agile tour 2011

Ne renvoyez pas vos architectes ! Utilisez-les avec agilité

Intégration du travail architectural dans un cycle de développement Agile

Jean-Louis Maréchaux

jl.marechaux@ca.ibm.com

Qui suis-je ?

jl.marechaux@ca.ibm.com

Jean-Louis Maréchaux

IBM Canada Lab / Rational software

- Product Development & Support / User Technologies

Rôles

- Collaboration avec l'équipe de développement des outils
  - ✓ Faciliter l'adoption des produits et des technologies
    1. Externe (clients)
    2. Interne (IBM)

    → Matériel de formation, articles techniques, conférences, démos, preuves de concept...
  - ✓ Architecture et développement
- Équipe « Agile Community Leadership »

Expérience

- Plus de 15 ans... Développement, architecture et méthodologies
- Petites et grandes organisations, consultation
- Equipes locales et distribuées (www.distributedscrum.com)

Introduction: Les difficultés de l'architecte

- Adoption d'une méthodologie agile
  - Nouvelle organisation du travail
  - Nouvelle terminologie

    → Confusion sur les rôles dans les projets Agiles

- Architecture: un seul terme, plusieurs nuances
  - Architecture logicielle, de solutions, fonctionnelle, des données, d'entreprise...
  - La frontière entre l'architecture et le design est difficile à définir
  - La discipline d'architecture est souvent mal comprise

- Mythes sur l'architecture et les méthodologies agiles
  - Rôle de l'architecte est mal compris / mal défini

- **Objectif de la session** : Comprendre comment le travail architectural s'intègre dans un cycle de développement agile
  - Transition vers agile
  - Équipes agiles qui travaillent avec des architectes

Agenda

- Introduction
- **> Mythes: architecture et agilité**
- Rôles et cycles agiles
- Qu'est ce que l'architecture?
- L'architecte « agile »
- Conclusion

Mythe: L'architecture ne fait pas partie des pratiques Agiles

Références reconnues du monde Agile et la pratique d'architecture

- **Agile Manifesto – 2001**
  Principes #9 et #11
- **Kenn Beck**
  Extreme Programming Explained, Addison Wesley 2005, p74 / p52
- **Mike Cohn**
  Succeeding with Agile, Addison Wesley 2009, p166
- **Scott Ambler**
  Architectural Envisioning: An Agile Best Practice, Ambysoft 2004-2011
- **Robert Martin (Uncle Bob)**
  The scatology of Agile architecture, Object Mentor blog 2009

- **James Shore**
  The art of Agile development, O'Reilly 2007, chapter 3
- **Ron Jeffries**
  Extreme Programming Installed, Addison Wesley 2007, p70
- **Mary Poppendieck**
  Implementing Lean Software Development, Addison Wesley 2006, p182
- **Alistair Cockburn**
  Agile Software Development, the cooperative game, Addison Wesley 2007, p252

Et bien d'autres...
(voir Références)

Mythe: Les équipes Agiles ne font pas de modélisation

- Enquête Dr Dobb's sur la modélisation et la documentation (2008)
  - 93% des équipes Agiles font du « whiteboard modeling »
  - 77% des équipes Agiles anticipent quelques besoins d'architecture
  - 66% des équipes Agiles font de la modélisation sur papier
  - 47% des équipes Agiles utilisent un outils électronique pour leur modélisation

- Non seulement les équipent Agiles font de la modélisation, mais elles en font même plus que les équipes « traditionnelles ».

