

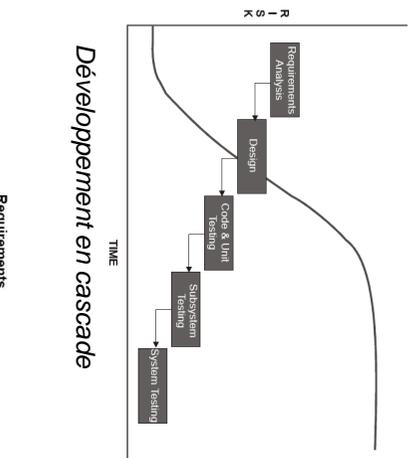
# IBM Rational software

## L'agilité avec Scrum

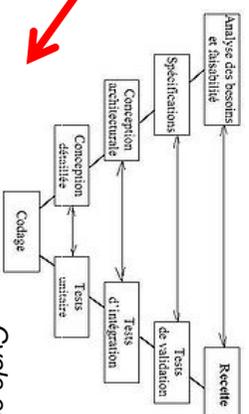
Jean-Michel Athané,  
IBM Certified IT Specialist



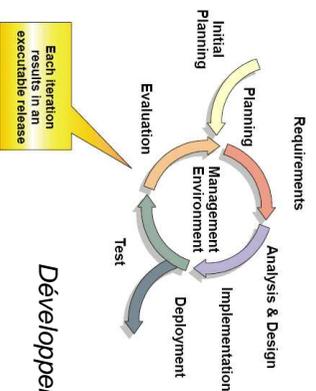
## Gestion de Projet Logiciel la recherche de méthodes de travail adaptées



Développement en cascade



Cycle en V



Développement itératif

La bonne conduite des Projets de Développement Logiciel a fait l'objet de nombreuses approches successives, pour mieux **s'adapter aux changements en cours de projet** et lever les risques au plus tôt



## Mener à bien un projet logiciel reste un vrai défi.. .. et les échecs restent nombreux

### Business realities

#### Software failure impacts the bottom line

- **Airbus**  
Non-compatible software used at different Airbus factories  
**\$6.1 billion** profit hit over the next four years, 2006
- **TJX**  
Cost of data breach at TJX soars  
**\$256 million** cost, 2007
- **J Sainsbury PLC (UK)**  
Supply-chain management system abandoned after deployment  
**\$527 million** cost, 2004
- **Hewlett-Packard**  
Problems with ERP system  
**\$160 million** loss, 2004

#### Only 34% of software projects succeed

*Only 37% of stakeholders are satisfied with the speed of internal application development, and just 42% are satisfied with the quality*

*Software related downtime costs industry almost \$300 billion annually*

Sources: Forrester Research, Standish CHAOS Report, Comparative Economic Normalization Technology Study, Business Week, CEO Magazine, Computerworld, InfoWeek, Fortune, The New York Times, Time, The Wall Street Journal, Boston Globe.



3

3

## La « révolution » Agile une nouvelle approche pragmatique avant tout

- Dans les années 90, apparaissent différentes méthodes dites « agiles » dont les plus célèbres sont « Extreme Programming » et « Scrum »,
  - ▶ Rapid Application Development (RAD, 1991)
  - ▶ Dynamic systems development method (DSDM, 1995, consortium anglais commercialisant le RAD)
  - ▶ Scrum (1996)
  - ▶ Feature Driven Development (FDD) (1999)
  - ▶ Extreme programming (XP, 1999)
  - ▶ Adaptive software development (ASD, 2000)
  - ▶ Crystal clear (2004)
- En 2001, les différents tenants de ces méthodes se réunissent pour définir le « Manifeste Agile » autour de ces 4 valeurs :
  - ▶ L'équipe (« Personnes et interaction plutôt que processus et outils »).
  - ▶ L'application (« Logiciel fonctionnel plutôt que documentation complète »)
  - ▶ La collaboration (« Collaboration avec le client plutôt que négociation de contrat »)
  - ▶ L'acceptation du changement (« Réagir au changement plutôt que suivre un plan »)

Source : [www.wikipedia.fr](http://www.wikipedia.fr)



