

# Agile : la démonstration manquante

*Olivier Gourment*

*Agile Tour Montréal 2010*

# Olivier Gourment

- ~ Vingtaine d'années d'expérience en TI
- ~ Cheminement professionnel :  
développeur, architecte, chef d'équipe,  
chargé de projet
- ~ Maintenant directeur du développement  
logiciel chez Adenyo Telecom Mobile
- ~ Pratique Agile depuis 2001 (XP). Certified  
ScrumMaster, Scrum Practitioner, PMP

# Olivier Gourment

- ~ Membre du comité organisateur Agile Tour Montréal 2010
- ~ Cofondateur du groupe des utilisateurs de Scrum Montréal [www.scrumusergroup.ca](http://www.scrumusergroup.ca)
- ~ Courriel : [ogourment@smarterportal.com](mailto:ogourment@smarterportal.com)
- ~ LinkedIn : [www.linkedin.com/in/ogourment](http://www.linkedin.com/in/ogourment)
- ~ Twitter : <http://twitter.com/ogourment>
- ~ Blog : <http://ogourment.wordpress.com>

# Le manifeste Agile

Écrit il y a presque 10 ans par des passionnés (plus de la moitié de programmeurs), experts reconnus (livres sur l'orienté objet, etc ...)

[www.agilemanifesto.org](http://www.agilemanifesto.org)

# Le manifeste Agile

**“Les individus et les interactions ...**

plus important que les processus et les outils

**Du logiciel qui fonctionne ...**

plus important que de la documentation exhaustive

**Collaborer avec le client ...**

plus important que négocier un contrat

**Répondre au changement ...**

plus important que suivre le plan”

# Le manifeste Agile

Euh ... ça veut dire quoi précisément ?

- ~ Je fais de la doc ou j'en fais pas ??
- ~ J'utilise des outils ou non ???
- ~ Est-ce que je garde des choses de ma métho ???
- ~ Je fais quoi si j'ai signé un contrat ????
- ~ Je mets quoi dans mon WBS ?????

# Le manifeste Agile

- ~ Les 12 principes donnent davantage de détails ... mais restent encore assez abstraits et ne s'appliquent pas à tous les contextes.
- ~ Exemples :
  - ~ absence de client (R&D)
  - ~ pas de code produit

Des questions valides sans réponses

...

**VOLONTAIREMENT**

...

Car chaque contexte est différent !



Il faut donc comprendre le  
**RAISONNEMENT** qui a conduit ces  
experts à écrire ces valeurs.

C'est la seule façon de les interpréter pour  
un contexte particulier.

