

# Les méthodes agiles

## Chapitre 4

[ 1 ]

## Chap.4 : Les méthodes agiles

Mots et concept clés : Qu'est ce que l'agilité ?,  
Itération, Sprint, User Story, Product backlog,  
Product owner

[ 2 ]

# Introduction : à l'origine des méthodes agiles ...

- Des projets « en V » qui aboutissaient à des échecs cuisants !
- Développement de méthodes de gestion alternatives dans les années 1990 et publication du « manifeste agile » en 2001
- Les méthodes agiles sont des méthodes de gestion de projet, moins structurées et plus légères que les cycles en cascade, censées donner de l'agilité pour contourner les obstacles et s'adapter aux particularités de chaque projet (d'après : Chantal Morley, Management d'un projet système d'information, Dunod, 7<sup>ème</sup> édition).

[ 3 ]

## I. Le mouvement agile

### A. Le manifeste agile

### B. Quelques modèles phares

1. Scrum
2. XP

[ 4 ]

# A. Manifeste pour le développement Agile de logiciels

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

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,
- Les 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.

[ 5 ]

## Principes sous-jacents au manifeste

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

*Nous suivons ces principes:*

- *Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.*
- *Livrez fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.*
- *Un logiciel opérationnel est la principale mesure d'avancement.*
- **Itérations courtes qui permettent de produire un livrable à chaque fois opérationnel.**
- *Accueillez positivement les changements de besoins, même tard dans le projet. Les processus Agiles exploitent le changement pour donner un avantage compétitif au client.*
- **Les besoins du client peuvent changer.**
- *Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.*
- **Collaboration MOE/MOA**
- *La méthode la plus simple et la plus efficace pour transmettre de l'information à l'équipe de développement et à l'intérieur de celle-ci est le dialogue en face à face.*
- **Privilégier l'oral, tenir compte des connaissances tacites, favoriser l'informel, ne pas tout formaliser.**

[ 6 ]

# Principes sous-jacents au manifeste

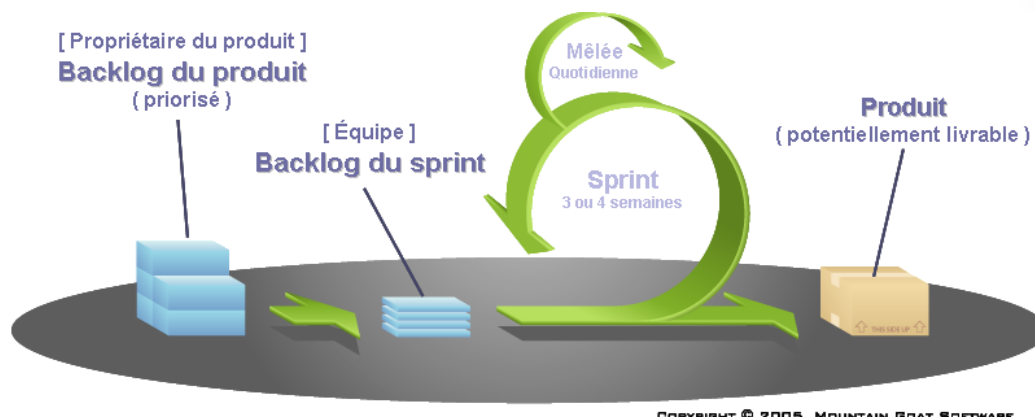
Source : <http://agilemanifesto.org/iso/fr/>

- Réalisez les projets avec des personnes motivées. Fournissez-leur l'environnement et le soutien dont ils ont besoin et faites-leur confiance pour atteindre les objectifs fixés.
  - Les processus Agiles encouragent un rythme de développement soutenable. Ensemble, les commanditaires, les développeurs et les utilisateurs devraient être capables de maintenir indéfiniment un rythme constant.
  - Une attention continue à l'excellence technique et à une bonne conception renforce l'Agilité.
  - La simplicité – c'est-à-dire l'art de minimiser la quantité de travail inutile – est essentielle.
  - Les meilleures architectures, spécifications et conceptions émergent d'équipes auto-organisées.
- Une vision du travail différente du modèle taylorien usuel : motivation des développeurs, rythme soutenable, équipes auto-organisées + excellence technique, qualité de l'architecture et simplicité des développements.
- À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence.
- Amélioration continue