4

4

## Les 12 principes du développement agile (1/2)

- « Notre première priorité est de satisfaire le client en livrant tôt et régulièrement des logiciels utiles ».
- « Le changement est bienvenu, même tardivement dans le développement. Les processus agiles exploitent le changement comme avantage compétitif pour le client ».
- « Livrer fréquemment une application fonctionnelle, toutes les deux semaines à deux mois, avec une tendance pour la période la plus courte ».
- « Les gens de l'art et les développeurs doivent collaborer quotidiennement au projet ».
- « Bâissez le projet autour de personnes motivées. Donnez leur l'environnement et le soutien dont elles ont besoin, et croyez en leur capacité à faire le travail ».
- « La méthode la plus efficace pour transmettre l'information est une conversation en face à face ».



Source : [www.wikipedia.fr](http://www.wikipedia.fr)

5

5

## Les 12 principes du développement agile (2/2)

- « Un logiciel fonctionnel est la meilleure unité de mesure de la progression du projet ».
- « Les processus agiles promeuvent un rythme de développement soutenable. Commanditaires, développeurs et utilisateurs devraient pouvoir maintenir le rythme indéfiniment ».
- « Une attention continue à l'excellence technique et à la qualité de la conception améliore l'agilité ».
- « La simplicité - l'art de maximiser la quantité de travail à ne pas faire - est essentielle ».
- « Les meilleures architectures, spécifications et conceptions sont issues d'équipes qui s'auto-organisent ».
- « À intervalle régulier, l'équipe réfléchit aux moyens de devenir plus efficace, puis accorde et ajuste son comportement dans ce sens ».



Source : [www.wikipedia.fr](http://www.wikipedia.fr)

6

6

## En résumé : qu'est-ce que l'Agilité ?

- Une approche **itérative et incrémentale** menée de façon **fortement collaborative** avec le **juste niveau de formalisme** pour produire un **logiciel de haute qualité** dans le **respect des coûts et des délais** et en **satisfaisant les besoins évolutifs** de ses clients.

### Principes :

- ▶ *Collaboration constante au sein de l'équipe en impliquant les parties prenantes*
- ▶ *Réponse rapide au changement à l'aide de plans d'itérations*
- ▶ *Livraison fréquente de logiciel en fonctionnement*
- ▶ *Tests et validations en continu*
- ▶ *Transparence/visibilité*
- ▶ *Processus adaptés "juste ce qu'il faut"*

#### Taux d'adoption :

- 69% des organisations mènent un ou plusieurs projets agiles
- 82% sont au delà d'un projet pilote
- 25% restants espèrent mener un projet Agile cette année (2008)

Source : Dr. Dobb's Journal ([www.ddj.com](http://www.ddj.com))  
2008 Agile Adoption Survey



7

7

## Qu'est-ce que Scrum ?



- Scrum a été créée par Jeff Sutherland à Easel en 1993
- "Scrum" signifie "mêlée", terme venant du rugby où les équipes doivent travailler étroitement ensemble pour prétendre à la victoire
- Scrum définit un cadre pour la planification et l'exécution de projet
- Scrum aide les équipes à **délivrer le plus rapidement possible de la valeur** en proposant une organisation de travail appropriée
- Scrum vise à
  - ▶ Améliorer la communication et maximiser la coopération
  - ▶ Détecter et éliminer tous les obstacles à la production de logiciel opérationnel
  - ▶ Maximiser la productivité
- Scrum est en générallement complété par d'autres pratiques agiles comme l'intégration continue

*Scrum se veut déployable sur des petits projets jusqu'à des grosses organisations*

