

Les méthodes Agiles

1.Approche classique vs Approche agile

1.1 L'approche classique

Dans une **approche classique** (comme le modèle en cascade ou en V), on commence par :

- Définir **tous les besoins** du client avant de commencer,
- Etablir un **planning fixe**,
- Suivre le projet **étape par étape**, sans retour en arrière.

Mais dans la réalité :

- Les **besoins changent souvent**,
- Le client **comprend mieux ce qu'il veut** en voyant le logiciel avancer,
- Les **retards** ou les **dépassements de budget** sont fréquents.

Une grande étude du **Standish Group (1994)** a montré que :

- 31 % des projets informatiques sont **abandonnés en cours**,
- 52 % sont **terminés avec retard et coûtent plus cher** que prévu,
- Seulement 16 % sont de **véritables réussites**.

En 2008, une nouvelle étude a montré une amélioration, mais encore **seulement 35 % de succès**. Les principales causes d'échec étaient :

- **Le manque d'implication du client**,
- **Les changements fréquents de besoins**.

1.2 L'approche agile

Les **méthodes agiles** ont été créées pour résoudre ces problèmes. Elles proposent une manière de travailler plus **souple, collaborative et réactive**.

Principes de base :

- **Le client participe** tout au long du projet.
- Le développement se fait **par étapes courtes (itérations)**.
- On **s'adapte aux changements** plutôt que de les éviter.

Avantages :

- Très flexible.
- Permet de livrer des versions fonctionnelles rapidement.
- Le client est satisfait, car il voit régulièrement les résultats.

Inconvénients :

- Il est parfois difficile d'estimer le coût et la durée totale.
- Demande une équipe impliquée et organisée.

👉 L'approche agile est surtout utilisée pour :

- Des projets **complexes** ou de **longue durée**,
- Des **besoins encore flous** ou évolutifs,
- Et quand le client **veut voir des résultats rapidement**.

1.3 Différence entre les deux approches

Approche classique	Approche agile
Plan fixe et rigide	Adaptation continue
Client peu impliqué	Client très présent
Documentation lourde	Documentation légère
Livraison en une fois	Livrages fréquentes
Difficile de changer	Changement accepté
Peu de visibilité avant la fin	Vision claire à chaque étape

👉 En résumé :

L'approche **classique** se concentre sur la **planification**, tandis que l'approche **agile** se concentre sur la **réaction rapide aux besoins du client**.

2. Le Manifeste Agile

2.1 Définition et historique

En **2001**, 17 experts en développement logiciel se sont réunis aux États-Unis pour réfléchir à une meilleure façon de travailler.

De cette rencontre est né le **Manifeste Agile**, un document qui décrit les **valeurs et les principes** fondamentaux de cette nouvelle approche. Leur but était simple :

“Améliorer la qualité des logiciels et réduire le taux d'échec des projets.”

Ils ont constaté que pour y arriver, il faut :

- **Placer le client au centre du projet**,
- **Favoriser la communication** dans l'équipe,
- **Accepter les changements** tout au long du développement.

2.2 Les 4 valeurs du Manifeste Agile

1. **Collaboration** → La collaboration et la communication sont plus importantes que les outils ou les processus.
2. **Application** → Mieux vaut livrer une application qui marche qu'une longue documentation théorique.
3. **Equipe** → Le client travaille avec l'équipe, au lieu de se limiter à signer un contrat.
4. **L'adaptation au changement** → On accepte le changement et on s'adapte plutôt que de suivre un plan rigide.

2.3 Les 12 principes agiles

- Livrer souvent des versions fonctionnelles du produit.
- Accueillir les changements même tard dans le projet.
- Travailler en étroite collaboration avec le client.
- Construire des projets autour de personnes motivées.
- Préférer la communication directe.
- Mesurer les progrès avec les logiciels livrés.
- Maintenir un rythme de travail soutenable.
- Favoriser la qualité technique.

- Faire simple.
- Laisser les équipes s'organiser elles-mêmes.
- Réfléchir régulièrement à l'amélioration du processus.

3. Qu'est-ce qu'une méthode agile ?

Une **méthode agile** est une manière de **gérer un projet** en suivant les valeurs et les principes du Manifeste Agile. Elle repose sur le travail en équipe, les itérations courtes et une communication constante avec le client.

