



# Du Chaos à l'Agilité



# Le guide du développeur en CDD

CDD : Chaos Driven Development ;)



**DON'T PANIC !**

# Qui suis-je ?

- Raphaël Rougeron
- 30 ans, marié, 1 enfant
- Développeur web à la CCIP

# Contexte

- Très petite équipe : 2 développeurs !
- Client interne
- Deadlines arbitraires
- Pas de chef de projet identifié
- Développeurs "multi-tâches" : plusieurs projets concurrents, à divers stades d'avancement

# Préambule

- Les méthodes agiles ne sont pas des religions : vous êtes libres de choisir les pratiques qui vous conviennent.
- Il y a bien sûr des contextes dans lesquels les méthodes traditionnelles (Waterfall, Spiral...), appliquées rigoureusement, fonctionnent très bien.  
Ex : logiciel de contrôle de la navette spatiale  
1 seul bug par version sur 420.000 lignes de code !

# Agile-driven

- Criticalité faible
- Développeurs expérimentés
- Besoins changeant très souvent
- Petites équipes
- Environnement chaotique

# Plan-driven

- Criticalité forte
- Développeurs inexpérimentés
- Besoins relativement stables
- Grandes équipes
- Culture de l'ordre

# Rapport CHAOS (Standish Group, 1994)

- 31,1 % de projets annulés
- 52,7 % de projets aboutis mais hors budget ou hors délais
- 16,2 % de projets réussis

# Facteurs défavorables

- politique(s), manque de soutien des managers
- priorités concurrentes
- spécifications incomplètes
- manque d'implication des utilisateurs
- manque de moyens
- attentes irréalistes (ou pas de besoins réels)
- changements des spécifications
- planification défailante

# Facteurs favorables

- implication des utilisateurs
- soutien des managers
- expression claire des besoins
- attentes réalistes
- échéances rapprochées
- vision claire des objectifs du projet
- équipe concentrée sur le projet

# Facteurs favorables

- implication des utilisateurs
- soutien des managers
- expression claire des besoins
- attentes réalistes
- échéances rapprochées
- vision claire des objectifs du projet
- équipe concentrée sur le projet

# Phase 1 : initialisation du projet

- Toutes les idées ne sont pas bonnes
- Savoir dire non
- Le sacro-saint "cahier des charges"
- 2 points-clés : vision et besoins

# Facteurs favorables

- implication des utilisateurs
- soutien des managers
- expression claire des besoins
- attentes réalistes
- échéances rapprochées
- vision claire des objectifs du projet
- équipe concentrée sur le projet



Phase 2 : Go !

# Implication des utilisateurs

Les valeurs d'XP :

- Communication
- Simplicité
- Feedback
- Courage
- Respect

# Echéances rapprochées

C'est le principe de base de la plupart des méthodes agiles pour minimiser les risques.

Chaque itération est un projet complet :

- analyse des besoins
- planning
- conception
- implémentation
- test
- documentation

# SCRUM

- itération = sprint
- mêlées quotidiennes
- tout est time-boxed
- scrum master en relation avec le client

# Liste de fonctionnalités (backlog)

| ID | Nom                   | Importance | Estimation | Description   | Notes  |
|----|-----------------------|------------|------------|---|--|
| 1  | Upload multi-fichiers | 30         | 4          | L'utilisateur doit pouvoir sélectionner plusieurs fichiers locaux et les transférer en leur appliquant des métadonnées communes           | Utiliser le plugin extjs                     |
| 2  | Fil RSS par tag       | 10         | 2          | Lorsqu'il visualise les résultats d'une recherche par tag, l'utilisateur peut cliquer sur un lien pour s'abonner au fil RSS correspondant | Limiter à 25 le nombre de photos dans le fil |

# Sprint planning : objectifs

- S'accorder sur un but pour le sprint
- Choisir une date pour présenter le résultat du sprint au client
- Obtenir des éclaircissement sur les fonctionnalités souhaitées
- Estimer le nombre de points (jours/hommes idéaux) par fonctionnalité
- Sélectionner les fonctionnalités à implémenter durant le sprint

#1

# Upload multi-fichiers

Importance

30

Descriptif

L'utilisateur doit pouvoir sélectionner plusieurs fichiers locaux et les transférer en leur appliquant des métadonnées communes

Estimation

Notes

Utiliser le plugin extjs

# Estimations et vélocité

- Raphaël + Jérémie = 28 jours/hommes
- Précédent facteur de concentration : 40 %  
(12 points / 30 jours)
- Vélocité estimée :  $40\% \times 28 = 11,2$
- Le total des estimations des fonctionnalités sélectionnées ne doit pas dépasser 11 !

# Les pratiques d'XP

- Programmation en binôme
- Jeu du Planning
- TDD
- Client sur site
- Intégration continue
- Refactoring
- Livraisons fréquentes
- Conventions de nommage
- Appropriation collective du code
- Conception simple
- Utilisation de métaphores
- Rythme soutenable

# TDD : les outils

- SimpleTest
- PHPUnit
- un nouveau venu : PHPSpec

# TDD : notre approche

- Tests unitaires couvrant la couche métier
- Approche "top-down", plutôt BDD en fait
- Tests fonctionnels uniquement pour certaines fonctionnalités "critiques" (authentification, ...)

# Intégration continue

- VCS !
  - Subversion + Trac + TortoiseSVN  
= le tiercé gagnant
  - l'intégration quotidienne des modifications impose une certaine discipline (ne pas se disperser)
- 3 environnements : dev, recette et prod
- Utilisation expérimentale de Capistrano :  
<http://devthatweb.com/view/deploy-any-project-using-capistrano-2>

# PHP est un langage agile !

- Simplicité
- Capacités d'intégration
- Bibliothèques, frameworks