

Dans ce premier chapitre nous revenons aux fondamentaux de la philosophie Agile. Nous allons détailler ensemble sa genèse ainsi que ses caractéristiques afin de comprendre comment elle s'est imposée dans le développement de logiciel.

1 Réponse à des problèmes

1.1 Cascade

Définition

Le cycle en cascade se caractérise par des phases séquentielles, qui se succèdent après la validation des livrables de la phase précédente.

La méthodologie *Waterfall* a largement été utilisée pour créer des logiciels. Elle suppose :

- une connaissance parfaite avant même de commencer
- que tout au long du développement le périmètre reste inchangé
- qu'on ne changera pas de direction au cours du développement
- les changements sont supposés être minimes.

La méthode *Waterfall* avait pour but de garantir que le produit final correspond à ce qui avait été spécifié au départ. Cette approche fonctionnait bien sur des applications complexes et monolithiques qui exigeaient de la discipline et des résultats clairs (année 70).

1.2 Le pivot vers la philosophie Agile

Le développement massif d'Internet a permis l'émergence de nombreuses petites entreprises qui créaient des applications web. Les équipes de développement de ces sociétés commençaient à remettre en question la méthodologie *Waterfall* et cherchaient des moyens d'être plus efficace. Ayant une structure organisationnelle et un projet moins complexe qu'une *entreprise legacy systems* elles pouvaient donc être à l'écoute et répondre plus rapidement aux besoins du client.

Les développeurs ont commencé à rejeter la planification de bout en bout et les spécifications prédéfinies. Nous sommes à la naissance de l'agilité :

- Un produit ne peut pas être entièrement spécifié au départ.
- L'économie est trop dynamique : l'adaptation du processus s'impose.
- Accepter les changements d'exigences, c'est donner un avantage compétitif au client.

1.3 Accepter l'incertitude

Affirmation

Il s'agit d'accepter une réalité et de comprendre que dans le développement logiciel tout n'est pas prévisible.

Chaque projet est une nouvelle expérience où il est très difficile de se mettre d'accord une fois pour toutes avec le client. En acceptant l'incertitude, on accepte également l'idée du changement :

- dans le périmètre du besoin
- dans la planification
- dans l'organisation de l'équipe

⇒ On s'adapte aux imprévus. En s'appuyant sur le feedback, l'expérience et le constat. Nous optons pour une démarche d'amélioration continue.

1.4 Accepter le changement

Affirmation

Accueillir le changement à bras ouverts plutôt que de le craindre et le combattre.

On sait que de nombreux paramètres sont imprévisibles lors du projet. Il s'agit donc de mieux contrôler cette imprévisibilité sans la nier en évitant d'être systématiquement obsédé par les plans initiaux obsolètes.

⇒ Une équipe agile se dote de pratiques et d'outils qui lui facilite l'accueil du changement.

1.5 Quelques chiffres

Une étude¹ menée en 1995 nous révèle que :

- 16% des logiciels sont finis dans les temps et le budget alloué
- 31% sont abandonnées
- 53% dépassent le coût et/ou le délai

C'est dans un contexte de gaspillage et de transformation du numérique quand en 2001 une équipe d'expert se réunit afin de rédiger le *Manifeste Agile*.

Depuis, *The Standish Group* publie annuellement le *CHAOS Report* qui regroupe un ensemble de statistique sur les projets informatiques. Par exemple en 2015, on dénombrait 29% des

1. <https://www.csus.edu/indiv/r/rengstorffj/obe152-spring02/articles/standishchaos.pdf>

projet finis en temps/budget, 19% abandonnés et 52% connaissaient des dépassements de cout/délais.

2 La philosophie Agile

Affirmation

L'agilité est une philosophie.

Vous ne pouvez pas rendre une équipe Agile si elle ne comprend pas la philosophie Agile. Si vous appliquez uniquement des outils, des pratiques alors vous n'aurez aucun résultat. L'agilité c'est se convaincre soit même puis convaincre les autres. En d'autre terme, il faut changer de *mindset*.

