

# AGILITE - INTRODUCTION

ESTM – DAKAR

Année 2018 – Mamadou Boli BA

Définition

Historique

Le manifeste Agile

Pourquoi l'agilité

La méthode  
SCRUM

# Sommaire

1. Définition
2. Historique
3. Le manifeste Agile
4. Pourquoi l'agilité
5. La méthode Scrum

# Le projet AGILE

## Définition

- « Une **méthode agile** est une approche itérative et incrémentale, qui est menée dans un esprit collaboratif avec juste ce qu'il faut de formalisme. Elle génère un produit de haute qualité tout en prenant en compte l'évolution des besoins des clients »
- Une réponse à l'inadéquation du cycle en V/cascade avec les nouveaux besoins applicatifs

# Le projet AGILE

## L'approche Incrémentale



Comment le client a exprimé son besoin



Comment le chef de projet l'a compris



Comment l'ingénieur l'a conçu



Comment le programmeur l'a écrit



Comment le responsable des ventes l'a décrit



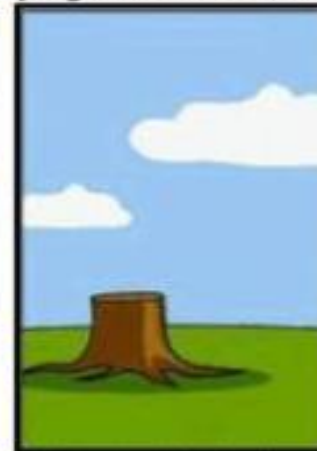
Comment le projet a été documenté



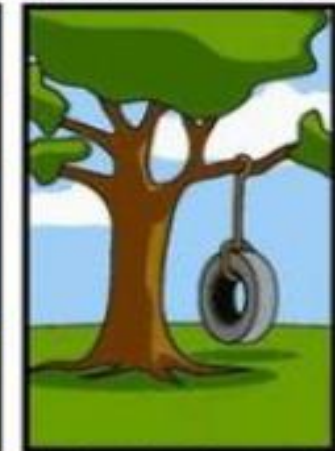
Ce qui a finalement été installé



Comment le client a été facturé



Comment la hotline répond aux demandes



Ce dont le client avait réellement besoin

# Le projet AGILE

## L'approche Incrémentale



*Simplicité:*

*l'art de maximiser la  
quantité de travail à ne  
pas faire*

# Le projet AGILE

## Historique et Manifeste Agile

« L'homme est le terme unique d'où il faut partir et auquel il faut tout ramener »

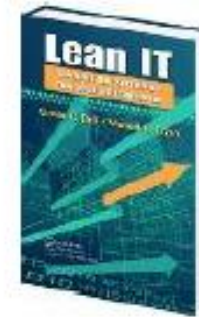
Denis Diderot

### Historique

- Fin 70 à 90: expérimentations empiriques
- 90: émergence des méthodes RAD, ASD, etc
- 1996: Scrum
- 1999: XP eXtreme Programming
- 2001: Manifeste Agile

# Le projet AGILE

## Introduction à l'agilité



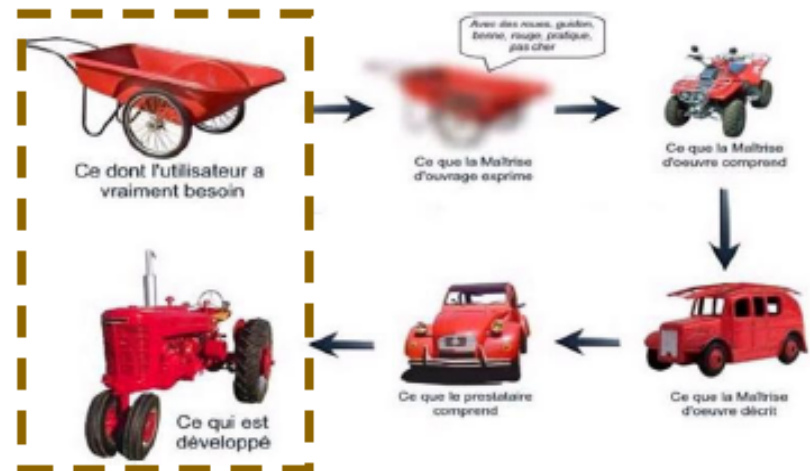


# Le projet AGILE

## L'agilité, une réponse à quelles préoccupations?

Les projets réalisés suivant un processus de développement en mode cascade :

1. sont confrontés aux **limites de la prédiction**,
2. ont un **effet tunnel** important : manque de visibilité sur les développements en cours,
3. et ont de grandes difficultés à revenir en arrière en cas **d'insatisfaction du client** à la sortie du tunnel.