# Exercices

Exercice 1 : produire des estimés

ou

Exercice 2 : trouver et corriger un bug

**NOTE IMPORTANTE : Les exercices  
sont INDISPENSABLES À LA  
COMPRÉHENSION de la suite !**

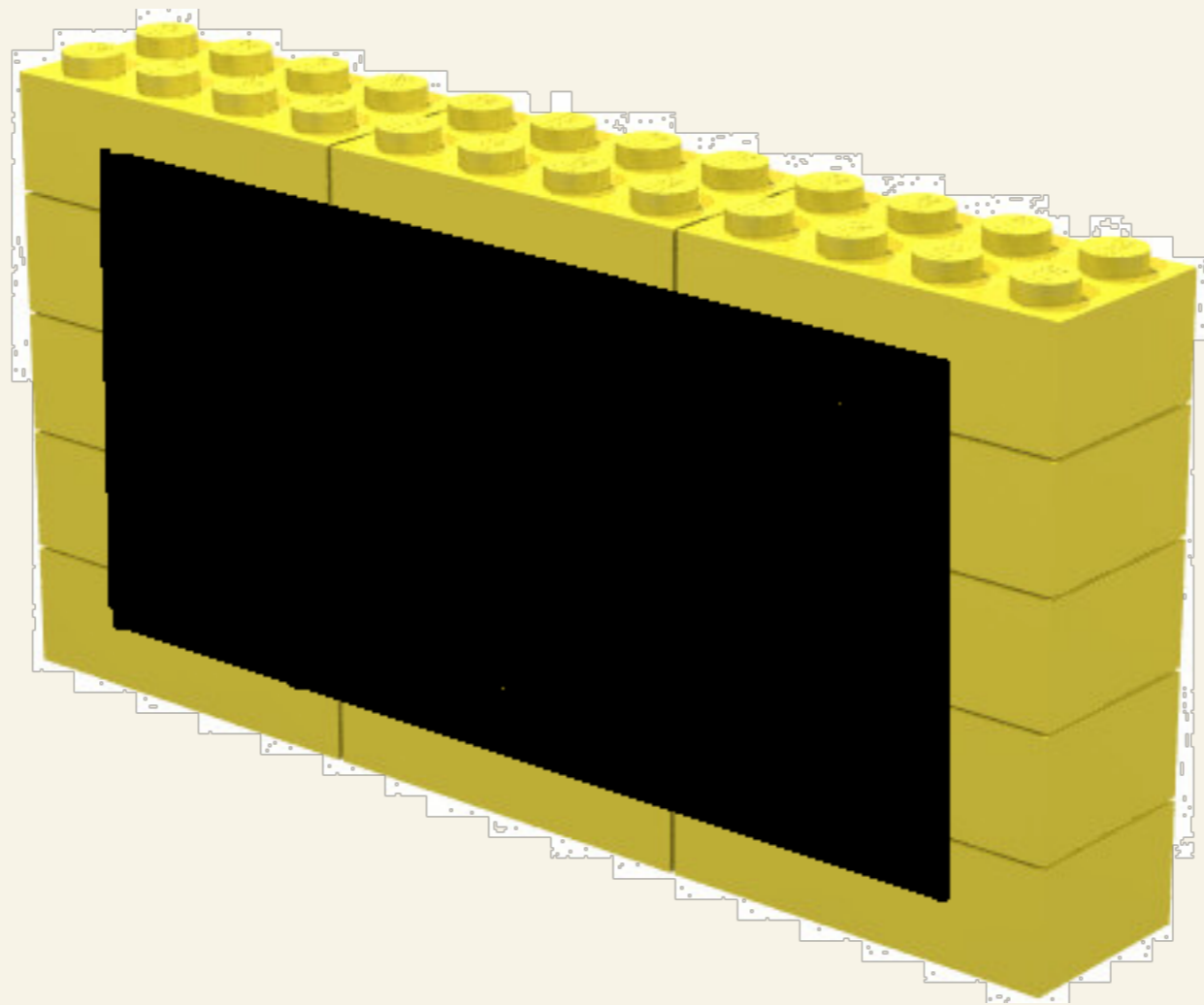
# Exercices



# Démonstration

(Démonstration impliquant de la  
programmation en direct)

# Démonstration



# Démonstration

- ~ Ce mur est-il bien fait ?
- ~ Comment le savoir ?
- ~ Qu'entend-on par "bien fait" ?
  
- ~ Qu'est-ce qu'on peut en conclure dans le cas d'un projet logiciel ?