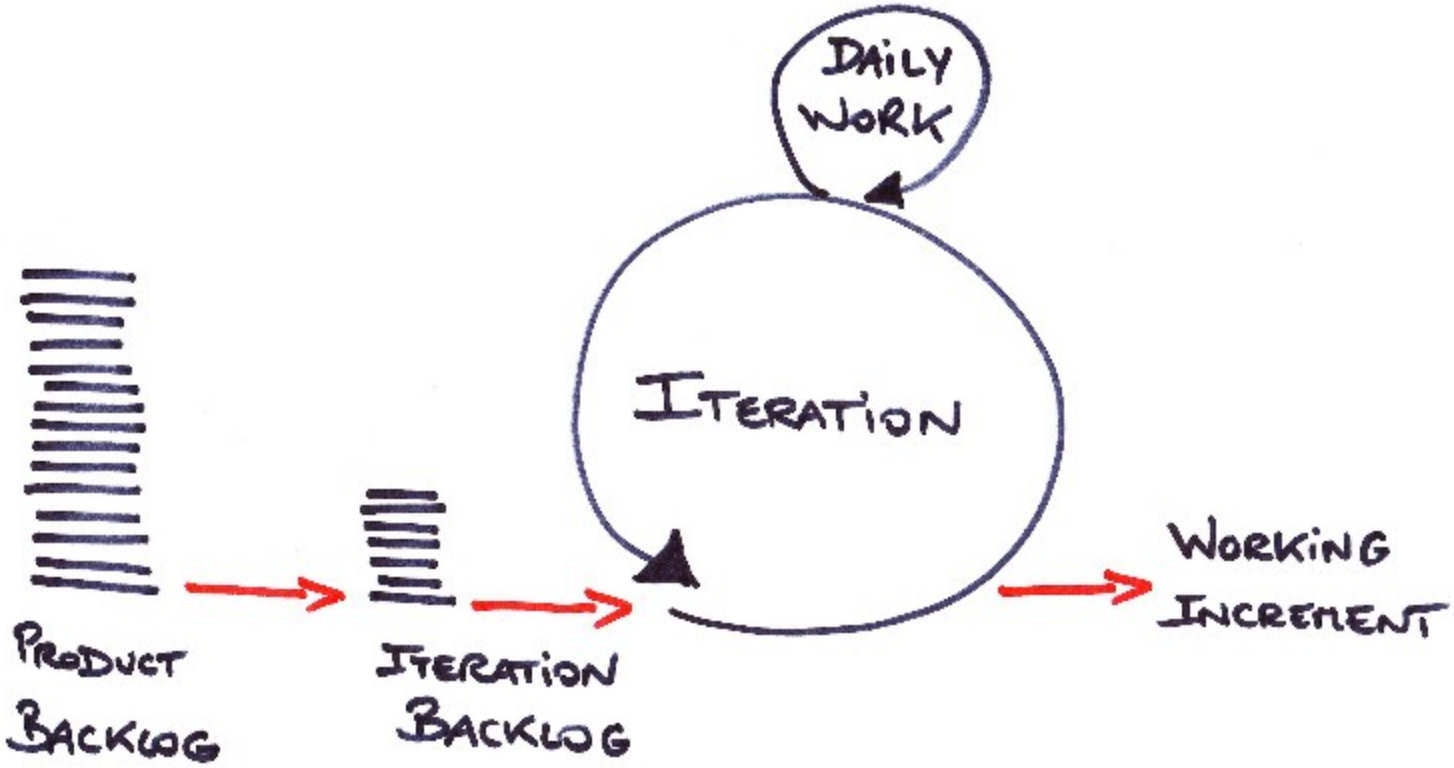
Source: http://www.ambysoft.com/surveys/modelingDocumentation2008.html

Agenda

- Introduction
- Mythes: architecture et agilité
- > **Rôles et cycles agiles**
- Qu'est ce que l'architecture?
- L'architecte « agile »
- Conclusion

Cycle de développement agile

- De nombreuses méthodes Agiles
  - XP, Crystal, DSDM, Scrum, OpenUP...
  - Toutes préconisent l'adoption d'un cycle itératif et incrémental



agile tour 2011

Les rôles définis par les méthodologies agiles

Quelques exemples des rôles couramment définis dans les méthodologies agiles:

- Le représentant du client
  - Établit les priorités pour l'équipe et est le contact avec le client (Product owner, XP customer)
- Un facilitateur
  - S'assure que l'équipe n'a pas d'obstacles pour atteindre ses objectifs (XP coach, ScrumMaster, Team lead)
- Une équipe de développement
  - Equipe auto-gérée qui construit le système (XP programmer, developer)

Pas d'architecte ou de concepteur?

Agenda

- Introduction
- Mythes: architecture et agilité
- Rôles et cycles agiles

> Qu'est ce que l'architecture?

- L'architecte « agile »
- Conclusion

Qu'est ce que l'architecture ?

Analyse architectural, une **activité itérative** pour:

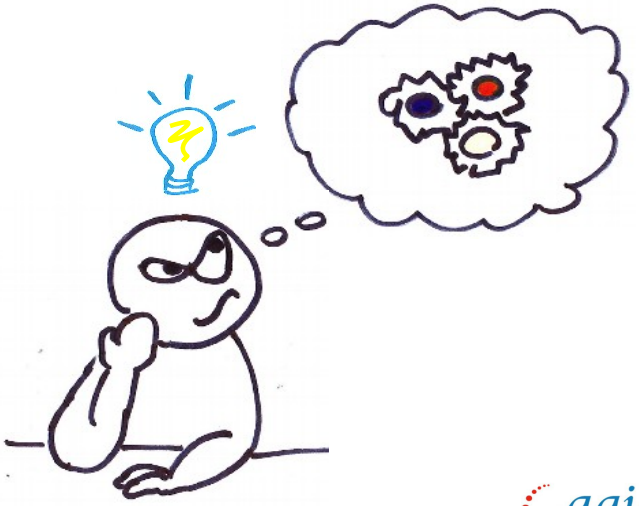
- Décrire la structure et les comportements d'un système
- Identifier et adresser les risques d'un système
- Fournir un contexte et guider le développement d'un système
- Moderniser un système
- Fournir un aperçu d'un système
  - Client
  - Équipe de développement ou de maintenance

La frontière entre l'architecture et le design est difficile à établir.