[ 7 ]

## B. Quelques méthodes phare

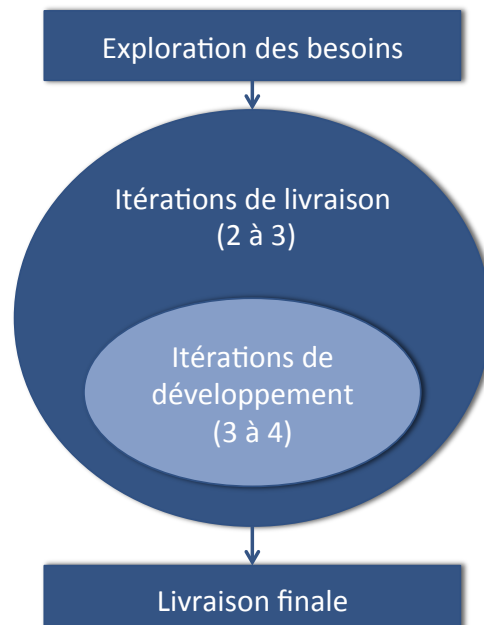
### 1. Scrum :



[ 8 ]

## B. Quelques méthodes phare

### 2. XP : eXtrem Programming



[ 9 ]

## II. Le recueil des besoins agile

1. Il faut avant tout partager une vision
2. Principe fondamental n°1 : Les besoins émergent petit à petit
3. Principe fondamental n°2 : La hiérarchisation des besoins

[ 10 ]

# 1. Il faut avant tout partager une vision

(parce que les méthodes classiques font souvent oublier l'objectif principal)

- Un acteur clé le **Product Owner** (ou chef de projet utilisateur)
  - Il communique la vision, décrit les fonctionnalités, connaît les priorités, peut changer d'avis ;
  - Il collabore, est exigeant, soutient l'équipe et donne de la reconnaissance.

[ 11 ]

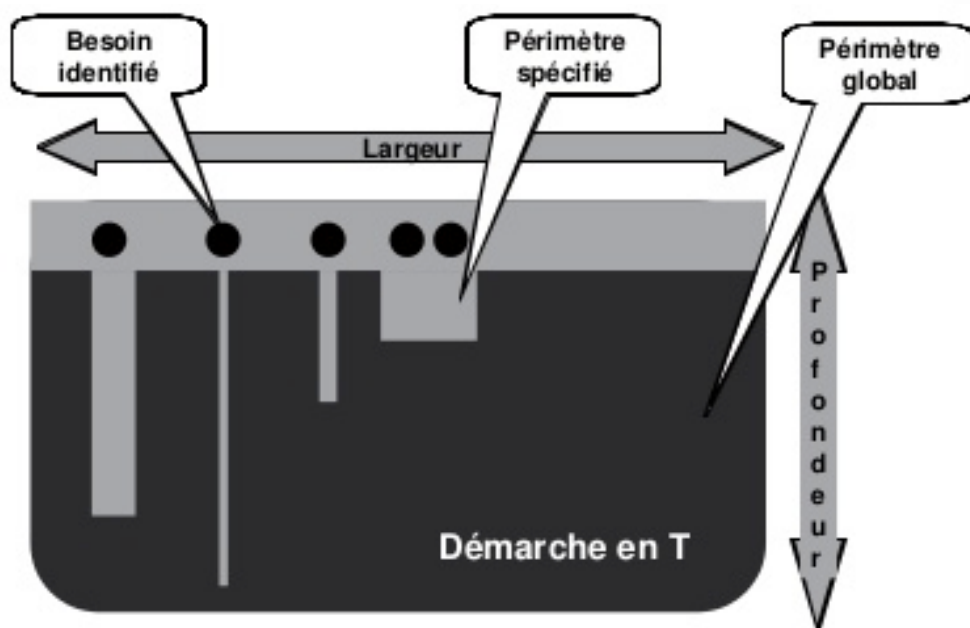
## 2. Principe fondamental n°1 : Les besoins émergent petit à petit

