

Un peu d'agilité dans nos projets?

Laurent Guillo
18 octobre 2014

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTEMES ALÉATOIRES



Plan

- **Agilité? ...**
- **Allez, on essaie...**
- **Résumé**



20/10/2014

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTEMES ALÉATOIRES



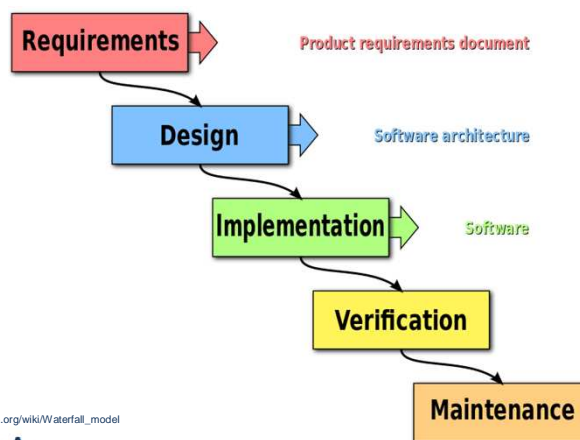
Plan

- **Agilité? ...**
 - **Les fondements**
 - Exemple de mise en œuvre
- Allez, on essaie...
- Résumé

Avant l'agilité...

(1/2)

- Le modèle en cascade



Avant l'agilité... (2/2)

- Le modèle en cascade présente des défauts:
 - Demandes des clients validées/confirmées tardivement
 - Conflit entre ce qui est demandé et ce qui est possible
 - Quasi impossible de finir une étape complètement avant de passer à la suivante (e.g. « design is a wicked problem »)
 - Pas adapté aux modifications

2001, Utah: « Agile Manifesto »

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

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

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

© 2001, the authors.
 All rights reserved. No part of this work may be reproduced without the prior written permission of the authors.

- 4 valeurs qui s'appuient sur ...

... 12 principes:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage..

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. .

Business people and developers must work together daily throughout the project..

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done. .

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation..

Working software is the primary measure of progress. .

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely..

Continuous attention to technical excellence and good design enhances agility.. .

Simplicity--the art of maximizing the amount of work not done--is essential. . .

The best architectures, requirements, and designs emerge from self-organizing teams. .

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly. .



20/10/2014

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTEMES ALÉATOIRES



Attention!

- Le manifeste
 - N'est pas une bible
 - Ne donne pas tel quel des règles de mise à œuvre
 - N'est pas une méthodologie mais un cadre
- Il insiste sur:
 - La collaboration entre tous
 - Le développement adaptatif
 - Des cycles itératifs courts et incrémentaux



20/10/2014

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTEMES ALÉATOIRES



Exemple de principes mis en avant:

Our highest priority is to satisfy the customer through **early and continuous delivery** of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage..

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. .

Business people and developers must **work together daily** throughout the project..

Build projects around **motivated individuals**. Give them the environment and support they need, and **trust them** to get the job done. .

The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation**..

Working software is the primary measure of progress. .

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely..

Continuous attention to technical excellence and good design enhances agility.. .

Simplicity--the art of maximizing the amount of work not done--is essential. . . .

The best architectures, requirements, and designs **emerge from self-organizing teams**. .

At regular intervals, the team **reflects on how to become more effective**, then tunes and adjusts its behavior accordingly. .



20/10/2014

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTEMES ALÉATOIRES



2 Méthodes agiles principales

- 1999 : XP (Kent Bech, Ward Cunningham)
- 2001: Scrum (Ken Schwaber, Mike Beedle)
- (+1 1962-20xx: Kanban/Lean)

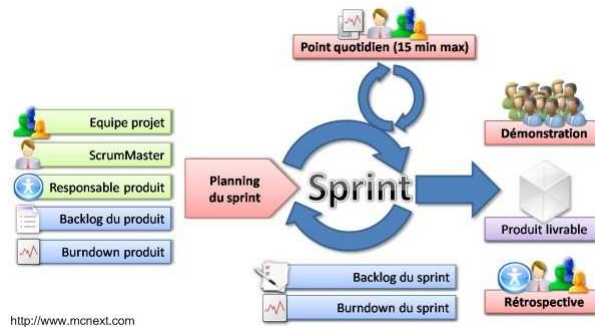


20/10/2014

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTEMES ALÉATOIRES



Toutes s'appuient sur des cycles itératifs



Plan

- **Agilité? ...**
 - Les fondements
 - **Exemple de mise en œuvre**
- Allez, on essaie...
- Résumé