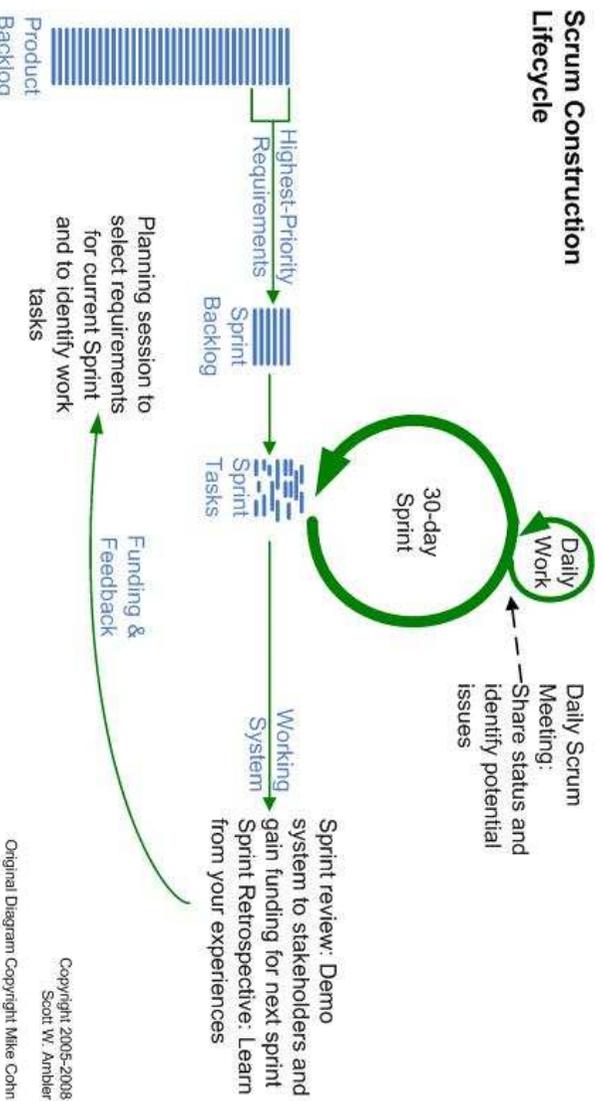


8

8

# Le Cycle de Vie Scrum

## Scrum Construction Lifecycle



Copyright 2005-2008  
Scott W. Ambler  
Original Diagram Copyright Mike Cohn

Source: [www.amblysoft.com/essays/agileLifecycle.html](http://www.amblysoft.com/essays/agileLifecycle.html)

# Les Eléments Fondamentaux de Scrum

- Le processus SCRUM se veut simple, facile à adopter, performant, réactif, et repose sur une vraie dynamique d'équipe.
- Il comporte donc peu de rôles, peu d'artéfacts, et un certain nombre de meetings de coordination destinés à en contrôler rigoureusement le déroulement.

### Roles

- Product Owner
- Scrum Master
- Team

### Artifacts

- Product Backlog
- Sprint Backlog
- Burndown Charts

### Meetings

- Sprint Planning
- Daily Meeting
- Sprint Review
- Sprint Retrospective

## Les Éléments Fondamentaux de Scrum

**Roles**

- Product Owner
- Scrum Master
- Team

- **PRODUCT OWNER**
  - ▶ Représente les clients et les utilisateurs
  - ▶ Définit la liste des fonctionnalités qui doivent être dans le produit final
  - ▶ Etablit un ordre de priorité
  - ▶ Approuve le contenu des releases
  
- **TEAM**
  - ▶ L'équipe de ceux qui développent l'application
  - ▶ Attribue des points de complexité aux fonctionnalités demandées
  - ▶ Décline ces fonctionnalités en tâches de développement qui sont évaluées en durée de réalisation
  
- **SCRUM MASTER**
  - ▶ Anime le processus
  - ▶ Protège l'équipe des perturbations extérieures
  - ▶ Lève les obstacles rencontrés

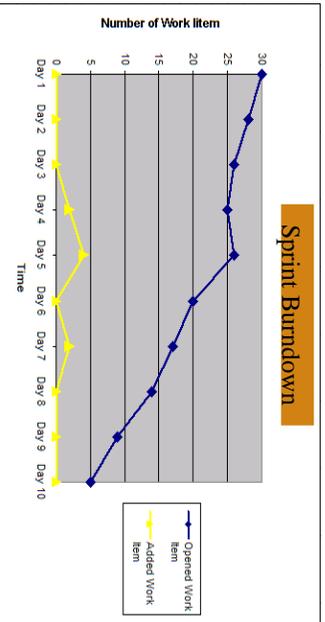


## Les Éléments Fondamentaux de Scrum

**Artifacts**

- Product Backlog
- Sprint Backlog
- Burndown Charts

- **PRODUCT BACKLOG**
  - ▶ Liste des fonctionnalités qui sont à implémenter dans le produit
  - ▶ A chaque fonctionnalité est associée un certain nombre de points de complexité, à la suite d'une discussion avec les membres de l'équipe
  
- **BURNDOWN CHART**
  - ▶ Courbe d'évolution du « reste à faire » en nombre d'heures de travail pour le sprint courant.
  
