

# Méthodes agiles, Use Case, User story..

Marc de Beaumont

# Sommaire

Les méthodes agiles

Scrum

Use case

User stories

# Les méthodes agiles

- Le « Manifeste pour le développement logiciel agile », en 2001, aux Etats Unis, signé par 17 experts en développement logiciel.
- 4 valeurs principales :
  - Les individus et leurs interactions avant les processus et outils
  - Les fonctionnalités opérationnelles avant la documentation
  - La collaboration avec le client plutôt que la contractualisation des relations
  - L'acceptation du changement plutôt que la conformité aux plans
- Déclinées en 13 principes applicables dans toute démarche agile

## Les méthodes agiles (2)

- 6 concepts, qui doivent être considérés de façon interdépendante :
  - Valeur
  - Incertitude
  - Client
  - Individus
  - Équipe
  - Contexte.

## Les méthodes agiles (3)

- La valeur ne peut être créée qu'avec une collaboration avec le client.
- Une équipe ne peut être performante sans une contribution significative des individus qui la composent.
- L'incertitude ne peut être maîtrisée que si vous prenez en considération la spécificité du contexte.

# Les méthodes agiles (4)

- Les principales méthodes agiles :
  - ASD (Adaptative Software Development)
  - Les méthodologies Crystal
  - DSDM (Dynamic Software Development Method)
  - RAD (Rapid Application Development)
  - Lean
  - Scrum
  - UP (Unified Process)
  - XP (eXtreme Programming)

# Les méthodes agiles (5)

- Avantages :
  - Apport de valeur ajoutée (hiérarchisée, priorisée)
  - Adaptabilité (grâce au feedback du client)
  - Visibilité accrue (sur travail effectué, et restant à faire)
  - => Planification et arbitrages nécessaires sont facilités
  - Réduction des risques : grâce au feedback permanent, les dérives et dysfonctionnements sont détectés précocement et peuvent être amoindris, par l'acceptation du changement
  - Qualité logicielle

# Scrum

- Les racines : article de Takeuchi et Nonaka, paru en 1986 dans Harvard Business Review,
- Qui met en avant les vertus des petites équipes pluridisciplinaires intégrées et soudées dans une stratégie plus flexible qu'habituellement.
- Ken Schwaber et Jeff Sutherland ont développé Scrum en 1993.
- Le cycle de vie Scrum est rythmé par des itérations de 4 semaines, les sprints.
- La liste des exigences initiales, dressée et hiérarchisée avec le client constitue le référentiel, le product backlog.



# Use Case

- Une technique de formalisation et de modélisation UML, recommandée entre autres, dans la méthode du Processus Unifié (use-case driven development)
- Décrit d'un point de vue fonctionnel, les échanges entre le système à développer et les acteurs externes (utilisateur ou autre système), dans un contexte particulier
- Décompose l'usage qui est fait du produit ou du système à développer en un but précis.

## Use Case (2)

- La séquence d'étapes pour atteindre ce but est décrite dans un scénario nominal
- Mais différentes séquences doivent être envisagées, selon les conditions qui peuvent varier, pour atteindre le même but ou échouer ; ce sont les scénarios d'exception ou d'échec

## Use Case (3)

- Un UC se présente généralement sous forme textuelle, même s'il est parfois représenté sous forme de diagramme ou d'activités UML
- En tant que support de communication avec le client, il est important qu'il conserve sa forme textuelle, qui ne nécessite aucune formation pour le comprendre et le valider.

## Use Case (4)

- Si l'on adopte cette technique pour formaliser les exigences, il faut savoir qu'un UC ne décrit que les exigences fonctionnelles,
- Qu'il faudra compléter avec d'autres exigences : interfaces externes, règles métier, formats de données, performance,...
- Alistair Cockburn, grand spécialiste des UC, considère
  - le cas d'utilisation comme « le moyeu d'une roue »
  - et les autres types d'exigences comme « des rayons partant dans des directions différentes ».

# User stories

- Une exigence du système à développer
- formulée en une ou deux phrases
- dans la langue de l'utilisateur
- pour servir un but
- Sa granularité doit permettre à l'équipe de réalisation d'estimer son coût et de la réaliser entièrement au cours d'une itération.

## User stories (2)

- Le formalisme n'est pas imposé
- Rien n'est rédigé
- Les user stories sont des aide-mémoire qui retracent, souvent sur une fiche en bristol, voire un Post it,
- les échanges qui ont été menés en face à face avec le client

## User stories (3)

- **Avantage** : elles facilitent la démarche en deux temps :
  - Générales et grossières au début
  - Elles s'enrichissent ensuite d'exemples, de notes, et de cas de tests, au moment où elles sont traitées

Elles facilitent donc la communication, l'échange, l'ajout ou la suppression de détails.

Elles n'ont pas pour objectif d'être pérennes : seuls la fonctionnalité réelle et le code commenté documentent le produit.

## User stories (4)

- Inconvénient : la démarche fonctionne bien si le client est à proximité et disponible.
- Rassemblées dans un référentiel unique, elles constituent le « product backlog »



# Use case ou User stories ?

- Le choix de la technique dépend fortement du contexte et de la méthodologie adoptée.
- Les use cases seront plus utilisés dans la cadre de projets où l'équipe est dispersée géographiquement.
- Inversement, les user stories sont parfaitement adaptées si le client est disponible et le nombre d'utilisateurs réduit.