Exemple de mise en œuvre

- Concerne:
 - Les intervenants
 - Les actions menées
 - Le suivi du projet
 - L'analyse des actions menées

Intervenants

- Au nombre de 3:
 - Product Owner (PO) : connaît les besoins des clients
 - Team : équipe de développement
 - Scrum Master (SM) : responsable de l'application de la méthode agile

Actions

- Au nombre de 4:
 - Définir le **Product backlog**
 - Préparer le futur **Sprint** et les outils de suivi
 - Suivre/animer le sprint et mesurer son avancement
 - Analyser le déroulement du dernier sprint

- Qui fait quoi? A quel moment?

Product backlog (1/2)

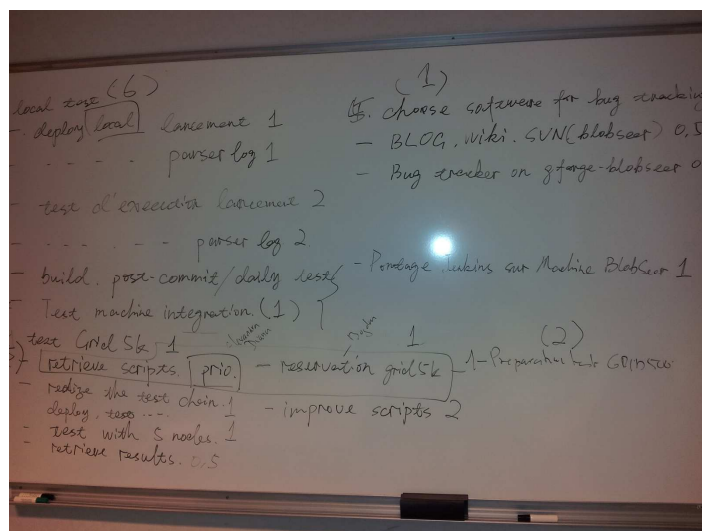
- Quoi?:
 - Liste ordonnée des fonctionnalités du (des) logiciel(s), des actions (doc, rapport, ...): les **stories**.
 - Leur durée est évaluée et elles alimentent les **Sprint backlog**

- Qui?:
 - Géré par le PO au début du projet et maj ensuite
 - Défini par le PO et la TEAM (le SM anime)

Product backlog (2/2)

- Quand?
 - Défini au début du projet mais non figé
- Conseils:
 - Nommer les stories avec un identifiant unique (entier) + nom/description courte
 - Sauvegarder le product backlog dans un espace commun (wiki, ...)

Exemple de product backlog en cours d'élaboration...



Le sprint backlog (1/2)

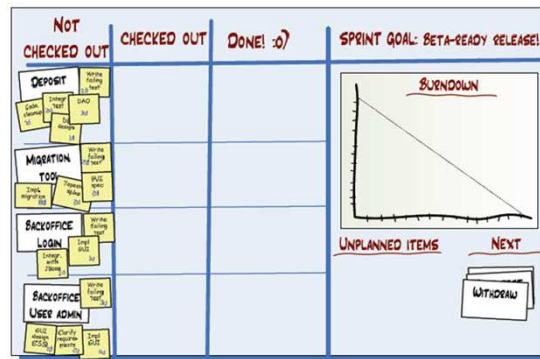
- Quoi?
 - Liste de stories extraites du product backlog composées de tâches chargées à réaliser par la TEAM
- Qui?
 - Défini **conjointement** par le PO et la TEAM en début de chaque cycle (chacun participe!)