Qu'est ce qu'un architecte ?

- Un rôle, pas une personne
  - Peut être joué par plusieurs personnes
  - Celui qui joue le rôle d'architecte peut aussi jouer d'autres rôles

- Définition du rôle: Coordonner la solution technique d'un système
  - Vision globale
  - Choix techniques
  - Réduction des risques



Ce qu'un architecte n'est pas...

Mauvaises pratiques souvent observées

- Tour d'ivoire
  - Dicter plutôt que guider
- Un spécialiste de documentation
  - Big design up front (BDUF)
- Titre honorifique
  - Outil de promotion, à l'ancienneté
- Sans expérience de développement
  - « Non-coding architect »

Agenda

- Introduction
- Mythes: architecture et agilité
- Rôles et cycles agiles
- Qu'est ce que l'architecture?

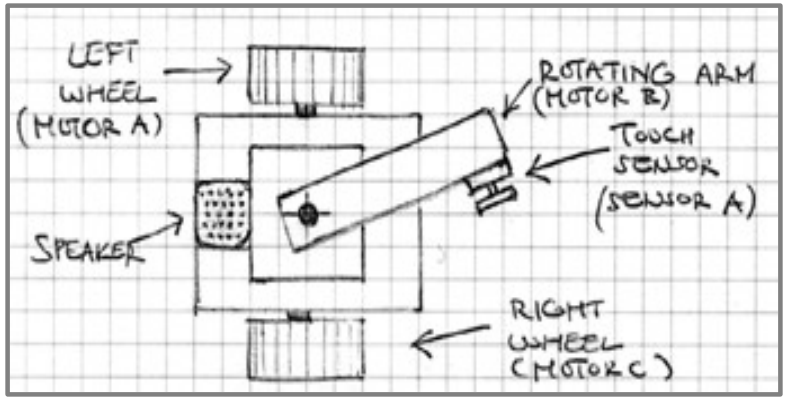
> **L'architecte « agile »**

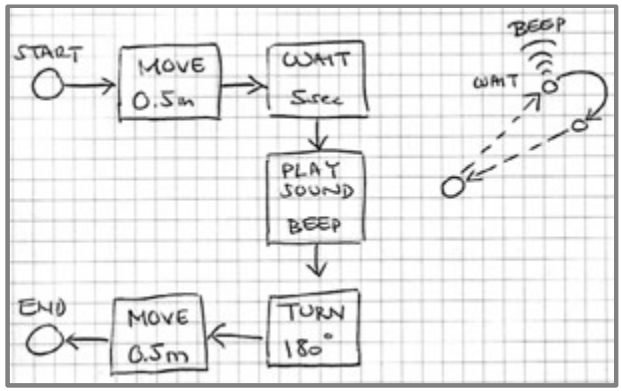
- Conclusion

Projets agiles: l'architecture est incontournable

- Chaque système a une architecture
  - Les éléments et mécanismes qui définissent la structure et les comportements d'un système

- Architecture ne veut pas dire documentation
  - Un gros document d'architecture n'est jamais lu
  - Parfois, une image vaut mille mots

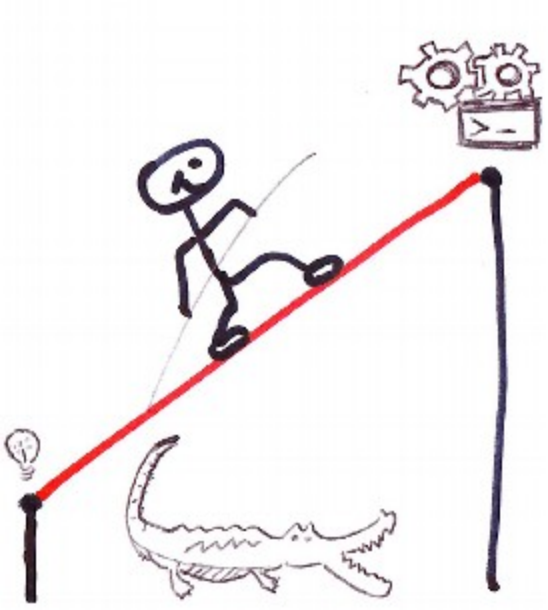




agile tour 2011

Les bonne pratiques de l'architecte agile

- Analyser les besoins
  - Fonctionnels et non-fonctionnels
  - Ne pas inventer des besoins
- Réutiliser les solutions qui ont fait leurs preuves
  - Meilleures pratiques
  - Expériences passées
- S'attaquer aux risques techniques
  - Très tôt dans le projet
  - En continu, au cours de chaque itération
- Valider l'architecture par du code
  - « Architecture exécutable »
- Travailler de façon itérative et incrémentale