- Au démarrage du projet, le client ne connaît pas forcément complètement son propre besoin : il est inutile de le 'forcer' à décrire de façon exhaustive un besoin qu'il ne connaît pas encore !
- Le maquettage en amont n'est pas toujours suffisant pour éclaircir un besoin flou : c'est une vue parcellaire et souvent embellie de la cible.
- Le mode 'push' (enchaînement de métiers) produit une distorsion à chaque transition ainsi que du stock (exemple : 200 pages de spécifications .... qui ne seront matérialisées en vrai logiciel qui fonctionne ... beaucoup plus tard)
- L'excès de prédiction est la conséquence d'engagement construits sur 'du sable' (un terrain mouvant) pour lesquels on imagine tout (et même son contraire !). Mais comme on sait que l'on ne peut pas avoir raison sur toutes nos prédictions, l'erreur est compensée par des provisions sur risques : symptomatiques de notre incapacité à tout prévoir.
- Ce n'est qu'en fin de chaîne que l'on s'aperçoit du résultat... ! Mais c'est trop tard, on cherche alors à justifier les écarts (ce n'était pas écrit, ce n'est pas un bug, vous aviez pourtant donné votre accord il ya 10 mois lors du copil n°14 : c'est écrit dans le CR ...)

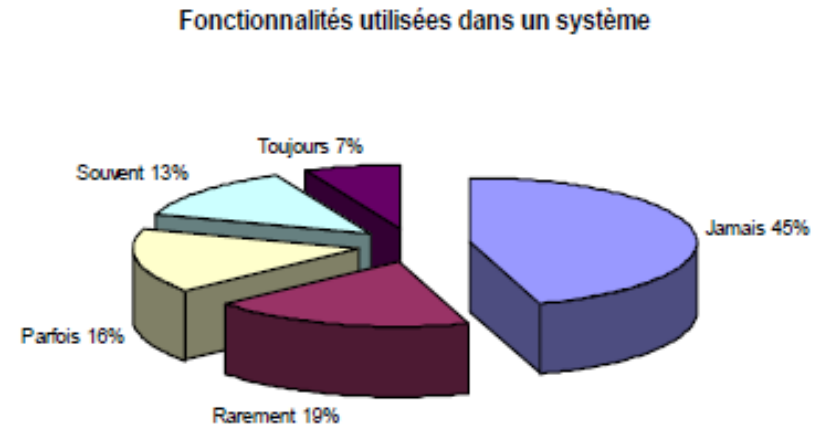


# Le projet AGILE

L'agilité, une réponse à quelles préoccupations?

Seules 20% des fonctions  
sont réellement utilisées !

L'agilité promeut le  
« *Juste nécessaire* »



Standish Group study Reported at XP2002 by Jim Chairman

Proposition  
de Valeur  
de l'Agilité

Maximiser  
la **valeur métier**  
pour le **budget disponible**  
dans un **délai contraint**  
dans un environnement **changeant**

# Le projet AGILE

## Le manifeste AGILE

### Agile Manifesto

Nous découvrons comment mieux développer des logiciels  
par la pratique et en aidant les autres à le faire.  
Ces expériences nous ont amenés à valoriser :

**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

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

*source <http://agilemanifesto.org/iso/fr/>*

# Le projet AGILE

## Le manifeste AGILE

L'équipe : 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

<http://agilemanifesto.org/iso/fr/>

# Le projet AGILE

## Les 12 principes de l'agilité

- Satisfaire le client en livrant **tôt et régulièrement** des logiciels utiles.
- Le changement est **accepté**, même tardivement dans le développement. Le changement est un **avantage** compétitif pour le client.
- Livrer **fréquemment** une application **fonctionnelle**.
- **Collaboration quotidienne** des experts métier et les développeurs.
- Equipe **motivées**, soutien et confiance en cette équipe.
- Conversation en **face à face**.
- **Logiciel fonctionnel** = unité de mesure de la progression du projet.
- Rythme **durable** de développement
- Excellence technique et qualité de la conception
- Simplicité, **minimiser le travail à faire**
- Equipes **auto-organisées** => meilleures architectures, spécifications et conceptions
- **Réflexions rétrospectives régulières** -> ajuster son comportement.

# Le projet AGILE

## Pourquoi favoriser l'agilité

### ■ Constat

- 90 % des projets terminent en retard
- 74% des projets sont annulés ou échouent
- 59% des fonctionnalités développées dans un produit en mode cascade ne sont jamais ou rarement utilisées !