Le sprint backlog (2/2)

- Quand?
 - En début de chaque itération (de 1 à 3 semaines)
- Conseils:
 - Nommer le sprint
 - Rappeler les objectifs du sprint
 - Préciser la durée, dates début-fin
 - Sauvegarder le sprint dans un espace commun (wiki, ...)
 - Associer un « livrable » à chaque tâche

Initialiser les outils de suivi: les post-it et le burndown chart (1/3)

- Quoi?
 - Ens. de Post-it
 - associé à une tâche.
 - permet de suivre son état
 - Burndown chart:
 - graphe du travail restant à faire



From « Scrum and XP from the trenches »

Initialiser les outils de suivi: les post-it et le burndown chart (2/3)

- Qui?:
 - Le PO, la TEAM et le SM
- Quand:
 - Une fois le sprint backlog défini
 - Vont évoluer lors du suivi du sprint

Initialiser les outils de suivi: les post-it et le burndown chart (3/3)

- Conseils:
 - Ne pas hésiter à
 - inscrire uniquement les numéros des stories (1., ...) et les tâches (1.1, 1.2, ...) sur les post-it (ils sont alors récupérables ;-)
 - Établir une feuille de correspondance numéro-nom de la tâche + les durées
 - Post-it aux murs mais aussi dans cahier ss pb (comme le burndownchart)

Suivi du sprint

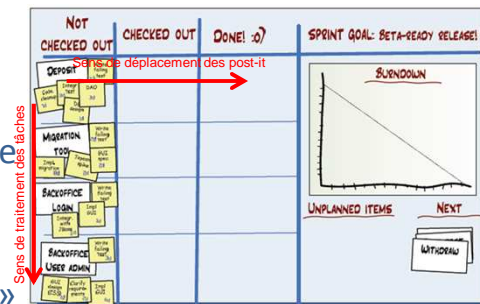
- A l'aide d'une réunion « *quotidienne* » courte:
 - Le daily scrum (ou stand-up meeting)
 - Animée par le SM
- But : identifier/détecter les pbs grâce
 - Aux échanges lors de la réunion
 - Au tableau des post-it
 - Au burndown chart
- Les pbs ne sont pas résolus pdt la réunion

Scrum meeting

- Chaque élément de la TEAM répond à 3 questions:
 1. qu'est ce que j'ai fait hier?
 2. qu'est ce que je vais faire aujourd'hui?
 3. qu'est-ce qui m'empêche d'avancer?

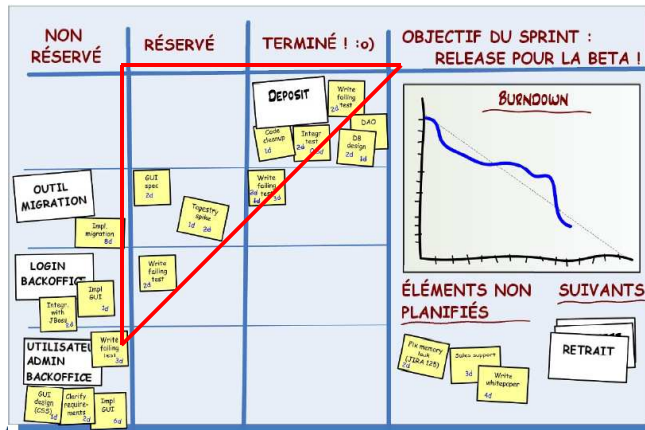
Mise à jour des post-it

- Gestion des post-it :
 - N'est « done » que si vraiment fini!
 - Ne pas modifier l'ordre des tâches
 - Détection rapide de pattern « inquiétants »



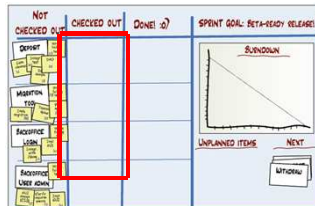
Pattern attendu

- Un triangle: signe d'un bon déroulement

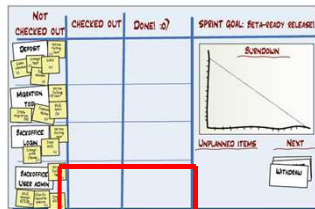


Patterns à éviter

- Le rectangle « checked out »