# C'est quoi Agile ?

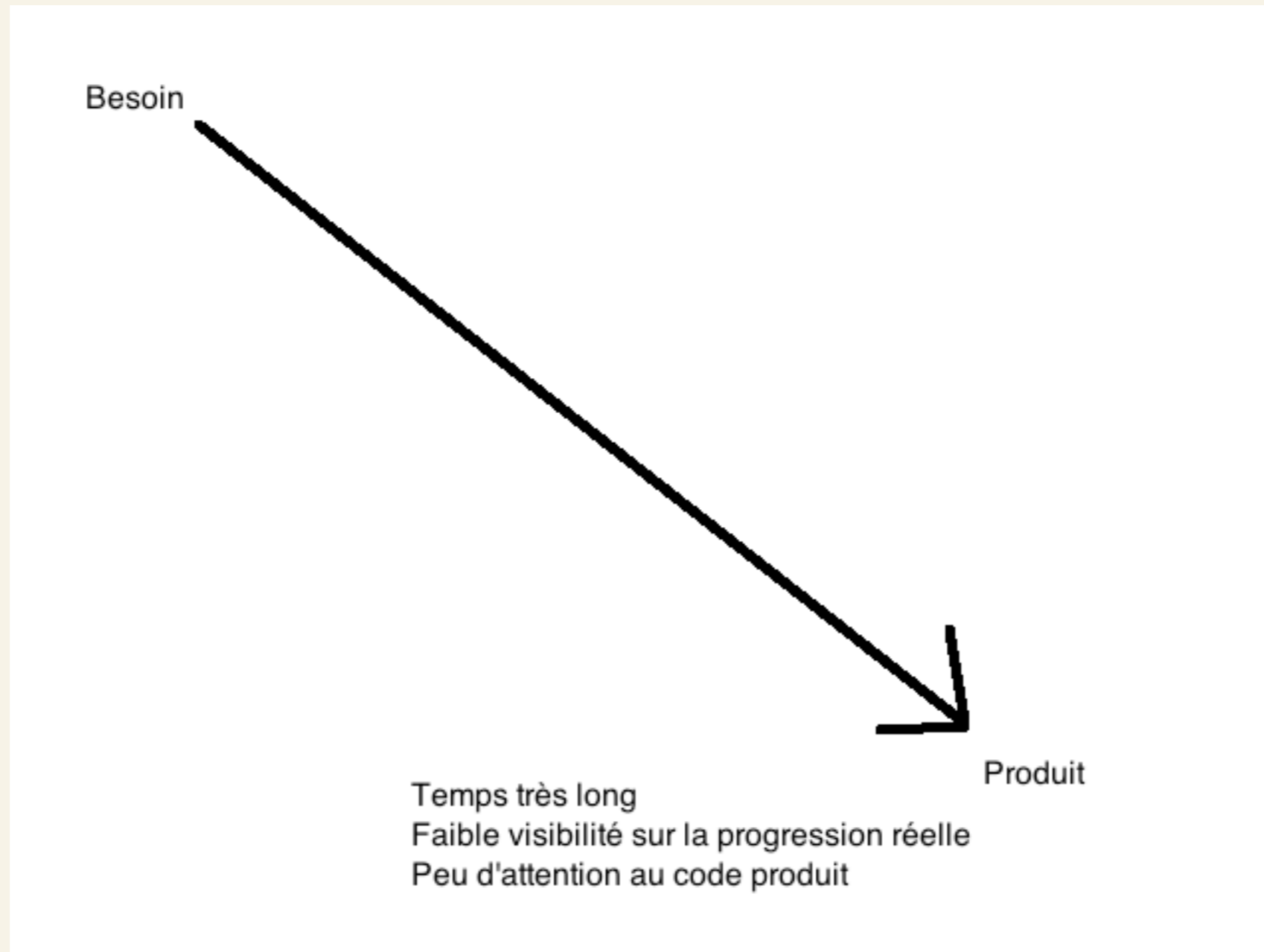
- ~ Les 3 premiers principes sont clairs :  
le but est de **LIVRER RAPIDEMENT**  
et **FRÉQUEMMENT** de la **VALEUR**.

