OpenUP

Duvivier Julien - Orban Pierre-Yves

30/04/2013

Plan

1. OpenUP, c'est quoi ?
2. Principes
3. Organisation
   a. **Méthode**

      (1) Rôles, (2) Disciplines, (3) Tâches , (4) Artifacts, (5) Directives
   b. **Processus**

      (1) Patterns, (2) Cycle de vie itération, (3) Delivery process
4. Micro incrément
5. Influences

OpenUP, c'est quoi ?

- 2005: Basic Unified Process par IBM
- 2006: OpenUP et EPF par Eclipse Foundation

- Process itératif basé sur RUP
  - Minimal
  - Complet
  - Extensible

- Approche agile
  - Collaboration et communication

Principes

1. Collaborer pour aligner ses intérêts et partager les connaissances

2. Equilibrer les priorités afin de maximiser le bénéfices des parties prenantes

3. Se concentrer sur l'architecture très tôt pour minimiser les risques et organiser le développement

4. Evoluer pour continuellement recevoir des feedbacks et s'améliorer

Principes // Manifeste agile

1. Collaborer pour aligner ses intérêts et partager les connaissances
   --> *"Individuals and interactions over process and tools"*

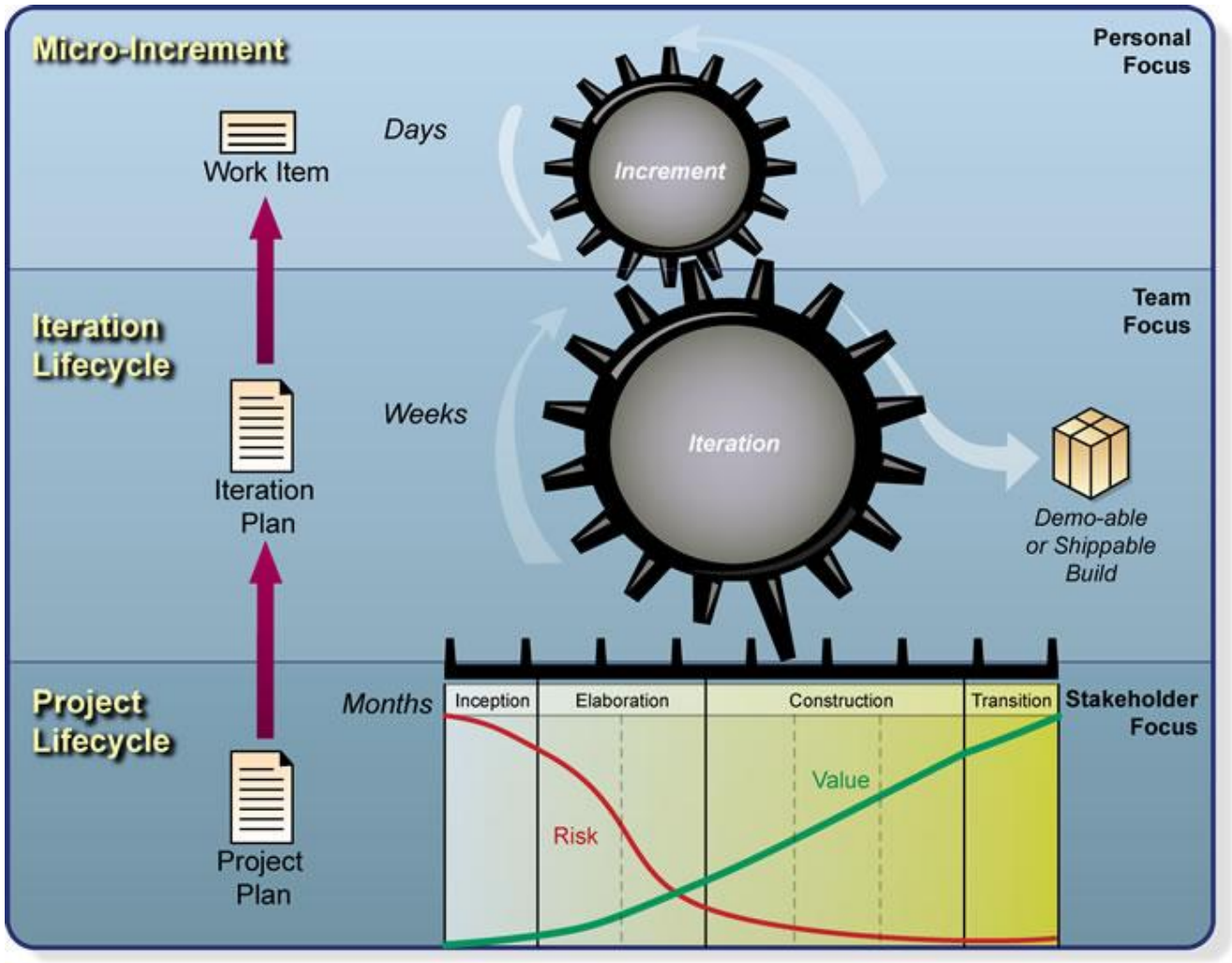
2. Equilibrer les priorités afin de maximiser le bénéfices des stakeholders
   --> *"Customer collaboration over contract negotiation"*