2.1 Définition

Affirmation

Agile signifie : flexibilité, répétition, ouverture aux changements

Etre Agile c'est :

- adopter une approche itérative et incrémentale
- menée dans un esprit collaboratif
- pour générer un produit de haute qualité
- en tenant compte de l'évolution client

L'agilité s'oppose à des modèles classiques de développement logiciel (e.g. Cycle en V, Cascade). On y travaille par cycle afin d'obtenir du **feedback**. Le feedback est essentiel, car il permet de :

- Revoir les priorités
- Analyser les changements à faire
- Supprimer certains besoins

⇒ Tout ceci contribue à l'amélioration continue de l'équipe et donc du produit.

Affirmation

Le client change d'avis, nous pouvons (plus facilement) l'absorber.

2.2 Implication du client

Avec l'agilité nous impliquons le client au quotidien dans le processus de création. En lui fournissant rapidement des *Minimum Viable Product* nous nous assurons que :

- Le produit répond toujours aux attentes de l'utilisateur et des clients
- De minimiser le cout d'un changement de direction, voire de l'anticiper
- Le client à une visibilité
- ⇒ On évite ainsi le gaspillage d'argent

2.3 L'équipe propriétaire du processus

Affirmation

Dans une approche d'amélioration continue, l'équipe est propriétaire de son processus.

Il est avant tout essentiel que le management lui en donne la permission. Sinon l'équipe ne fera pas évoluer le système et réduisant les différents framework (Scrum, Kanban, etc) à de la simple gestion de tâches.

Chaque membre de l'équipe doit se sentir libre d'intervenir là où il faut, quand il le faut et sur sa propre initiative pour optimiser le flux de travail. Chacun doit ainsi envisager de travailler pour maximiser les performances du système et non sa propre productivité :

- Proposer des solutions pour améliorer le processus global
- Travailler en dehors de ses compétences ou domaine habituel
- ...

2.4 Un produit de haute qualité

L'agilité c'est produire un produit de *haute qualité*. Mais que signifie le terme *haute qualité* ?

- Livraison de valeur continue : on sélectionne et développe les fonctionnalités qui ont une importance capitale pour le client. On évitera de satisfaire tous les besoins initiaux qui ne sont plus utiles ni porteur de valeur.
- Feedback permanent : on recueille l'avis du client en lui livrant régulièrement une version intermédiaire aboutie du produit.
- Campagne de tests : permet au défaut d'être détecté et corrigé immédiatement.
- Refactoring : on évite une dégradation du code en améliorant sa lisibilité et maintenabilité : en supprimant du code dupliqué, etc. On veut un code bien conçu.
- Adopter une approche adaptative : on vérifie régulièrement la qualité du processus (implique directement la qualité du produit). On discute et on prend les décisions adéquates pour améliorer le processus.

2.5 Facilitateur Agile

Définition

Le Chef de Projet n'a plus le rôle autoritaire et de celui qui va assigner les tâches.

L'agilité préconise l'auto-organisation, ainsi le rôle du manager est de s'occuper de *l'humain* (montée en compétence, bien-être des salariés, etc). Il a pour objectif de faire évoluer l'organisation dans le bon sens (amélioration continue des process).

⇒ Il donne un cadre de travail.

3 Le manifeste Agile

Le manifeste Agile est créé en 2001 dans un contexte où les entreprises sont trop concentrées sur planifier et documenter. Elles en oublient l'essentiel : la satisfaction client.

3.1 4 principes

- 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

Affirmation

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

Le *plus que* est très très important. L'agilité ne dit pas de supprimer la deuxième partie ; tout est important, MAIS on privilégiera la première partie.