- Pourquoi ?
  - Le marché est instable...
  - Le client découvre petit à petit l'outil qui se construit...
  - Les choix à opérer font évoluer les priorités...
- Comment ?
  - Démarche itérative de recueil-hiérarchisation-réalisation
  - On parle de démarche en T

[ 12 ]

# Recueil des besoins : démarche en T

Source : gestion de projet agile, Véronique Messenger, Eyrolles 2013



[ 13 ]

## 3. Principe fondamental n°2 : La hiérarchisation des besoins

- « Développer les fonctionnalités par ordre de priorité »
  - Critères de priorisation :
    - Bénéfice attendu,
    - Coût de développement,
    - Opportunités d'apprentissage pour l'équipe,
    - Risques\*.
  - 3 modèles : La matrice de Kano, Les poids relatifs de Wiegers, la méthode MoSCoW.
- ! C'est le product owner (et/ou les utilisateurs) qui priorise, pas l'équipe de développement !

[ 14 ]

## II. Le recueil des besoins : conclusion

- Le recueil des besoins se fait pas à pas,
- Le changement est accepté,
- Les besoins sont précisés au fil du projet,
- Les besoins sont consignés et priorisés sous forme de **USER STORIES** dans le **PRODUCT BACKLOG**.

[ 15 ]

## User story

- Exigence du système à développer,
- Formulée en 1 ou 2 phrases (par exemple sur un post-it),
- Dans le langage de l'utilisateur.
- Elle doit permettre :
  - d'estimer le coût de développement,
  - de réaliser la fonctionnalité au cours d'une itération.
- Pas de formalisme imposé mais...
  - En tant que <rôle>
  - Je veux <fonctionnalité>
  - Afin de <bénéfice>

[ 16 ]



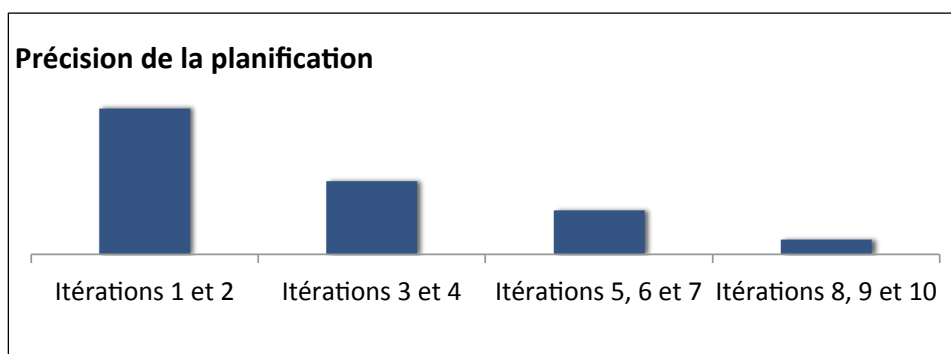
### III. La planification agile

- ① Principe de base : Une planification précise à CT, grossière et incertaine à LT
- ② Évaluation de l'enveloppe globale
- ③ Les 5 niveaux de planification

[ 17 ]

- ① Principe de base : Une planification précise à CT, grossière et incertaine à LT

- Les besoins changent et évolue...
- Inutile de planifier à LT (au delà d'un mois l'incertitude est trop forte), on avance pas à pas, **on ne planifie avec précision que l'étape (l'itération) suivante.**

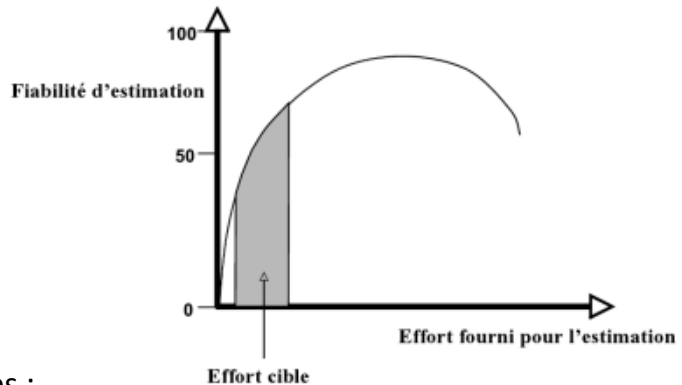


- + **À la fin de chaque étape on capitalise l'expérience** avant de planifier l'étape suivante.