3. Se concentrer sur l'architecture très tôt pour minimiser les risques et organiser le développement
   --> *"Working software over comprehensive documentation"*

4. Evoluer pour continuellement recevoir des feedbacks et s'améliorer
   --> *"Responding to change over following a plan"*

Organisation

Niveaux :



Méthode: les rôles (1)

- Rôles
  - Parties prenantes
  - Analyste
  - Architecte
  - Développeur
  - Testeur
  - Project Manager
  - Any role

- + 6 rôles spécifiques du déploiement
- + 2 rôles spécifiques de l'environnement

Méthode: les disciplines (2)

La méthode se concentre sur les disciplines suivantes :

1. Exigences
2. Architecture
3. Développement
4. Test
5. Project Management
6. Configuration & change management

Méthode: les tâches (3)

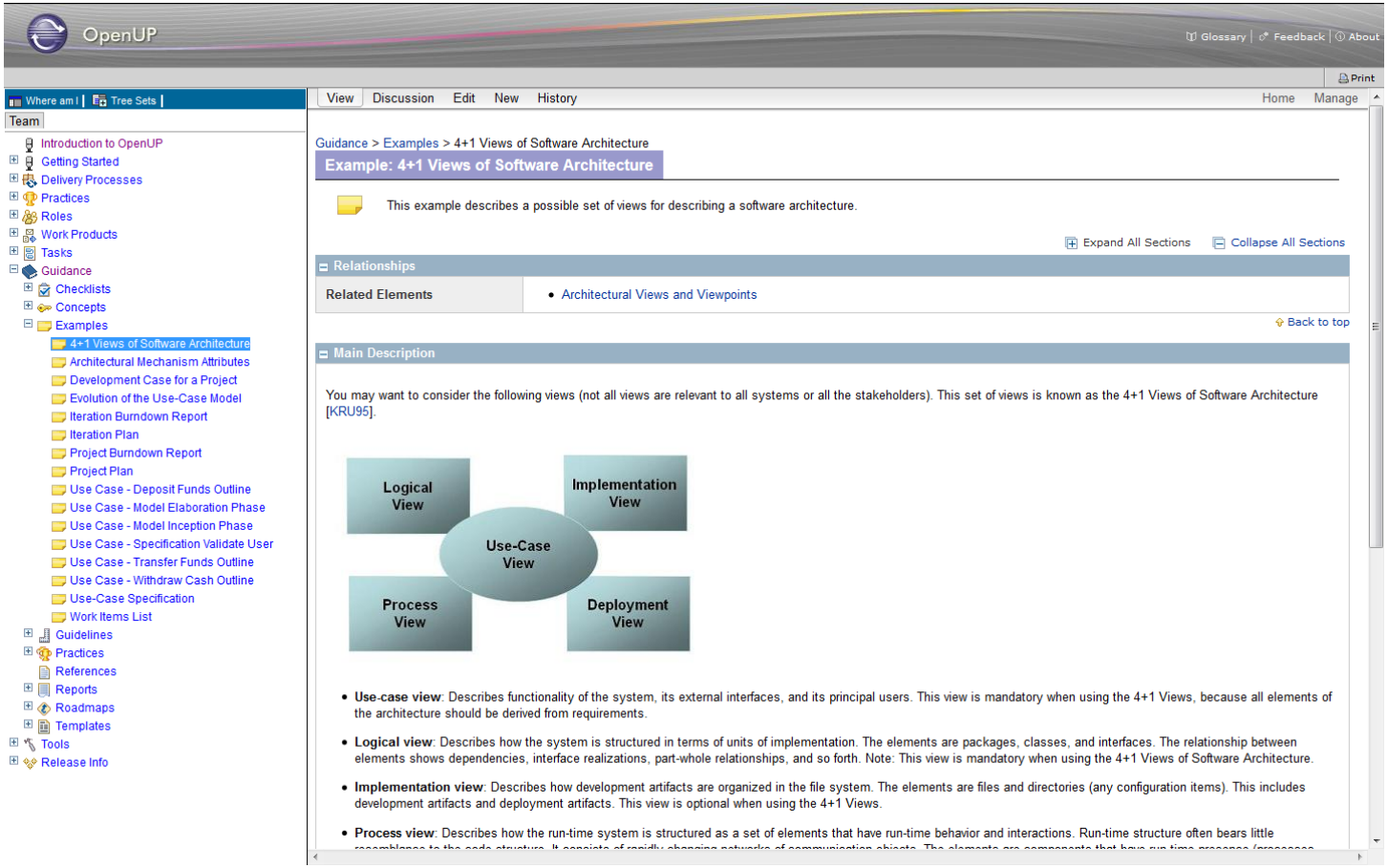
- *"Unit of work a role may be asked to perform"*
- Primary performer vs Additional performers
- Environ 35 tâches définies par défaut réparties dans 7 catégories (5 disciplines + deployment & environnement)
  Ex:
  - Envision the architecture (Architecture)
  - Package the release (Deployment)
  - Create test cases (Test)
  - Deploy the process (Environment)
  - ...

