Les Méthodes Agiles

description et rapport à la Qualité

Benjamin Joguet
Rémi Perrot
Guillaume Tourgis

Plan

Pourquoi Agile?

- Cycles de développement courts
- Spécifications des clients «volatiles»

⇒ Méthodes lourdes inadaptées

- 2001 : «Agile Alliance» définit les méthodes Agiles

But : Augmenter le niveau de satisfaction des clients tout en rendant le travail de développement plus facile

Qu'est ce qu'une méthode Agile ?

- 2 caractéristiques fondamentales :
  - «Adaptatives» plutôt que prédictives
    - favorables aux changements
    - planification plus souple
  - Orientées vers les personnes plutôt que vers les processus
    - travailler avec les spécificités de chacun
    - responsabilité

Le manifeste

Nous découvrons de meilleures façons de développer des logiciels en le faisant et en aidant les autres à le faire

Personnes et interactions

Applications fonctionnelles

Collaboration avec le client

Acceptation du changement

Par dessus

Procédures et outils

Documentation pléthorique

Négociation de contrat

Planification

Éléments privilégiés

Les valeurs

- Priorité aux **personnes** et aux **interactions** par rapport aux **procédures** et aux **outils**
  - *Travail en groupe, communication*

- Priorité aux **applications fonctionnelles** par rapport à une **documentation pléthorique**
  - *Documentations succinctes à jour, documentation permanente du code*

- Priorité de la **collaboration avec le client** par rapport à la **négociation de contrat**
  - *Feedback régulier du client, solution répondant réellement aux attentes*
  - *Grande maturité du client, relation de confiance*

- Priorité de l'**acceptation du changement** par rapport à la **planification**
  - *Planning flexible, modifications possibles après 1ère version du système*

Les principes

4 valeurs déclinées sur 12 principes généraux :

- Notre plus grande priorité est de satisfaire le client en lui livrant **très tôt** et **régulièrement** des **versions fonctionnelles** de l'application source de valeur
  - *client peut décider de la mise en production de l'application*

- **Accueillir les demandes de changement à bras ouverts**, même tard dans le processus de développement. Les méthodologies agiles exploitent les changements pour apporter au client un avantage concurrentiel
  - *Produire des systèmes flexibles*

- Livrer **le plus souvent possible** des **versions opérationnelles** de l'application, avec une fréquence comprise entre deux semaines et deux mois, avec une préférence pour l'échelle de temps la plus courte
  - *Objectif : livrer une application qui satisfasse aux besoins du client*

Les principes (2)

- Clients et développeurs doivent **coopérer quotidiennement** tout au long du projet
- Construire des projets autour d'**individus motivés**. Leur donner l'environnement et le support dont ils ont besoin et leur **faire confiance** pour remplir leur mission
- La méthode la plus efficace pour **communiquer des informations** à une équipe et à l'intérieur de celle-ci reste la conversation en **face à face**
- Le fonctionnement de l'application est le **premier indicateur** d'avancement du projet
- Les méthodes agiles recommandent que le projet avance à un **rythme soutenable** : développeurs et utilisateurs devraient pouvoir maintenir un rythme constant indéfiniment
    - *Adapter le rythme pour préserver la qualité du travail sur la durée du projet*

Les principes (3)

- Porter une attention continue à l'**excellence technique** et à la **conception** améliore l'agilité
  - *Maintenir le code source propre, clair et robuste*

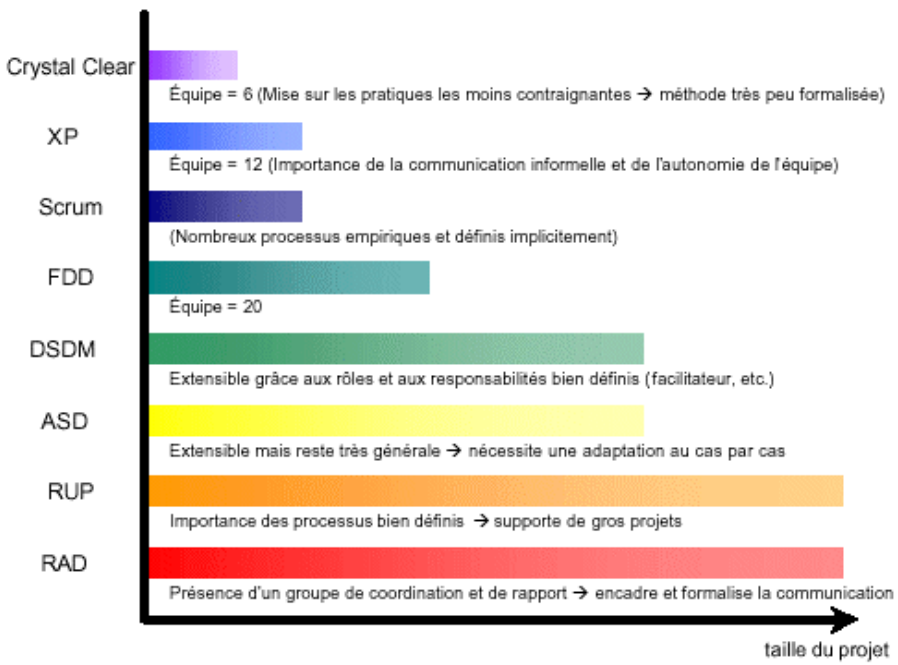
- La **simplicité**, art de maximiser la quantité de travail à ne pas faire, est **essentielle**
  - *Répondre le + simplement aux besoins actuels pour que celui ci soit adaptable*