### ■ Avantages

- Réduction des couts, amélioration de la rentabilité
- Amélioration des délais de livraison
- Augmentation de la satisfaction client
- Maîtrise du produit et des changements
- Livraison de produits de haute qualité
- Amélioration continue des processus et des pratiques de l'entreprise

# Le projet AGILE

## Pourquoi favoriser l'agilité

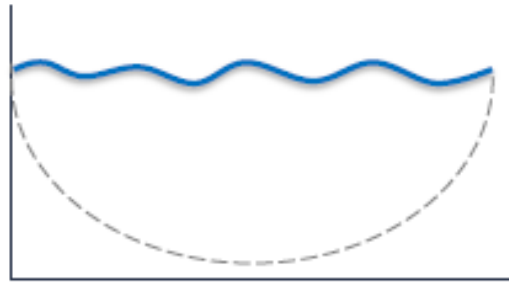
- Les aspects financiers
  - Réduction des coûts de maintenance de l'application
  - Time to market : délais réduits
  - Adaptation plus rapide aux évolutions du marché
  - Payback plus rapidement atteint



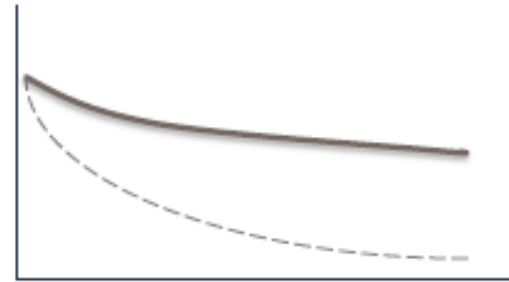
# Le projet AGILE

## Pourquoi favoriser l'agilité

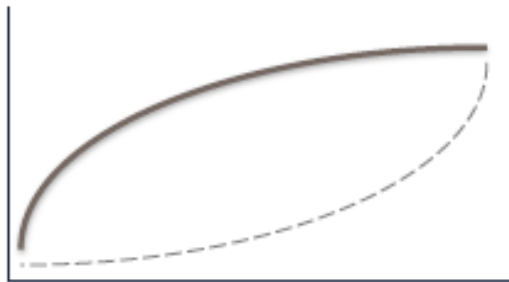
Visibilité



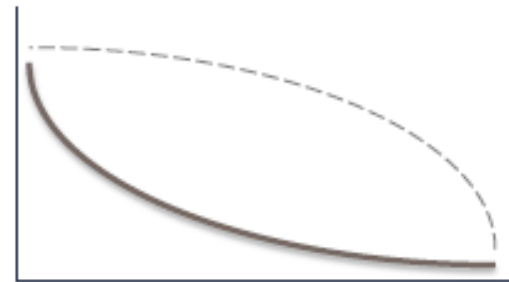
Adaptabilité



Business Value



Risques



— Développement Agile    - - - - Développement traditionnel

# Le projet AGILE

## Ce que l'agile n'est pas

- **Du « *quick and dirty* »** : l'agile met un accent fort sur la qualité de la production.
- **Des maquettes, du jetable** : en agile nous travaillons par itérations sur le vrai produit cible.
- **Du « chaos »** : au contraire, l'agile nécessite une rigueur et un investissement personnel accrus :
  - Implication des acteurs : plus de polyvalence.
  - Respect des processus : règles d'ingénierie (contrôle des sources, tests, documentation suffisante, intégration continue), participation régulière aux « cérémonies », maintient rigoureux des « artefacts ».
  - Ponctualité et cadencement : livraisons à l'heure, début et fin des « cérémonies ».

# Le projet AGILE

## Focus sur SCRUM



Equipe soudée, client inclus,  
qui pousse dans le même  
sens  
pour atteindre un objectif  
partagé

# Le projet AGILE

## SCRUM EN UN SLIDE

- *Framework* de gestion de projet agile
- Formalisé par Ken Schwaber et Jeff Sutherland en 1995

### Principes

- Livraison de valeur à des intervalles réguliers (*sprints*).
- 3 rôles : *Product Owner*, *Scrum Master*, l'équipe  
réunions: *daily stand-up*, *sprint planning*, *sprint review*, *sprint retrospective*.
- artefacts : *Product Backlog*, *sprint backlog*, *project dashboard*, *burndown(up) chart*, *vélocité*, *liste d'impediments*.

# Le projet AGILE

## Les pratiques SCRUM

**SCRUM**



**PRATIQUES XP**

Utilisation la plus courante de l'agilité

**Cadre de management projet  
ADAPTATIF**

Rôles  
Cérémonial  
Planification  
Gestion des besoins  
Estimations  
Suivi avancement

**Pratiques d'ingénierie du logiciel  
INDUSTRIELLES**

Test Driven Development  
Intégration continue  
Tests automatiques  
Conception Simple  
Refactoring  
Standard de codage

# Le projet AGILE

## Les points forts de Scrum

- Scrum est un *framework* composé de bonnes pratiques de gestion de projet qui peuvent être utilisées en mode « boîte à outils ». Cela rend Scrum **customisable et adaptable au contexte**.
- Entre autres, cela permet d'adapter Scrum à des projets de développement, d'intégration, de paramétrage, à la maintenance d'applications, etc.
- Scrum est la méthode agile la plus répandue au monde, ce qui veut dire que son vocabulaire est très largement connu et compris : *sprint*, *backlog*, *dashboard*, *Scrum Master*, *Product Owner*... ne nécessitent plus (beaucoup) d'explication, ce qui facilite leur implémentation.