[ 18 ]

## ② Évaluation de l'enveloppe globale

- Pour démarrer le projet et contractualiser, sans gaspiller du temps à tenter de réaliser une estimation précise.



- Deux méthodes préconisées :
  - a. La méthode Delphi (par analogie)
  - b. La méthode des story points

! Le client doit savoir que cette enveloppe lui garantit un nombre d'itérations... mais pas la liste des fonctionnalités qu'il a fourni !

[ 19 ]

## ③ Les 5 niveaux de planification

- La visions du produit : le product backlog
- La road map
- Le plan de release
- Le plan de l'itération
- Le cycle quotidien

[ 20 ]

## IV. Le suivi de projet agile

[ 21 ]

## Conclusion

- Quelques définitions :
  - Méthode agile :
  - Itération :
- Ouverture : pourquoi ne pas sortir du débat stérile  
« méthodes classiques » et « méthodes agiles » ? La gestion  
est une discipline contingente : à chaque contexte sa méthode  
appropriée !

[ 22 ]

## I.3. Recueil et formalisation du besoin : quelques techniques

- Analyse en largeur
  - « Brainstorming »
  - « Benchmark »
- Analyse en profondeur et description détaillée \*
  - Pour commencer : Interview (technique des 9 cases) et « Workshop »
  - Pour affiner avant de développer : « Gap analysis » et Observation de l'utilisateur « in situ »
- Formalisation
  - Cas d'utilisation UML (cf. cours UML)
  - ou User stories
  - Les Cas d'utilisation ou les User Stories sont synthétisés dans le **product backlog**

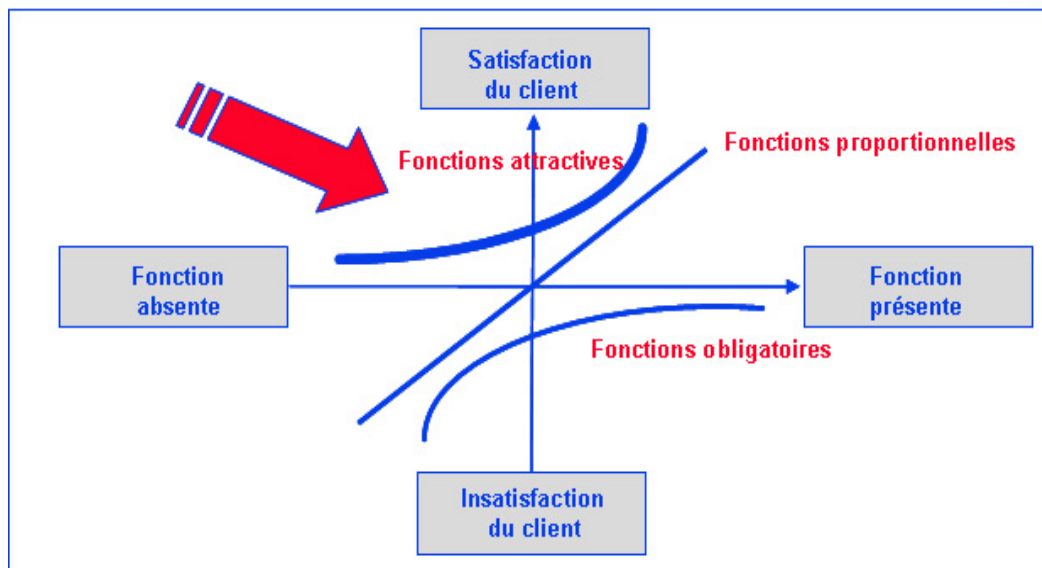
[ 23 ]

## Solution de Selling ou technique des 9 cases

	Quel est le Problème ?	Qui est Impacté ?	Visualiser la solution
<b>Question ouverte</b>			
« Dites-moi » « Racontez-moi » « Et puis »	1	4	7
<b>Contrôle</b>			
Combien ? Quand ? Où ?	2	5	8
<b>Confirmation</b>			
« Si je comprends bien... si oui,... si non ... »	3	6	9

[ 24 ]