- Les meilleures architectures, spécifications et conceptions sont le fruit d'**équipes** qui **s'auto-organisent**
  - *Partage des responsabilités par volontariat*

- A intervalles de temps réguliers, l'ensemble de l'**équipe s'interroge** sur la manière de devenir encore plus efficace, puis **ajuste son comportement** en conséquence
  - *Environnement en perpétuelle évolution*

Les méthodes Agiles

- De nombreuses méthodes, sans compter les méthodes maisons
- Selon la taille du projet / de l'équipe



eXtreme Programming

- Génèse
  - Initiative de Kent Beck en 1996
  - Projet vitrine à la DRH de Chrysler
- Principes
  - Pousser à l'extrême les bonnes pratiques de génie logiciel
  - 4 valeurs de bases -> 12 pratiques
  - Suit complètement le manifeste Agile

eXtreme Programming

Les 4 valeurs fondamentales

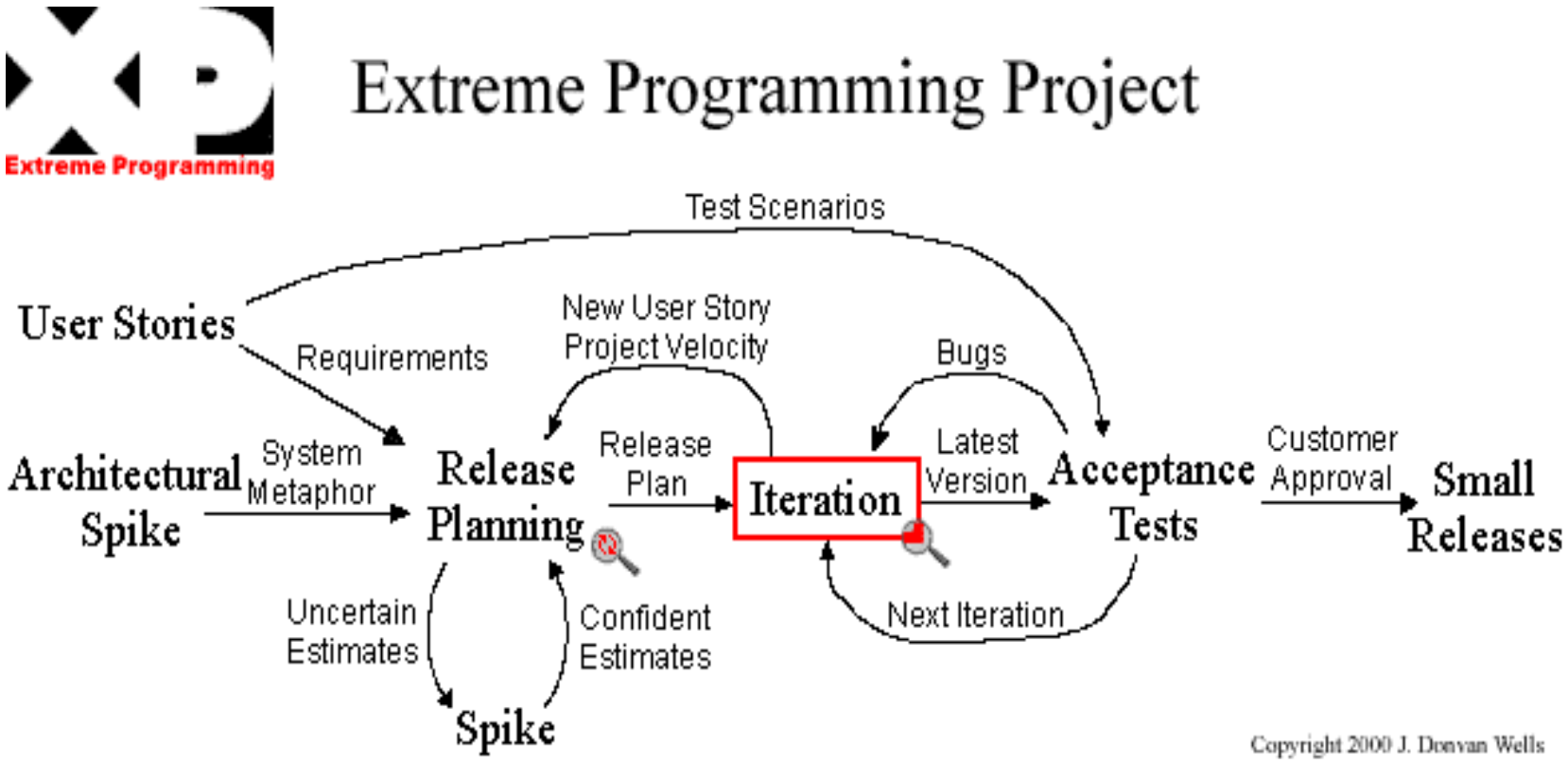
- **Communication**
  Tout doit être pretexte à communiquer. Priviligier les échanges en face à face.
- **Simplicité**
  Faire simple avant tout
- **Rétroaction (feedback)**
  Utiliser les expériences passées, et les remarques du client le plus possible
- **Courage**
  Il faudra peut-être jeter des semaines de travail pour suivre de nouvelles exigences client