- **SPRINT BACKLOG**
  - ▶ Liste des fonctionnalités qui sont à implémenter dans l'itération (SPRINT) courante, déclinées en tâches ayant chacune une durée de réalisation.



## Les Éléments Fondamentaux de Scrum

- **SPRINT PLANNING**
  - ▶ Etablit la liste des tâches à réaliser dans le **SPRINT** qui commence

- **DAIYLY MEETING (15 min)**

- ▶ Chaque développeur de la **TEAM** dit
  - ▶ Ce qu'il a fait depuis le dernier meeting
  - ▶ Ce qu'il compte faire jusqu'au prochain meeting
  - ▶ Les obstacles (**IMPEDIMENTS**) qu'il a détecté

- Meetings

  - Sprint Planning
  - Daily Meeting
  - Sprint Review
  - Sprint Retrospective

- **SPRINT REVIEW**

- ▶ L'équipe fait une démonstration du code développé à toutes les parties prenantes
- ▶ Le Product Owner évalue les atteintes du Sprint

- **SPRINT RETROSPECTIVE**

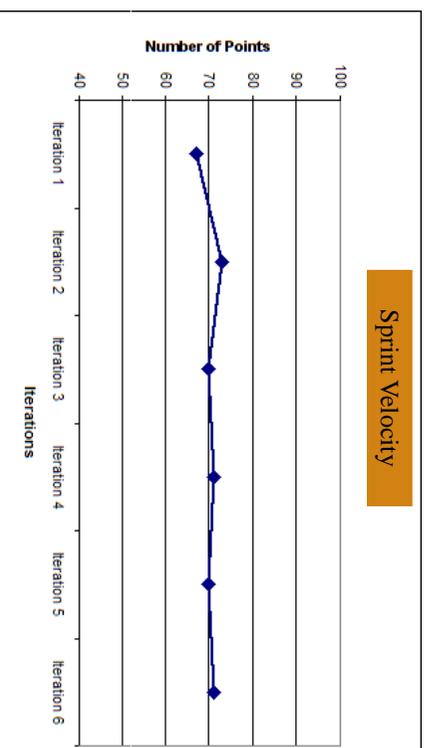
- ▶ Le SCRUM master évalue avec l'équipe
  - ▶ Que doit-on continuer à (bien) faire ?
  - ▶ Que doit-on arrêter de (mal) faire ?



14

## Vélocité

- Pour chaque élément du backlog produit, on attribue une valeur en points qui est une évaluation du travail à fournir. ("*planning poker*")
- On peut ainsi établir la vélocité de l'équipe, c'est à dire sa capacité moyenne de réalisation – en points - pour chaque sprint.



- L'intérêt de cette démarche est d'avoir une idée du travail requis pour réaliser chaque fonctionnalité sans pour autant lui donner une valeur en jours que le product owner serait tenté de considérer comme définitivement acquise.
- En revanche, on utilise la **vélocité** pour planifier le projet à l'échelle macroscopique de façon fiable et précise.



15

## La tour de Babel des terminologies “agiles”

Disciplined Agile	XP	Scrum	Other
Team Lead	Coach	Scrum Master	Project Manager
Iteration	Iteration	Sprint	Timebox
Daily Stand Up Meeting	Daily Stand Up	Scrum Meeting	Coordination Meeting
Retrospective	Retrospective	Sprint Retrospective	Reflection Meeting
Product Owner	Customer	Product Owner	Stakeholder Representative
Team Member	Extreme Programmer	Team Member	Developer
Iteration Review	-	Sprint Review	-

Chaque méthode “agile” a sa propre terminologie.  
Il y a peu de chances que l'on aille vers une standardisation