# C'est quoi Agile ?

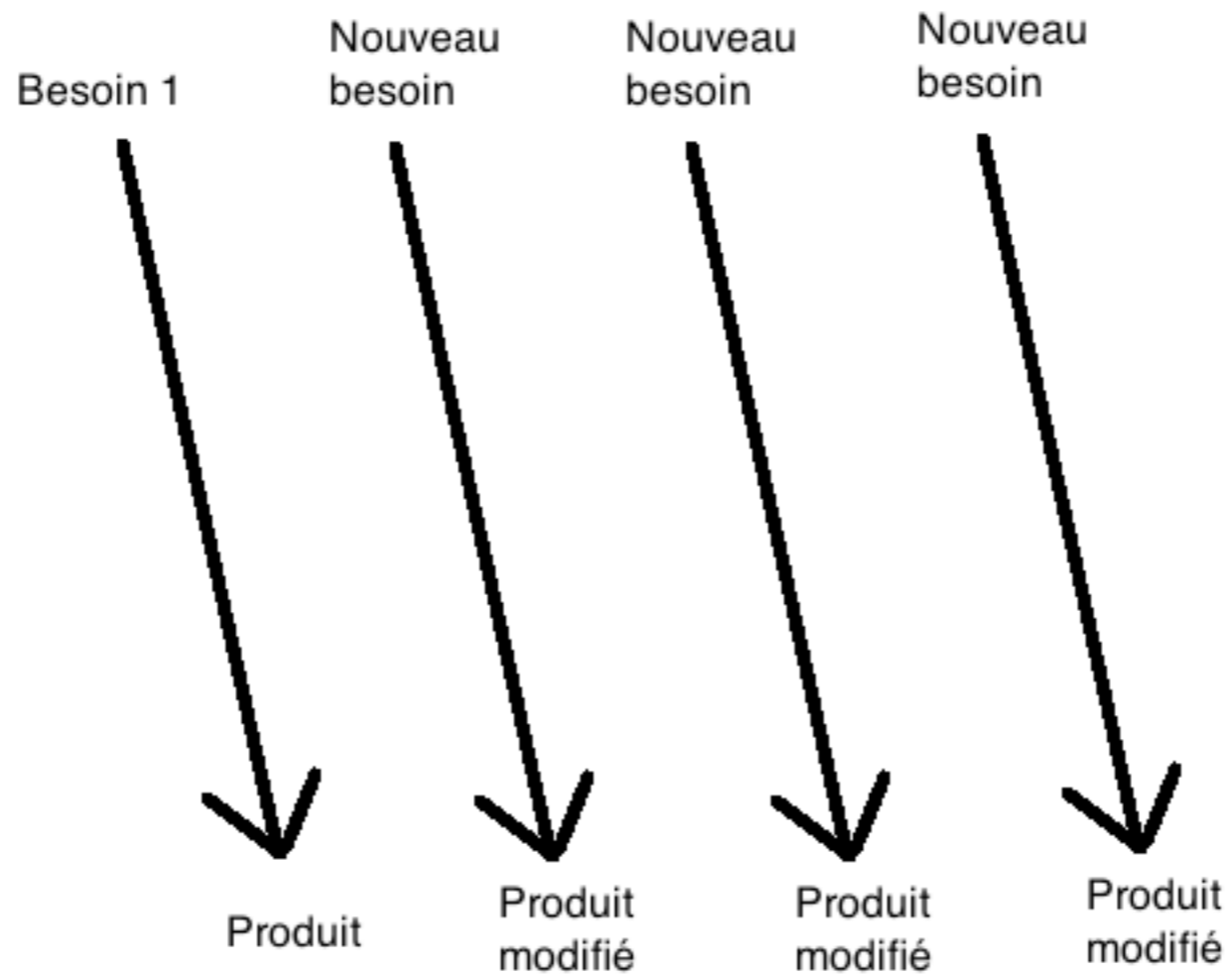




# Non Agile



# Agile



Livraison rapide  
Transfert de connaissance optimisé  
Attention à la facilité de modifier le code

# C'est quoi Agile ?

- ~ Les principes 4 à 12 évoquent les moyens interdépendants d'y parvenir :
1. **FACILITER L'APPRENTISSAGE** des équipes de réalisation (livrer par itération et de façon incrémentale)
  2. Exécuter avec **QUALITÉ** les livrables techniques (code ou autre). La **RAPIDITÉ** de modification est la clé !

# Les 12 principes (1/3)

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.

# Les 12 principes (2/3)

5. 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.

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

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

# Les 12 principes (3/3)

9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

# CQFD

Questions ?

[ogourment@smarterportal.com](mailto:ogourment@smarterportal.com)