalités demandées par le client et de garantir une bonne qualité du produit.

***La gestion de projet agile***

Il s’agit d’une méthode de gestion de projet révolutionnaire caractérisée par une démarche itérative et incrémentale. En d’autres termes, les projets agiles sont construits lentement, étape par étape. Dans la gestion de projet traditionnelle, on attend généralement d’avoir exécuté l’ensemble des étapes d’un projet avant de soumettre le produit final au client. Avec l’approche agile, les clients peuvent évaluer la qualité du projet en cours de réalisation et intervenir pour rectifier le tir au besoin. Ainsi, il risque moins d’y avoir des disparités entre les attentes des clients et le produit final. La gestion de projet traditionnelle est souvent très rigide, et toute modification au plan initial peut poser un casse-tête. Les projets agiles sont généralement beaucoup plus souples, collaboratifs et interactifs, et la plupart des changements ajoutent de la valeur aux projets.

Par rapport à l’ingénierie logicielle traditionnelle, le développement de logiciels agiles cible principalement des systèmes complexes et le développement de produits avec des caractéristiques dynamiques, non déterministes et non linéaires. Il est souvent difficile d’obtenir des estimations précises, des plans stables et des prévisions à l’avance, et leur confiance est probablement faible. Les praticiens agiles chercheront **à réduire l’acte de foi** nécessaire avant d’obtenir des preuves de valeur. Les exigences et la conception sont tenues pour émergents. De grandes spécifications à l’avance causeraient probablement beaucoup de déchets dans de tels cas, c’est-à-dire qu’ils ne sont pas économiquement solides. Ces arguments de base et ces expériences antérieures de l’industrie, tirés d’années de succès et d’échecs, ont contribué à façonner la faveur du **développement agile** en faveur du **développement adaptatif, itératif** et **évolutif.**