



Les méthodes Agile

Simon ALEXANDRE - CETIC



Plan

- Overview
- Agile ne signifie pas...
- Agile signifie...
- Objectifs poursuivis
- Pourquoi les méthodes Agile apparaissent-elles ?
- Principales causes des échecs de projets
- Solutions proposées par Agile
 - Implication du client
 - Développement itératif et incrémental
- Autres concepts fondamentaux Agile
- Sur quels types de projets utiliser une méthode Agile ?
- Exemple: Scrum
- Quelques références



Overview

- Quelques exemples de méthodes Agile:
 - Extreme Programming, Scrum, Feature Driven Development, ASD, DSDM,...
- Apparition dans les années 90'
- Issues de l'industrie
- Diffusion au sein des entreprise ↗ depuis 2000



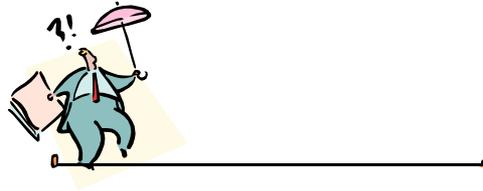
Agile ne signifie pas...

- Pas de planning
- Pas de cahier des charges
- Pas de gestion de projet
- Pas d'attention à la qualité
- Coding pur et dur
- *Quick and dirty*



Agile signifie...

- Qu'entendre par méthode Agile ?
 - "Les méthodes de développement de type Agile suivent un mode de **développement itératif** et **incrémental**, une **planification de projet évolutive** et encouragent les **release fréquentes** au client. Elles incluent également toute une série d'autres valeurs et pratiques qui encouragent l'agilité et une réponse aux changements." (adapté de Craig Larman 2003)



Objectifs poursuivis

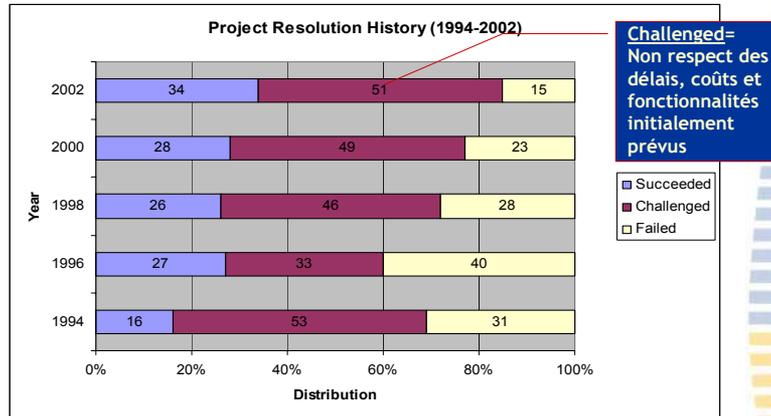
- Méthodes Agile mettent l'accent sur:
 - client
 - rapidité
 - changement
- Software Manifesto 2001
 - *Individuals and interactions* over *processes and tools*
 - *Working software* over *comprehensive documentation*
 - *Customer collaboration* over *contract negotiation*
 - *Responding to change* over *following a plan*

Source: <http://www.agilemanifesto.org>



Pourquoi les méthodes Agile apparaissent-elles ?

- Taux d'échec des projets informatiques reste important (Rapport Chaos 2003)



Principales causes des échecs de projets

- Développements de type cascade-linéaire:
 - Taux échec >
- Cause:
 - Application trop rigide des cycles type cascade
 - facteurs du contexte qui impactent les projets et qui ne sont pas assez pris en compte par les cycles de type waterfall
 - o Taille des projets
 - o Changements d'exigences
 - o Inflation des fonctionnalités



Solutions proposées par Agile

- Les méthodes Agile répondent à ces problèmes en mettant l'accent sur deux aspects:
 - *L'implication du client*
 - *Les développements itératifs et incrémentaux*



Solutions proposées par Agile

- **Implication du client**
 - Base du logiciel = exigences du client
 - Postulat de départ: le client connaît ses besoins (i.e. exigences stables) et/ou les exigences de départ sont stables
 - Constat: situation assez rare
 - Solution: impliquer le client au maximum dans le projet
 - o Moyens: client sur site, développement chez le client, prototypes,...



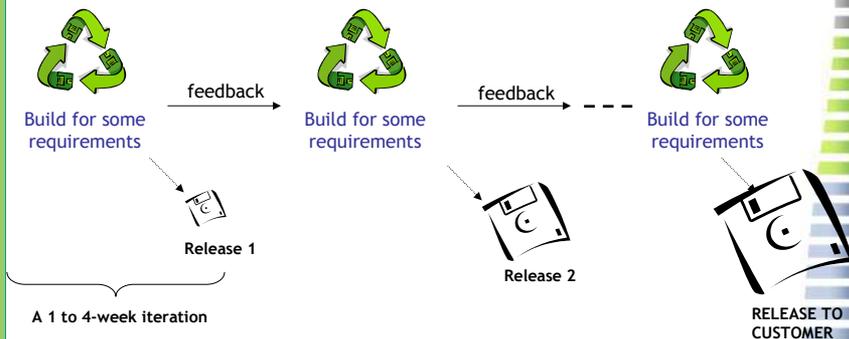
Solutions proposées par Agile

- **Développement itératif et incrémental:**

- Un **développement itératif** est une approche utilisée pour développer du logiciel au cours de laquelle l'ensemble du cycle se compose de plusieurs itérations successives.
 - o Chaque **itération** => mini projet composé (par ex.) de : Analyse des exigences, Design, Programmation, Tests
- Développement **Incrémental**: logiciel construit progressivement (par sous-systèmes successifs)