16

15

## Les obstacles fréquents du passage à SCRUM

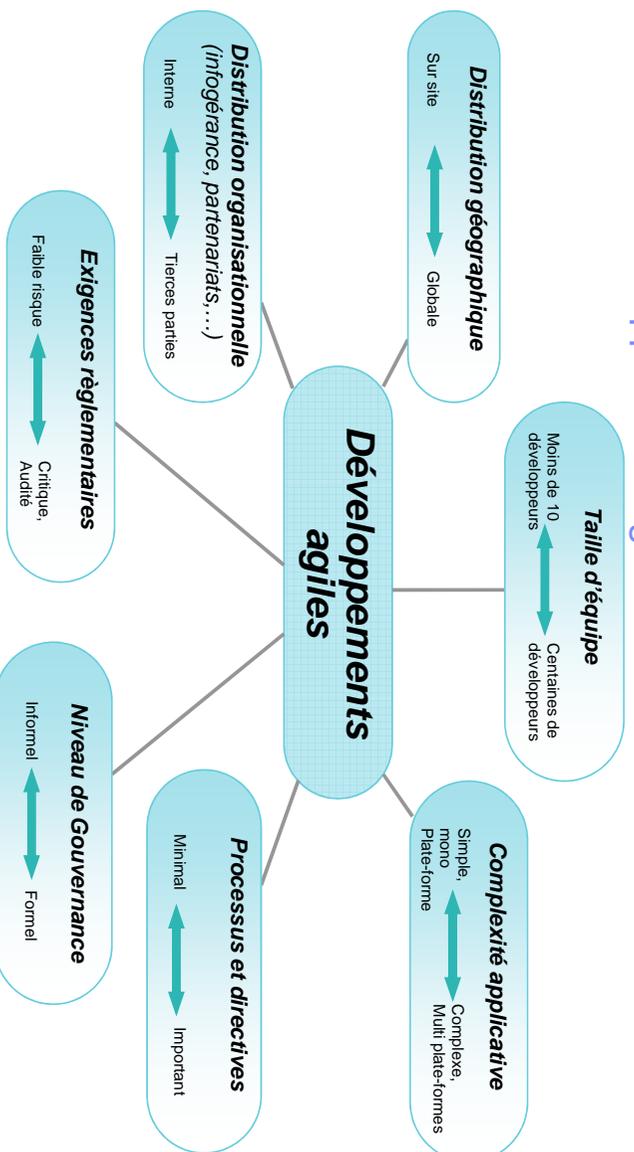
- Passer d'un processus en cascade à la pratique de SCRUM
  - ▶ Beaucoup d'équipes trouvent difficile au départ d'avoir des réunions d'avancement quotidiennes
  - ▶ Estimation des points pour les stories
- Trouver le bon Product Owner
  - ▶ Ce rôle revient souvent aux Business Analysts, mais ceux-ci n'ont pas l'habitude d'être associés directement au résultat du projet.
- Le ScrumMaster n'est pas un Team Leader
  - ▶ L'équipe se manage elle-même. Ce concept n'est pas facile à adopter
  - ▶ Trouver un Scrum Master qui comprend bien l'essence de SCRUM et de son rôle (facilitateur, gardien du processus)
- Les plans Scrum sont empiriques
  - ▶ Le planning du projet s'adapte en permanence
  - ▶ A chaque sprint, les équipes doivent s'adapter aux changements dans les exigences
  - ▶ Estimer correctement le travail (story points, task hours...)



17

16

## Le challenge : Porter les développements agiles à une échelle industrielle



### Learn more at:

- [IBM Rational software](#)
- [Integrated Requirements Management](#)
- [Process and portfolio management](#)
- [Change and release management](#)
- [Quality management](#)
- [Architecture management](#)
- [Rational trial downloads](#)
- [Leading Innovation Web site](#)
- [developerWorks Rational](#)
- [IBM Rational TV](#)
- [IBM Business Partners](#)
- [IBM Rational Case Studies](#)

© Copyright IBM Corporation 2008. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors and are not intended to be a commitment to future product offerings. The information contained in these materials may be subject to change without notice. Other company, product, or service names may be trademarks or service marks of others. Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.