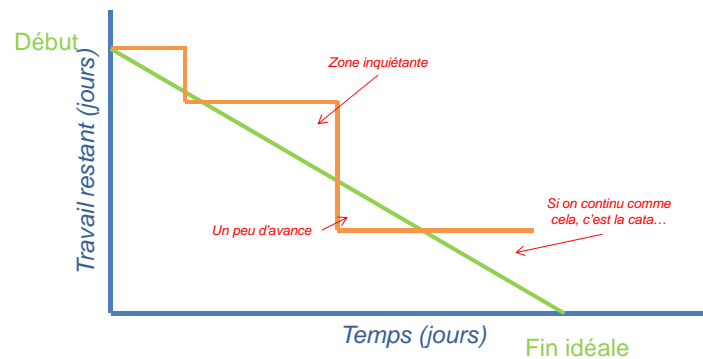


- Le rectangle bas horizontal:



Mise à jour du burndown chart

- A chaque tâche finie (vraiment...) maj le travail restant:



Rétrospective

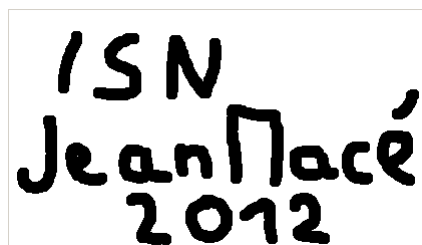
- Réunion en fin d'itération (PO, TEAM, SM)
- But: Analyser le déroulement du sprint
 - Vérifier si chaque tâche s'est déroulée comme prévue ou non (erreur d'estim., évt imprévu, ...)
 - Peut-on améliorer, par ex.,
 - Notre estimation des tâches?
 - Notre process?
- Noter les conclusions :
 - Ne pas reproduire les mêmes erreurs

Plan

- Agilité? ...
- **Allez, on essaie...**
- Résumé

Le sujet

- Les images non compressées ci-dessous occupent 99840 octets sur mon disque



- Pouvez-vous m'aider à réduire leur taille, i.e. les compresser, pour qu'elles prennent moins de taille sur mon disque?

Les rôles

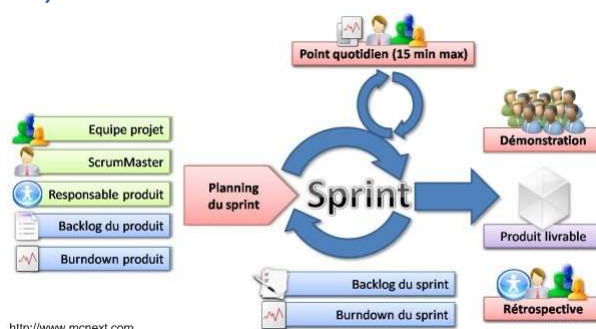
- Product Owner: un de vous (enseignant?)
- Team : les autres (élèves/étudiants?)
- Scrum Master : un de vous?... (enseignant?)

Les objectifs

- Etablir un product backlog (10')
- Initialiser un sprint (10')
- Rétrospective (sur l'exercice...) (5')
- Attention c'est « time-boxé » ;-)

Résumé

- 3 rôles, 4 actions



- Plusieurs pts non traités : vélocité, outils, ...

Résumé

- Attention:
 - L'agilité n'est pas une recette miracle
- Toutefois (retour d'expérience):
 - Améliore la communication entre les participants
 - Renforce l'adhésion au projet
 - Les cycles courts permettent de tester tôt un proto même incomplet

Questions, conseils;-)

Process
 One feature
at a time

Process
 Get the team in
a rhythm

Process
 It ain't over till
it's done


Process
 Timebox
the unknown

Process
 Honor
the timebox


Process
 Make it
visible

Process
 Eliminate
waste

Process
 Review code



INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTEMES ALÉATOIRES



20/10/2014

Pointeurs



James Shore & Shane Warden



Kent Beck
Martin Fowler



Henrik Kniberg
Préface de Jeff Sutherland, Mike Cohn



Ken Schwaber

www.agileadvice.com
www.scrumprimer.org
essentials.xebia.com
martinfowler.com
en.wikipedia.org/wiki/Agile_software_development



INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTEMES ALÉATOIRES



20/10/2014