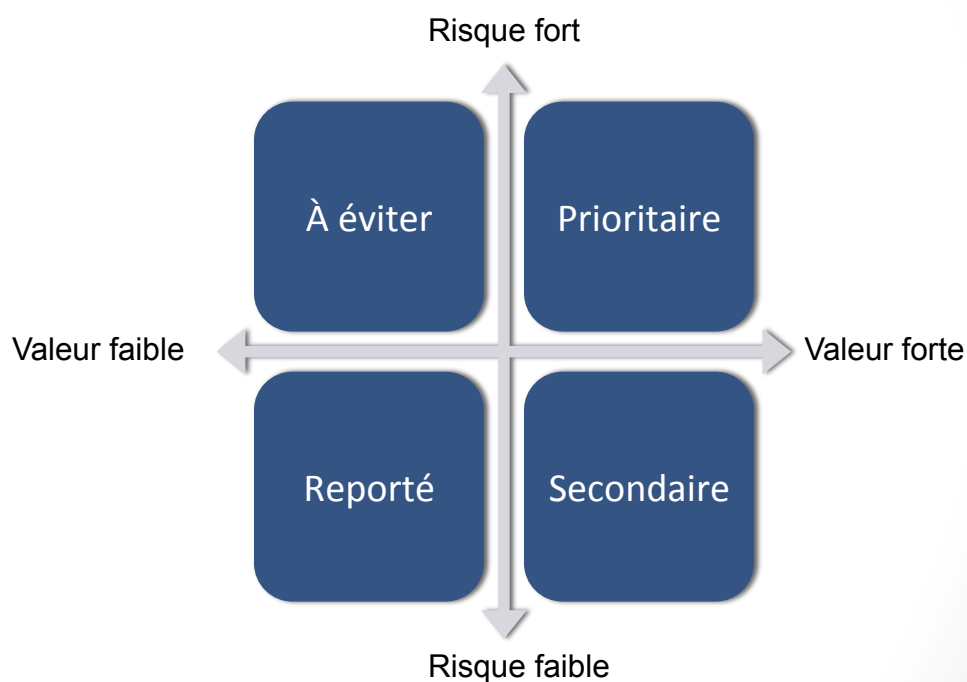
# Matrice de Kano



[ 25 ]

# Hiérarchisation en fonction des risques et de la valeur

Source : A. Mike Cohn, Agile estimating and planning, Prentice Hall, 2004



[ 26 ]

## ① Principe de base : Une planification précise à CT, grossière et incertaine à LT

- Conséquences importantes sur la relation client fournisseur :
  - Le contrat prévoit une enveloppe globale contenant un certain nombre d'itérations.
  - Quand les itérations sont consommées, le projet s'arrête même si toutes les fonctionnalités envisagées au démarrage n'ont pas été développées.
  - Mais le produit livré est un produit fini, livré avec les fonctionnalités les plus importantes pour le client, qui a pu s'ajuster à l'évolution des besoins au fil de sa création.
  - Le client accepte donc une incertitude contre un produit répondant mieux (?) à ses besoins.

[ 27 ]

## Principes sous-jacents au manifeste

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

*Nous suivons ces principes:*

- *Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.*
- *Livrez fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.*
- *Un logiciel opérationnel est la principale mesure d'avancement.*
- *Accueillez positivement les changements de besoins, même tard dans le projet. Les processus Agiles exploitent le changement pour donner un avantage compétitif au client.*
- *Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.*
- *La méthode la plus simple et la plus efficace pour transmettre de l'information à l'équipe de développement et à l'intérieur de celle-ci est le dialogue en face à face.*
- *Réalisez les projets avec des personnes motivées. Fournissez-leur l'environnement et le soutien dont ils ont besoin et faites-leur confiance pour atteindre les objectifs fixés.*
- *Les processus Agiles encouragent un rythme de développement soutenable. Ensemble, les commanditaires, les développeurs et les utilisateurs devraient être capables de maintenir indéfiniment un rythme constant.*
- *Une attention continue à l'excellence technique et à une bonne conception renforce l'Agilité.*
- *La simplicité – c'est-à-dire l'art de minimiser la quantité de travail inutile – est essentielle.*
- *Les meilleures architectures, spécifications et conceptions émergent d'équipes auto-organisées.*
- *À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence.*

[ 28 ]