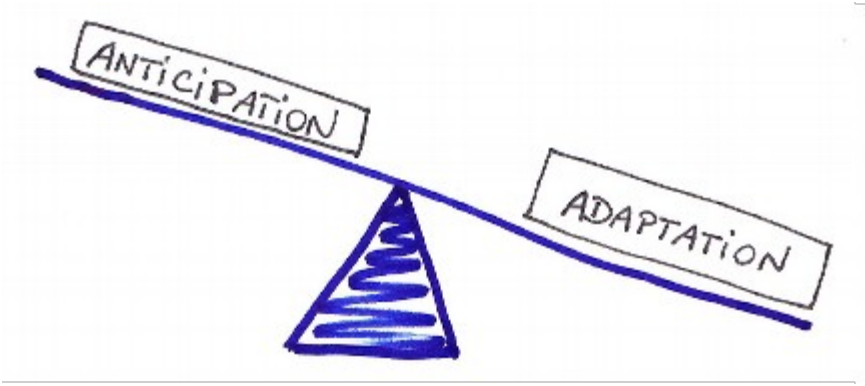


Projets agiles: architecture intentionnelle et émergente

- **Architecture intentionnelle**
  - Décisions délibérées / quelques besoins anticipés
  - L'anticipation est par définition imparfaite

- **Architecture émergente**
  - L'architecture s'adapte aux besoins
  - Les corrections sont inévitables, à chaque itération
  - L'architecture émerge du développement, en fonction des défis techniques rencontrés

- L'architecture « agile » consiste à trouver le juste équilibre entre **anticipation** et **adaptation**



Source: Succeeding with Agile, M. Cohn, 2007

Agilité à plusieurs échelles

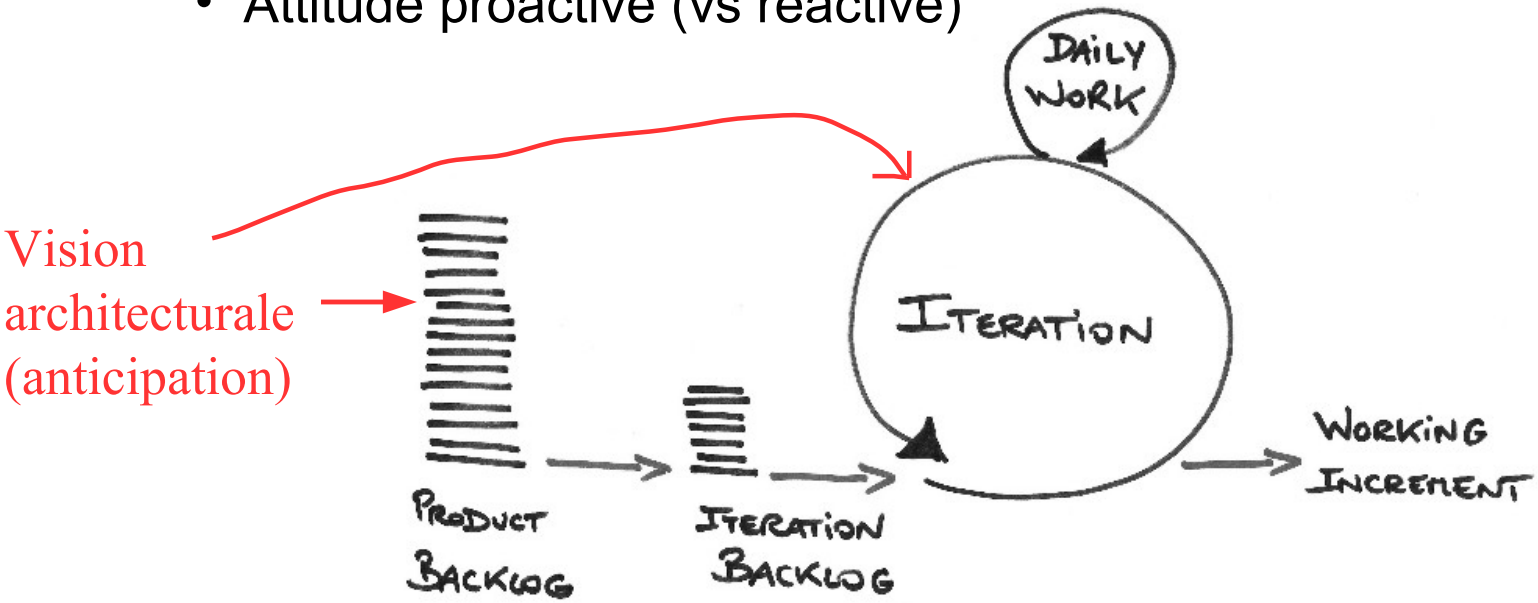
- Tous les projets ne sont pas les mêmes
  - Logiciel bureautique, application web, jeu vidéo, système transactionnel, système critique haute disponibilité...
  - L'agilité doit s'adapter aux besoins des projets