Le projet est divisé en **petites parties appelées “sprints”** (ou itérations).

Chaque sprint dure en général entre **2 et 4 semaines** et se termine par une **version utilisable du logiciel**.

Pendant chaque sprint :

- L'équipe définit les **objectifs à court terme**,
- Développe et teste les fonctionnalités,
- Présente le résultat au client,
- Adapte le travail pour la suite selon les retours.

Cette méthode permet d'avancer **progressivement**, en gardant toujours le client au centre du projet.

◆ Les caractéristiques d'une méthode agile

- Le travail se fait **par étapes courtes** et répétitives.
- **Le client est impliqué** du début à la fin.
- **Les objectifs sont révisés régulièrement**.
- Le logiciel est **amélioré à chaque version**.
- La communication est **quotidienne et transparente**.

◆ Les points forts des méthodes agiles

- Bonne communication** : les membres de l'équipe échangent souvent, ce qui renforce la cohésion.
- Collaboration client-équipe** : le client voit l'avancement et donne son avis.
- Moins de documents inutiles** : on se concentre sur le code et le produit réel.
- Qualité constante** : les tests sont faits en continu.
- Livraison rapide** : le client voit des résultats concrets régulièrement.
- Amélioration continue** : chaque itération permet de corriger et d'améliorer le produit.

4. Les principales méthodes agiles

Les méthodes agiles les plus utilisées dans les entreprises sont Scrum et SAFe.

Selon une étude *VersioOne (2019)* :

- Scrum représente 54 % des projets agiles (souvent pour une seule équipe).
- SAFe représente 30 % (pour plusieurs équipes qui travaillent ensemble).

◆ Scrum

- C'est la **méthode agile la plus utilisée**. Elle repose sur des **sprints** (cycles courts de travail).
- Ses rôles principaux sont :
 - **Product Owner** (gère les besoins du client),
 - **Scrum Master** (s'assure du bon fonctionnement de l'équipe),
 - **Développeurs** (réalisent le travail technique).

- Le travail est organisé à l'aide d'un **Scrum Board** avec trois colonnes : *A faire, En cours, Terminé.*

◆ **Kanban**

- Cette méthode met l'accent sur le **flux continu du travail**.
- On utilise un **tableau Kanban** pour suivre les tâches (similaire à un tableau Trello).
- Elle est très **visuelle et simple à mettre en place**.
- Idéale pour les **projets en maintenance** ou ceux sans plan fixe.

◆ **Scrumban**

- C'est un **mélange entre Scrum et Kanban**.
- Elle combine la **planification de Scrum** avec la **flexibilité de Kanban**.
- Très utile pour les projets **évolutifs ou complexes**.

◆ **Extreme Programming (XP)**

- C'est une méthode qui pousse les **bonnes pratiques** au maximum.
- Elle encourage le **refactoring**, les **tests fréquents**, et la **propriété collective du code**.
- Le client est toujours disponible pour répondre rapidement aux questions.
- L'objectif est d'obtenir une **excellente qualité de code**.

◆ **SAFe (Scaled Agile Framework)**

- Utilisée pour les **grandes entreprises** ou les projets avec **plusieurs équipes**.
- Elle introduit une structure organisée pour coordonner tout le monde.
- Chaque équipe travaille sur une partie du produit global.

 **5. Comparaison rapide des principales méthodes**

Critère	Scrum	Kanban	Scrumban	XP
Organisation	Sprints planifiés	Flux continu	Mix Scrum/Kanban	Cycles courts de code
Changement	Attente du sprint suivant	Immédiat	Flexible	Adaptable
Documentation	Légère	Légère	Légère	Légère
Rôle du client	Product Owner	Actif	Actif	Très impliqué
Objectif	Livraison régulière	Flux constant	Mix agile	Code de haute qualité

Conclusion

Les **méthodes agiles** ont complètement transformé la manière de gérer les projets informatiques. Elles mettent en avant :

- La **collaboration**,
- La **flexibilité**,
- La **livraison rapide** de logiciels fonctionnels.

Grâce à leur approche **itératrice et incrémentale**, les équipes peuvent réagir rapidement aux changements et **améliorer le produit en continu**.

Elles garantissent ainsi un **logiciel de qualité**, livré dans les temps, et surtout en accord avec les **besoins réels du client**.