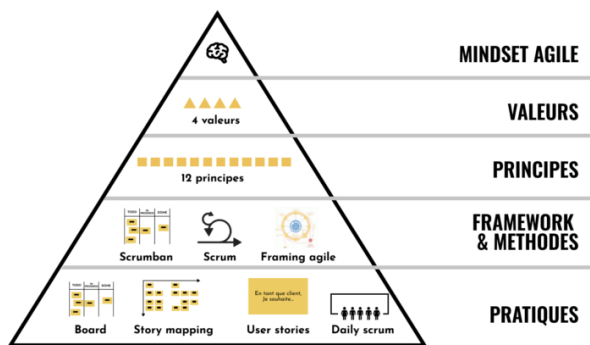
3.2 12 valeurs

Rendent le process flexible et focus sur les personnes

4 Créer une équipe Agile

Affirmation

Vous ne pouvez pas créer une équipe Agile sans comprendre et partir du haut de la pyramide.



Comme nous l'avons déjà évoqué, Agile ne signifie pas prendre des outils et les appliquer. Cela ne marchera pas. Vous devez (comme une secte) penser Agile. Pour cela il faut commencer par le haut de la pyramide pour finalement arriver aux pratiques.

Personne ne va changer ses petites habitudes, va comprendre les avantages. On va avoir une équipe *on fait comme-ci* ...

Créer une équipe Agile c'est travailler en premier sur le mindset puis seulement après sur les frameworks et pratiques.

4.1 Adopter un mindset Agile

Affirmation

Agility is principally about mindset, not practices.



5 Pourquoi devenir Agile

5.1 Approche Agile vs traditionnelles

Dans un modèle en cascade, on ne commence la phase suivante uniquement lorsque la précédente est finie. Par conséquent :

- en cas de détection des défauts, nous devons revenir en arrière.
- en cas de retard, toutes les autres phases sont retardées.

L'agilité permet d'éviter ces problèmes. Nous travaillons les besoins de l'utilisateur de manière dynamique.

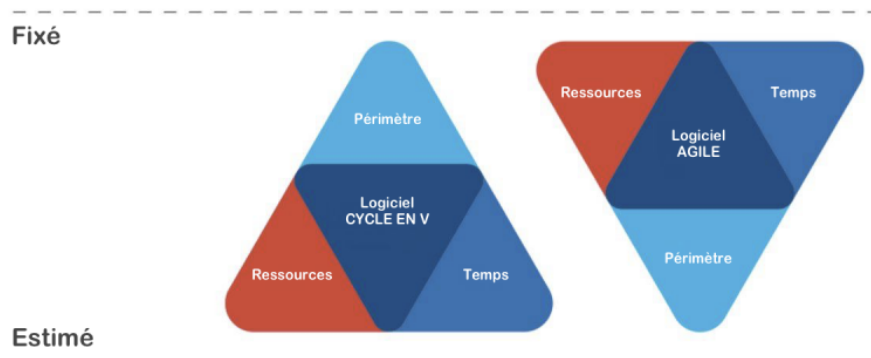


FIGURE 1.1 – Triangle de fer

Waterfall

- Périmètre : fixe
- Coût/Temps : estimé

Agile

- Périmètre : peut changé (écoute du client)
- Coût/Temps : fixe (engagement sur des itérations)