- La complexité des projets influence les pratiques agiles
  - Complexité technique
  - Équipes distribuées
  - Complexité organisationnelle
  - Taille des équipes
  - Standards et régulations

L'architecte, un membre de l'équipe de développement (1)

La vision architecturale (quelques heures ou jours de travail)

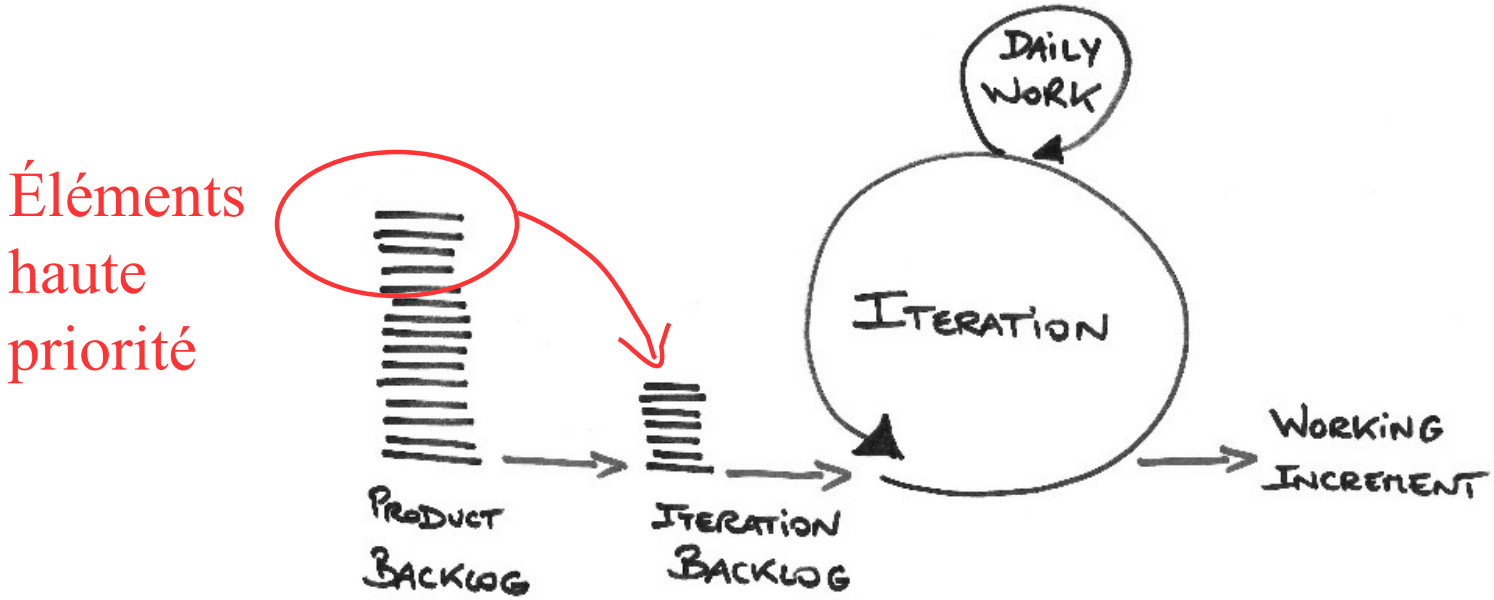
- L'architecte travaille avec l'équipe
  - Idée globale du système à développer
  - Influencer le contenu de la « product backlog »
    - Attitude proactive (vs réactive)



L'architecte, un membre de l'équipe de développement (2)

Les priorités du « product backlog »

- L'architecte travaille avec le représentant du client (product owner)
  - Eduquer sur les risques techniques
  - Influencer les priorités

