

Agile

Thomas Clavier <thomas.clavier@univ-lille1.fr>

C'est quoi ?

Être agile et pas faire de l'Agile.

C'est avant tout un état d'esprit partagé par l'ensemble des participants à un projet.

Être agile

"Les firmes qui survivent dans le long terme ne sont pas celles qui sont les plus fortes ou les plus intelligentes, mais celles qui s'adaptent le mieux aux changements d'environnement"

Hiroshi Okuda, Toyota

Pourquoi l'agilité ?

- Méthode en V
- Effet tunnel
- Contrat
- Perdant / perdant !

Le manifeste agile

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

L'équipe

« Personnes et interaction plutôt que processus et outils »

Dans l'optique agile, l'équipe est bien plus importante que les moyens matériels ou les procédures. Il est préférable d'avoir une équipe soudée et qui communique composée de développeurs moyens plutôt qu'une équipe composée d'individualistes, même brillants. La communication est une notion fondamentale.

L'application

« Logiciel fonctionnel plutôt que documentation complète »

Il est vital que l'application fonctionne. Le reste, et notamment la documentation technique, est secondaire, même si une documentation succincte et précise est utile comme moyen de communication. La documentation représente une charge de travail importante, mais peut pourtant être néfaste si elle n'est pas à jour. Il est préférable de commenter abondamment le code lui-même, et surtout de transférer les compétences au sein de l'équipe (on en revient à l'importance de la communication).

La collaboration

« Collaboration avec le client plutôt que négociation de contrat »

Le client doit être impliqué dans le développement. On ne peut se contenter de négocier un contrat au début du projet, puis de négliger les demandes du client. Le client doit collaborer avec l'équipe et fournir un feed-back continu sur l'adaptation du logiciel à ses attentes.

L'acceptation du changement

« Réagir au changement plutôt que suivre un plan »

La planification initiale et la structure du logiciel doivent être flexibles afin de permettre l'évolution de la demande du client tout au long du projet. Les premières releases du logiciel vont souvent provoquer des demandes d'évolution.

Valeurs et principes

Des cycles courts, un produit en production à chaque fin de cycle et des producteurs de valeurs qui s'améliorent continuellement.

Histoire

- Ve siècle av. J.-C : L'Art de la guerre de Sun Tzu
- Les Bâisseurs de cathédrales
- 1880 : Taylorisme
- 1908 : Fordisme

Toyota

- 1950, Toyota
- Kaizen : amélioration continue, « Mieux qu'hier, moins bien que demain. »
- 5S : propre et ordonné, mais surtout organisé selon l'usage
- Kanban : les fiches sur les chaînes de montage

D'autres exemples

- 1983 : Yamaha et MBK
- Canon et son Canon Products System
- 1980 : Sony, Fujitsu
- 2001 : La poste japonaise, Ito-Yokado avec l'aide de Toyota

Favi : 40 ans d'agilité

- adaptations de toutes les méthodes japonaises au contexte local
- des fiches
- L'homme est bon => pas de sanction mais de l'action
- Quoi ? Qui ? = Pourquoi ?
- 1 jour, 20 ans

Les grands du web

- 2 pizzas team
- Devops chez Amazon
- du nombre de relations à l'organisation des sociétés
- créer des groupes sociaux culturel
- privilégier le "time to market", donc le client !
- générique implique complexité donc délais + coût

Les méthodes

- Lean
- Scrum
- Kanban
- Devops
- Extreme Programming

Principe

- Travailler ensemble : collectif
- Concentré sur un sous ensemble de tâches
- Simplicité, efficacité, qualité

Scrum

- un backlog produit
- un backlog d'itération
- des rendez-vous
- un produit
- une équipe
- un product owner
- un scrum master

Les rendez-vous Scrum

- Planning Poker
- Découpage en tâches
- Mêlée quotidienne
- Démonstration
- Rétrospective