# Le projet AGILE

## Les pré requis pour faire du Scrum

- Disponibilité et engagement du métier/MOA : *Product Owner* est un travail temps plein.
- Périmètre pas détaillé initialement au delà du nécessaire pour la charte projet.
- Contractualisation adéquate :
  - Idéalement, *time box* avec clause de sortie à la fin de chaque sprint. Ceci entraine une notion d'engagement sur le résultat (résultat != périmètre prédéfini, résultat = satisfaction du client)
  - Le forfait classique (*fixed price* sur un périmètre) n'est pas vraiment adapté. Deux « astuces » permettent de faire des projets Scrum dans le cadre d'un forfait classique :
    - La régie « forfaitisé » : n développeurs pendant m sprints, engagement de livrer aux dates agréés.
    - Le forfait flexible, où l'ensemble du *backlog* est chiffré d'avance (un peu à l'encontre de Scrum...) et où l'on peut remplacer des *user stories* par d'autres de poids équivalent.

# Le projet AGILE

## Les pré requis pour faire du Scrum

- Pari de la confiance : critique surtout au début, pour amorcer la pompe.
- Les *user stories* :
  - Nécessaires car Scrum est tourné « utilisateur » et « preuve » : non seulement elles servent à organiser l'analyse et la conception, mais surtout les scénarios de test, qui doivent respecter une cohérence vu des utilisateurs.
  - A tout effet pratique : user story = use case (UML/RUP).
  - Un « utilisateur » d'une *user story* peut très bien être un deuxième système.
  - Un pattern simple de *user story* : «En tant que <rôle>, je veux <but/souhait> afin de <avantage/profit>».

# Le projet AGILE

## Quand faire du scrum

- Au-delà d'une certaine durée (> 1 mois).
- Avec des équipes d'au maximum 10 personnes.
  - Il est possible de découper un projet plus grand en plusieurs sous-projets Scrum, tout de même.
- Si l'environnement le permet :
  - Le système qualité n'oblige pas à avoir des revues d'analyse et de conception, par exemple.
  - Le *business* est prêt à s'investir et à consacrer du temps régulièrement sur toute la durée du projet.
  - Pas en contact avec un partenaire aligné sur une méthode en « V »
- 100% Scrum, ou bien les bonnes pratiques pertinentes :
  - Scrum est un *framework*, l'ensemble de ses processus ne sont toujours pas nécessaires ni pertinents.
  - Il est tout à fait possible d'implémenter Scrum partiellement, et toujours en retirer une partie de ses avantages (p.ex: en mode « forfait flexible »)

# Le projet AGILE

## Quelques idées reçues

- A périmètre fixe, l'utilisation de Scrum n'entraîne pas des gains de productivité automatiques.
  - Toutefois, quand Scrum est appliqué, des gains de productivité à périmètre égale sont fréquents de par l'appropriation du projet par l'équipe et la forte motivation qui en découle : l'équipe se sent plus responsable du succès du projet.
- Scrum permet de livrer plus de valeur, par forçement plus de lignes de code ou plus de fonctionnalités.
- Scrum marche mieux à partir d'un certain nombre de sprints (heuristique = 4), car ses processus nécessitent d'une équipe stable.
  - Ceci est encore plus vrai lors du premier projet Scrum.

# Le projet AGILE

## Clés de succès

- Engagement du métier/MOA.
- Cadre contractuel propice.
- Pari de la confiance.
- Analyse et conception continues sous forme d'*user stories*.
- Idéalement > 4 sprints (>3 mois), surtout si premier projet Scrum et/ou nouvelle équipe.

Lors des premiers projets agiles, il est indispensable d'avoir un accompagnement d'un coach Scrum :

- Pour l'équipe et le Scrum Master.
- Mais aussi pour le *Product Owner* et le métier.

# Le projet AGILE

## Conclusion

- Le développement agile repose sur des principes simples et naturels :
  - la satisfaction et l'implication du client
  - l'accent mis sur les capacités individuelles et l'expertise
  - la collaboration, la confiance et la communication
  - le développement itératif, simple et robuste
  - l'intégration continue
  - l'adaptation aux changements