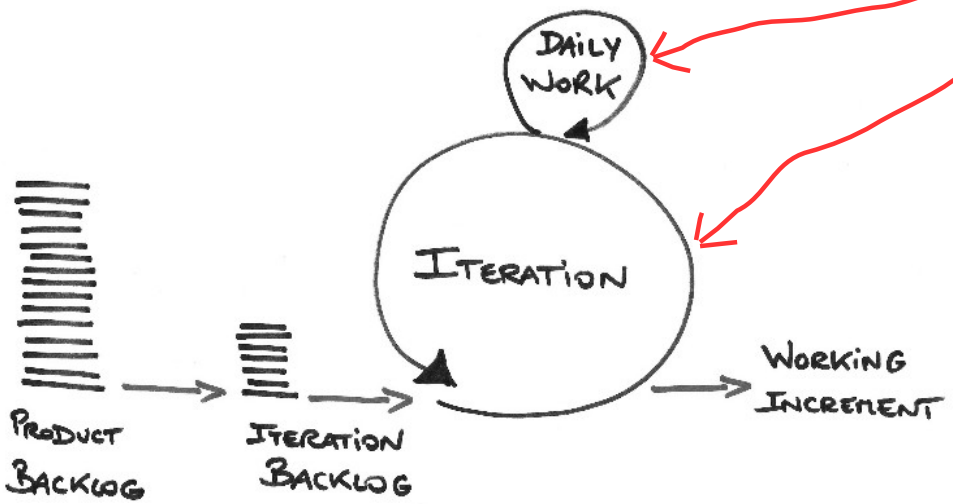


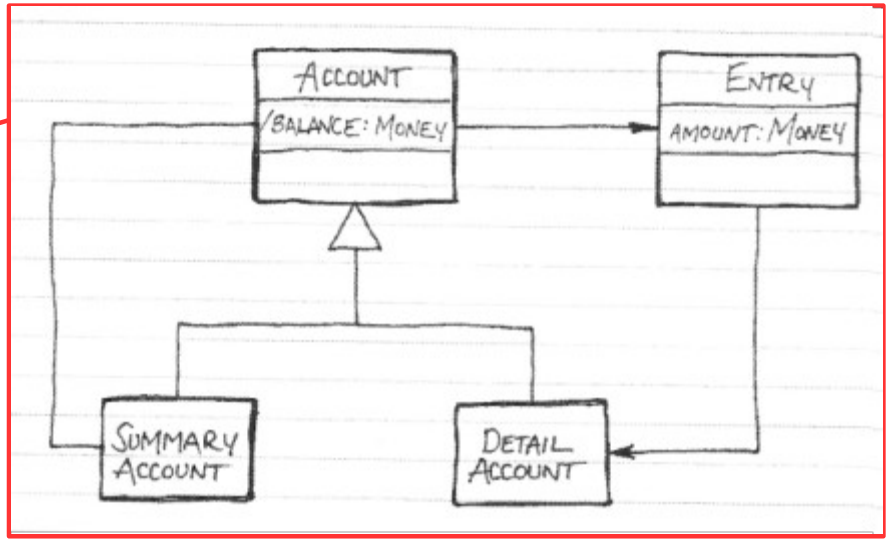
agile tour 2011

L'architecte, un membre de l'équipe de développement (3)

Les priorités du « Iteration backlog » et les session de design « just-in time »

- L'architecte travaille avec l'équipe
  - Priorités pour l'itération
  - Estimation du travail de l'itération
  - Modélisation pour l'itération
  - Refactoring





Source: Extreme Programming Installed, R. Jeffries, 2007

Agenda

- Introduction
- Mythes: architecture et agilité
- Rôles et cycles agiles
- Qu'est ce que l'architecture?
- L'architecte « agile »