Méthode: Artefacts (4)

- *"Something that is produced, modified or used by a task."*
- Pas de formalisme dans la représentation des artefacts
  - Ex: Photo d'un tableau blanc présentant l'architecture
- Environ 30 Artifacts définis par défaut réparties dans 7 catégories
  Ex:
  - Glossary (Requirement)
  - Risk list (Project management)
  - Deployment plan (Deployment)
  - ...

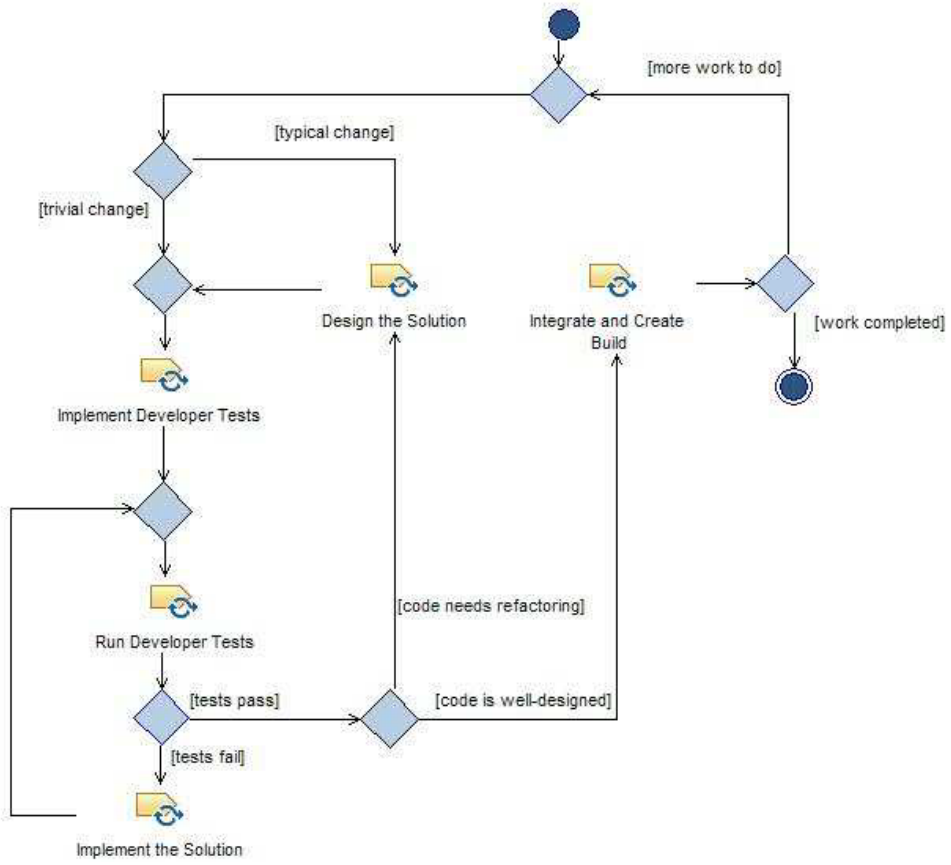
Méthode : Directives (5)