eXtreme Programming

Les 12 principes

- Planification
- De petites mises à jour
- Métaphore système
- Design simple
- Tests permanents
- Factoriser
- Programmation en binôme
- Propriété collective du code
- Intégration continue
- 40 heures de travail/semaine
- Client sur le site
- Standards d'écriture

eXtreme Programming



eXtreme Programming

Limites et évolutions

- Limites

  Sans application rigoureuse de toutes les pratiques, pas de bons résultats

  - Pratiques non conformes à la culture d'entreprise (client sur site, binôme)

- Evolution

  - Tranformation en méthodes maisons par changement réfléchit de certaines pratiques
  - Nouvelles méthodes dérivées : Code Science

Agilité et Qualité Logiciel

La méthodologie Agile :

- « Un encouragement au bricolage »

  ou

- « Une méthode d'une efficacité miraculeuse » ?

XP et SW-CMM

SW-CMM :

- solutions de management
- améliorations systématiques définies

XP :

- solutions pour petites équipes
- besoins très « évolutifs »

XP et SW-CMM niv.2

XP gère :

- planning
- supervision projet
- gestion des exigeances

XP ne gère pas :

- sous-traitance
- assurance qualité
- gestion de configuration

=> insuffisant

XP et SW-CMM niv.3

XP gère :

- ingénierie de produit logiciel
- coordination intergroupes
- revue par ses pairs
- focalisation organisationnelle sur les processus

XP ne gère pas :

- gestion intégrée du logiciel

non explicite :

- formation
- définition /amélioration processus de l'organisation

XP et SW-CMM Conclusion

- tenir compte des spécificités : pas ou peu d'organisation management, travail coopératif
- pas du bricolage, fonctions vitales de l'ingénierie logiciel couvertes

=> synergie théoriquement envisageable

Agile et normes

- ISO 12207 : non
- ISO 9126 : oui, avec élargissement code => modélisation
- TickIt Guide : DSDM conforme à ISO 9001 => DSDM pont vers la certification ISO

Retour d'expériences

- résultats sondage (Bernhard Rumpe & Astrid Schröder - Munich University of Technology)

- satisfactions :
  - propriété commune du code, intégration/tests continus
  - programmation par paire, focus sur les « bons » buts
- problèmes :
  - adaptation management, développeurs, entreprise
  - métaphores, on-site customer

Conclusion Agile : une mode ?

- Semble adapté à un besoin :
  - pas de sous-traitance
  - équipe motivée
  - environnement non-critique
  - équipe légère
  - projet non complexe

- Solution d'urgence :
  - objectif satisfaction client immédiate
  - objectif risque/coût minimal
  - supporter les imprévus

- Du point de vue démarche qualité :
  - en cohérence limitée
  - pas de certifications encore en phase

- cependant, indispensable de connaître Agile