> **Conclusion**

Conclusion

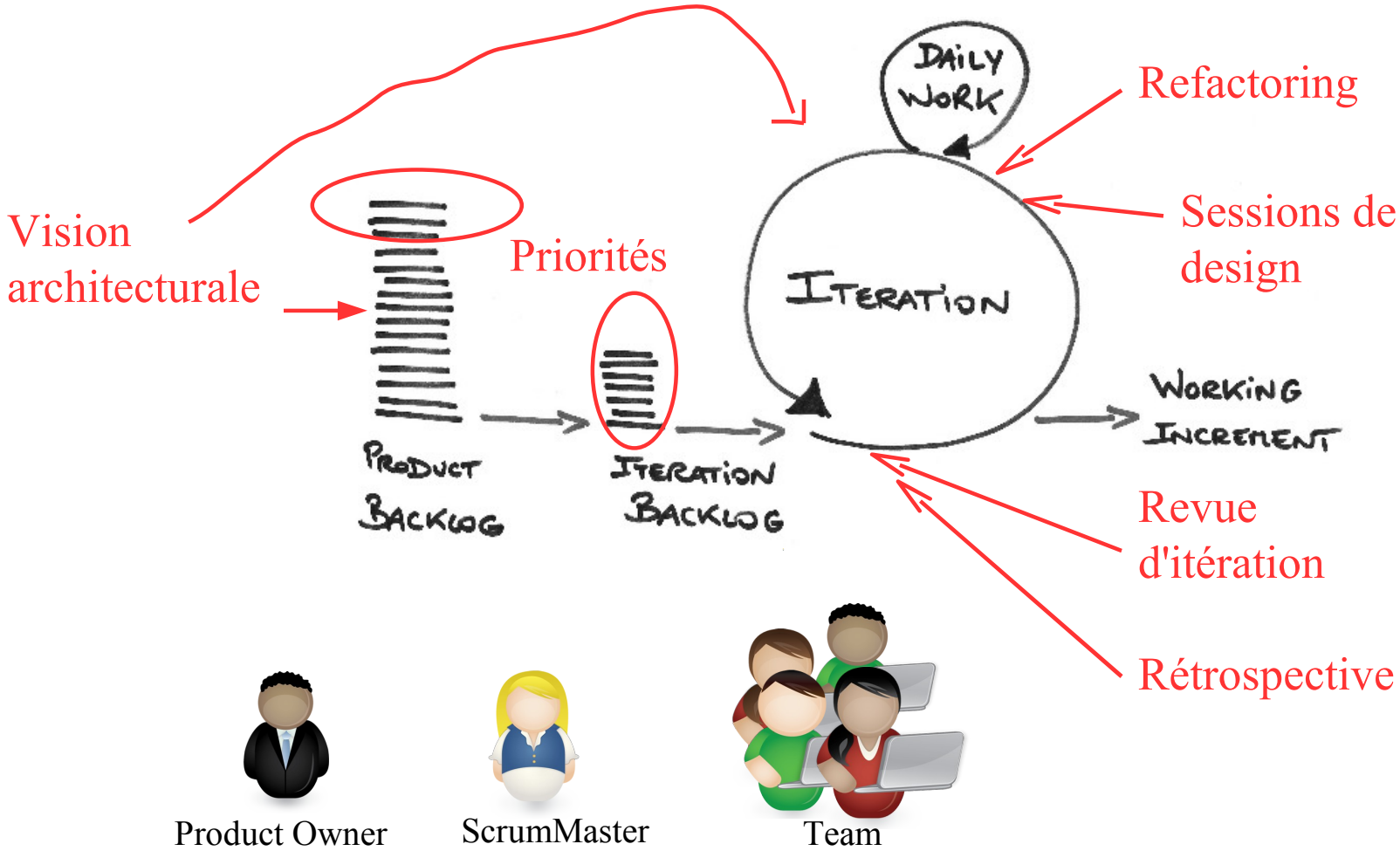
- Les méthodologies agiles encouragent les activités d'architecture et de conception
  - L'architecture doit être itérative et incrémentale
  - Pas de Big Design Up Front (BDUF)
  - Little Design Up Front (LDUF) peut aider (vision architecturale)

- L'architecture « agile » est intentionnelle et émergente
  - Anticipation et adaptation

- L'architecte agile intervient tout au long du cycle de développement
  - Un rôle de l'équipe de développement

Les interventions d'un architecte: Récapitulatif

Besoins en architecture tout au long du cycle de développement agile.



Plus d'information sur des techniques d'architecture

IBM. English

developerWorks. Technical topics Evaluation software Community Events

- Rational
- New to Rational
- Products
- Downloads
- Technical library
- Community & forums
- Events

Related links
- How to buy
- Training
- News
- Support

developerWorks > Technical topics >

Rational software

Technical resources and best practices for the Rational software delivery pl

A practical guide to developing enterprise architecture

With humor, Franki Schafrik guides you through identifying the questions that enterprise architecture must answer, developing an approach and models, and analyses to see where you can make improvements. She assumes that you use modeling software for your architecture. More >

Featured content | Rational Jazz resources | Complex & Embedded Systems

18 Oct 2011 — Show descriptions | **Hide descriptions**

- Develop TXSeries CICS apps with Rational Developer for Power Systems Software, Part 2
- Apply agile project management methods for more efficient hardware development
- Business Agility: A technical perspective from the IBM Rational software staff
- Define application architecture with Rational Software Architect, Part 1
- Define application architecture with Rational Software Architect, Part 2

→ More content RSS

- Part 1: Envision the architecture
- Part 2: Iteratively refine the architecture

Jazz Community Site (http://jazz.net)

Home | About Jazz | Roles | Projects | Downloads | Forums | Library | Development | Community | Jean-Louis Marechaux

jazz® COMMUNITY SITE

We're building a new generation of products to make software and systems development more collaborative, productive, and enjoyable.