Développement itératif et incrémental



Autres concepts fondamentaux Agile

- La plupart des méthodes Agile mettent l'accent sur:
 - *Intégration du changement*
 - *Cycles de développement courts et release fréquentes*
 - *Design le plus simple possible*
 - *Refactoring*
 - *Pair programming*
 - *Test-driven development*



Sur quels types de projets utiliser une méthode Agile ?

- Choix de la méthode
- Plusieurs facteurs à prendre en compte dont:
 - Criticité du système (pas life critical!)
 - Petite équipe (10-15)
 - Equipe composée d'une majorité de seniors
 - Maîtrise de la gestion de projet
 - Exigences floues et variables
 - Client disponible (sur site = optimal)
- **Attention: Ne s'applique pas à TOUS les TYPES de PROJETS !!!**

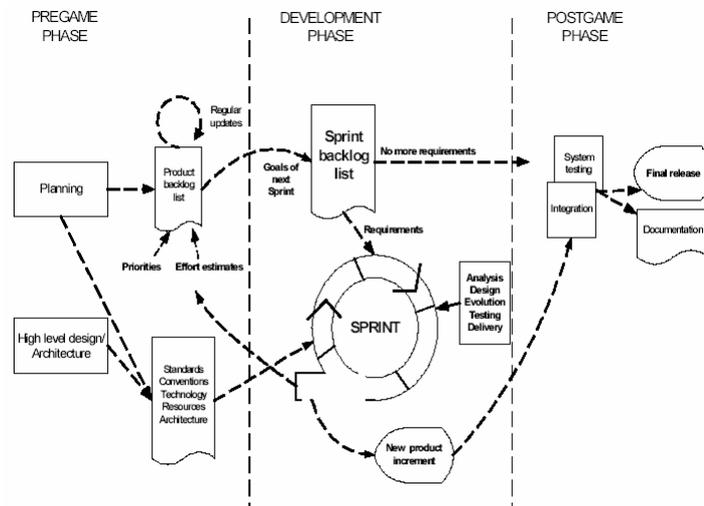


Exemple: Scrum

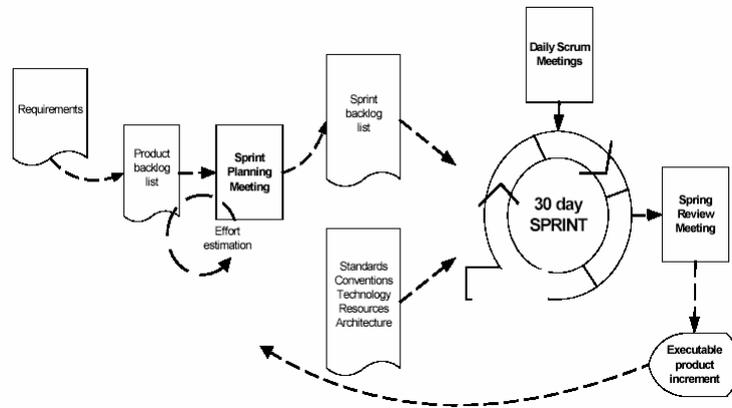
- Origine industrielle: Appliquer les théories de processus de control industriel dans une démarche qui réintroduit les idées de flexibilité, adaptabilité et de productivité
- L'idée centrale de Scrum:
 - Le développement de systèmes implique plusieurs variables (Exigences, Calendrier, Ressources & Technologie)
 - variables susceptibles de changer en cours de projet
 - Rend le processus de développement:
 - o prédictible
 - o Complexe
 - Etre conscient de ces facteurs
 - Flexibilité → répondre aux changements



Scrum: processus global



Scrum: Sprint



Quelques références

- Abrahamsson, P., *Agile Software Development Methods: Review and Analysis*, VTT Publications 478, 2002.
- Richard Turner, Barry W. Boehm, *Balancing Agility and Discipline: A Guide for the Perplexed*, 2003.
- Craig Larman, *Agile and Iterative Development: A Manager's Guide*, 2003.
- Matt Stephens, Doug Rosenberg, *Extreme Programming Refactored: The Case Against Xp*, 2003.
- Kent Beck, *Extreme Programming Explained: Embrace Change*, 1999.
- Lisa Crispin, Ken S. Rosenthal, *Testing Extreme Programming*, 2002.
- <http://www.agilealliance.org>