5.2 Tableau des différences

Thème	Approche traditionnelle	Approche agile
Cycle de vie	En cascade ou en V, sans rétroaction possible, phases séquentielles.	Itératif et incrémental.
Planification	Prédictive, caractérisée par des plans plus ou moins détaillés sur la base d'un périmètre et d'exigences définies et stables au début du projet.	Adaptative avec plusieurs niveaux de planification (macro- et microplanification) avec ajustements si nécessaires au fil de l'eau en fonction des changements survenus.
Documentation	Produite en quantité importante comme support de communication, de validation et de contractualisation.	Réduite au strict nécessaire au profit d'incréments fonctionnels opérationnels pour obtenir le feedback du client.
Équipe	Une équipe avec des ressources spécialisées, dirigées par un chef de projet.	Une équipe responsabilisée où l'initiative et la communication sont privilégiées, soutenue par le chef de projet.
Qualité	Contrôle qualité à la fin du cycle de développement. Le client découvre le produit fini.	Un contrôle qualité précoce et permanent, au niveau du produit et du processus. Le client visualise les résultats tôt et fréquemment.
Changement	Résistance voire opposition au changement. Processus lourds de gestion des changements acceptés.	Accueil favorable au changement inéluctable, intégré dans le processus.
Suivi de l'avancement	Mesure de la conformité aux plans initiaux. Analyse des écarts.	Un seul indicateur d'avancement : le nombre de fonctionnalités implémentées et le travail restant à faire.
Gestion des risques	Processus distinct, rigoureux, de gestion des risques.	Gestion des risques intégrée dans le processus global, avec responsabilisation de chacun dans l'identification et la résolution des risques. Pilotage par les risques.
Mesure du succès	Respect des engagements initiaux en termes de coûts, de budget et de niveau de qualité.	Satisfaction client par la livraison de valeur ajoutée.

5.3 Taux de réussite : par des chiffres

L'étude *Chaos Study* menée par le Standish Group affirme que

Affirmation

Agile projects are statistically 2x more likely to succeed and 1/3 less likely to fail than Waterfall projects

Waterfall

- Succès : 14%
- Échec : 21%

Agile

- Succès : 42%
- Échec : 8%

5.4 Quand utiliser l'agilité

Affirmation

l'agilité permet de délivrer de la valeur dans un "monde" avec un haut degré d'incertitude, de risque et de compétition. L'équipe pourra réagir le plus rapidement possible.

Lorsque d'un logiciel va être soumis à beaucoup de changement, il est intéressant de se pencher sur une utilisation Agile.

5.5 Frameworks Agile

Comme vous l'aurez compris Agile représente une philosophie globale pour le développement de logiciels. Elle met l'accent sur la création d'un produit de haute qualité via des itérations. Pour accomplir cet objectif, de nombreux *frameworks* ont été développés.

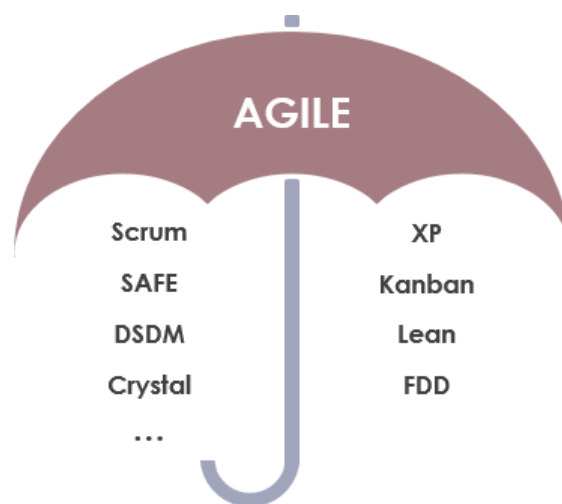


FIGURE 1.2 – Frameworks Agile populaires

6 Patterns organisationnels

Pour les développeurs nous avons les *Design patterns* qui répondent à des problèmes de programmation courants. Et pour la gestion de projet informatique, on retrouve les *patterns organisationnels*.

Les framework Agile ne sont pas seulement un engagement de patterns organisationnels. Ils définissent un cadre de travail composé de patterns organisationnels.

<https://sites.google.com/a/scrumplp.org/published-patterns/home?authuser=0>

Mais garder en tête

Affirmation

Les meilleures solutions sont celles qui émergent de l'équipe

Le rôle du Scrum Master (dans le cas Scrum) n'est ni de donner des solutions ni de montrer comment on fait. Le rôle du Scrum Master est d'apprendre à l'équipe à apprendre. Appliquer les méthodes, et tu te retrouves avec des exécutants !