Type to search

CLM 2011
Download 3.0.1 releases of RTC, RQM, and RRC

Rational DOORS Next Generation
Check out our newest requirements initiative

Sandbox Trials
Get an instant trial of CLM 2011 including RTC, RQM, and RRC

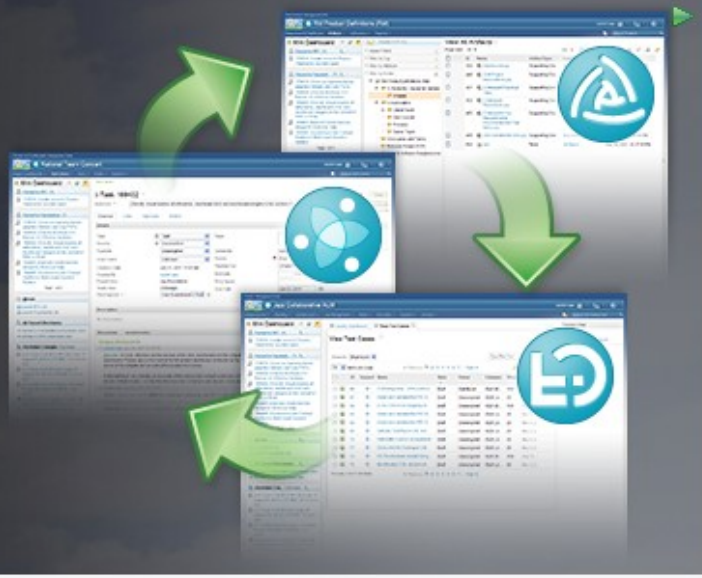
JazzHub
Free, public hosting, powered by Rational Team Concert

2

CLM 2011

The Rational solution for Collaborative Lifecycle Management is a ready-to-run integrated ALM solution comprising Rational Team Concert, Rational Quality Manager, and Rational Requirements Composer.

Learn more in the blog >



Rational Team Concert | Rational Quality Manager | Rational Requirements Composer | Rational DOORS Next Generation | Rational Insight | Design Management

5

Références

Web

- A Practical Guide to Distributed Scrum: http://www.distributedscrum.com/
- OpenUP: http://epf.eclipse.org/wikis/openup/
- Agile Modeling: http://www.agilemodeling.com/
- Surveys: http://www.ambysoft.com/surveys/
- Jazz Community Site: http://jazz.net/
- The Emphasys group training: Essentials of Agile Modeling

Blog

- http://blog.objectmentor.com/
- https://www.ibm.com/developerworks/mydeveloperworks/blogs/ambler/?lang=en
- http://jamesshore.com/Blog/

Books and Articles

- Define application architectures with Rational Software Architect (October 2011)
  - Part 1 - Envision the architecture
  - Part 2 - Iteratively refine the architecture
- A Practical Guide to Distributed Scrum, E. Woodward, S. Surdek, M. Ganis, IBM Press 2010
- Extreme Programming Explained, K. Beck, Addison Wesley 2005
- Succeeding with Agile, M. Cohn, Addison Wesley 2009
- Extreme Programming Installed, R. Jeffries, Addison Wesley 2007
- The art of Agile development, O'Reilly 2007
- Implementing Lean Software Development, M. Poppendieck, Addison Wesley 2006
- Agile Software Development, the cooperative game, A. Cockburn, Addison Wesley 2007

Références - Extraits (1)

- **Agile Manifesto - 2001**

Principle #9: « Continuous attention to technical excellence and good design enhances agility. »

Principle #11: « The best architectures, requirements, and designs emerge from self-organizing teams. »

- **Kenn Beck** (Extreme Programming Explained, Addison Wesley 2005, p74 / p52)

« Architects on a XP team look for and execute large-scale refactoring, write system level tests that stress the architecture, and implement stories. »

« The question is not whether or not to design, it is when to design. »

- **Mike Cohn** (Succeeding with Agile, Addison Wesley 2009, p166)

« Scrum team acknowledge that as nice as it might be to make all the design decisions up-front, doing so is impossible. [...] design is both intentional and emergent. »

- **Scott Ambler** (Architectural Envisioning: An Agile Best Practice, Ambysoft 2004-2011)

« Some people will tell you that you don't need to do any initial architecture modeling at all. However, my experience is that doing some initial architectural modeling in an agile manner offers several benefits. »

- **Robert Martin, Uncle Bob** (The scatology of Agile architecture, Object Mentor blog 2009)

« One of the more insidious and persistent myths of agile development is that up-front architecture and design are bad; that you should never spend time up front making architectural decisions. [...] This myth is not part of agile at all. Rather it is an hyper-zealous response to the real Agile proscription of Big Design Up Front (BDUF). »

Références – Extraits (2)

- **James Shore** (The art of Agile development, O'Reilly 2007, chapter 3)

« Expert designers and architects are still necessary. They contribute by guiding the team's incremental design and architecture efforts and helping team members see ways of simplifying complex designs. »

- **Ron Jeffries** (Extreme Programming Installed, Addison Wesley 2007, p70)

« If you aren't clear on just what to do, or if you want a little reassurance, or if you would like a few additional ideas, have a quick design session. »

- **Mary Poppendieck** (Implementing Lean Software Development, Addison Wesley 2006, p182)

« The objective of a good software architecture is to keep irreversible decisions to a minimum and provide a framework that supports iterative development. »

- **Alistair Cockburn** (Agile Software Development, the cooperative game, Addison Wesley 2007, p252)

« Create a simple architecture that handles all the known big rocks [...]. Handle the small rocks as they appear during the project. »

Merci

Jean-Louis Maréchaux

jl.marechaux@ca.ibm.com

agile tour 2011