- Ensemble de conseils et de checklist utiles pour atteindre les objectifs



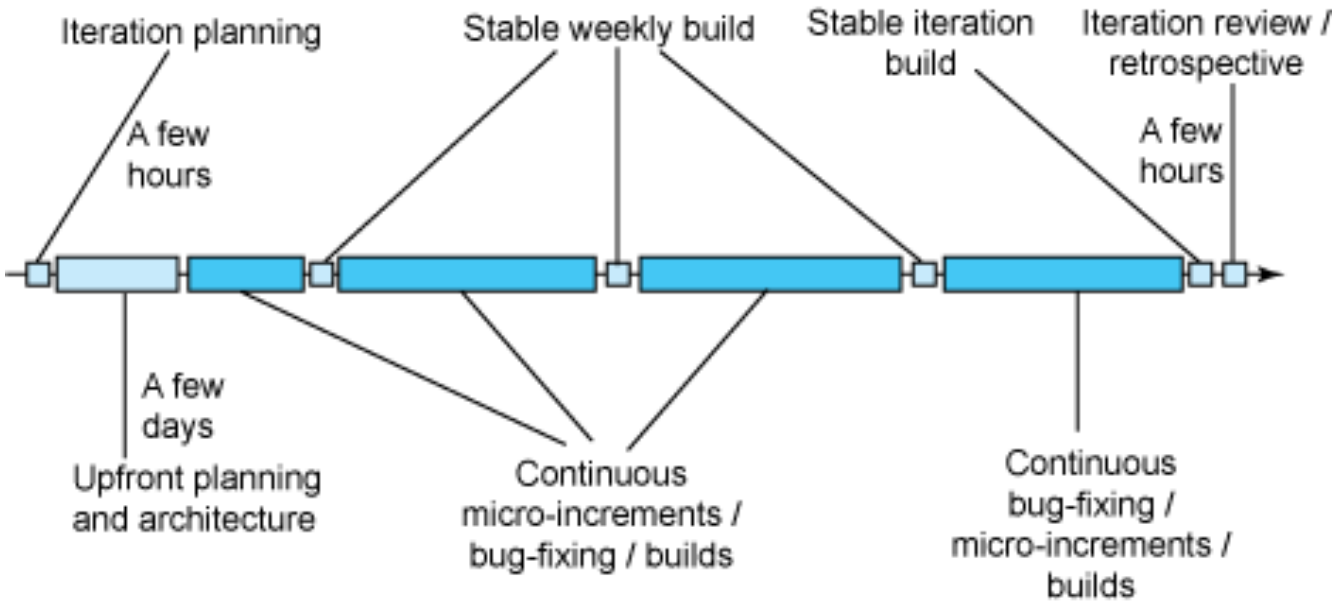
Processus : Patterns (1)

- *Définit l'ensemble des tâches à effectuer pour répondre à un besoin du projet.*
- Itéré tant que l'objectif n'est pas atteint.
- Unité de base du Delivery process



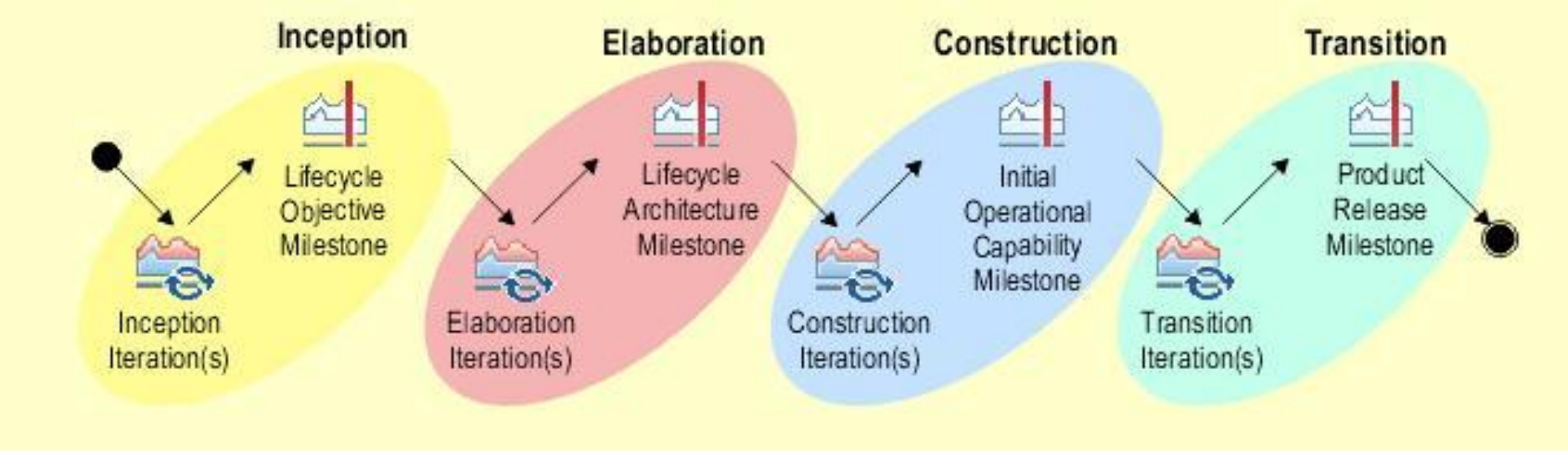
Processus : Cycle de vie d'une itération (2)

1. Meeting pour planifier l'itération
2. Exécuter et tester les micro incréments
3. Résolution des erreurs et possible ajout de nouvelles caractéristiques
4. Rétrospective et évaluation

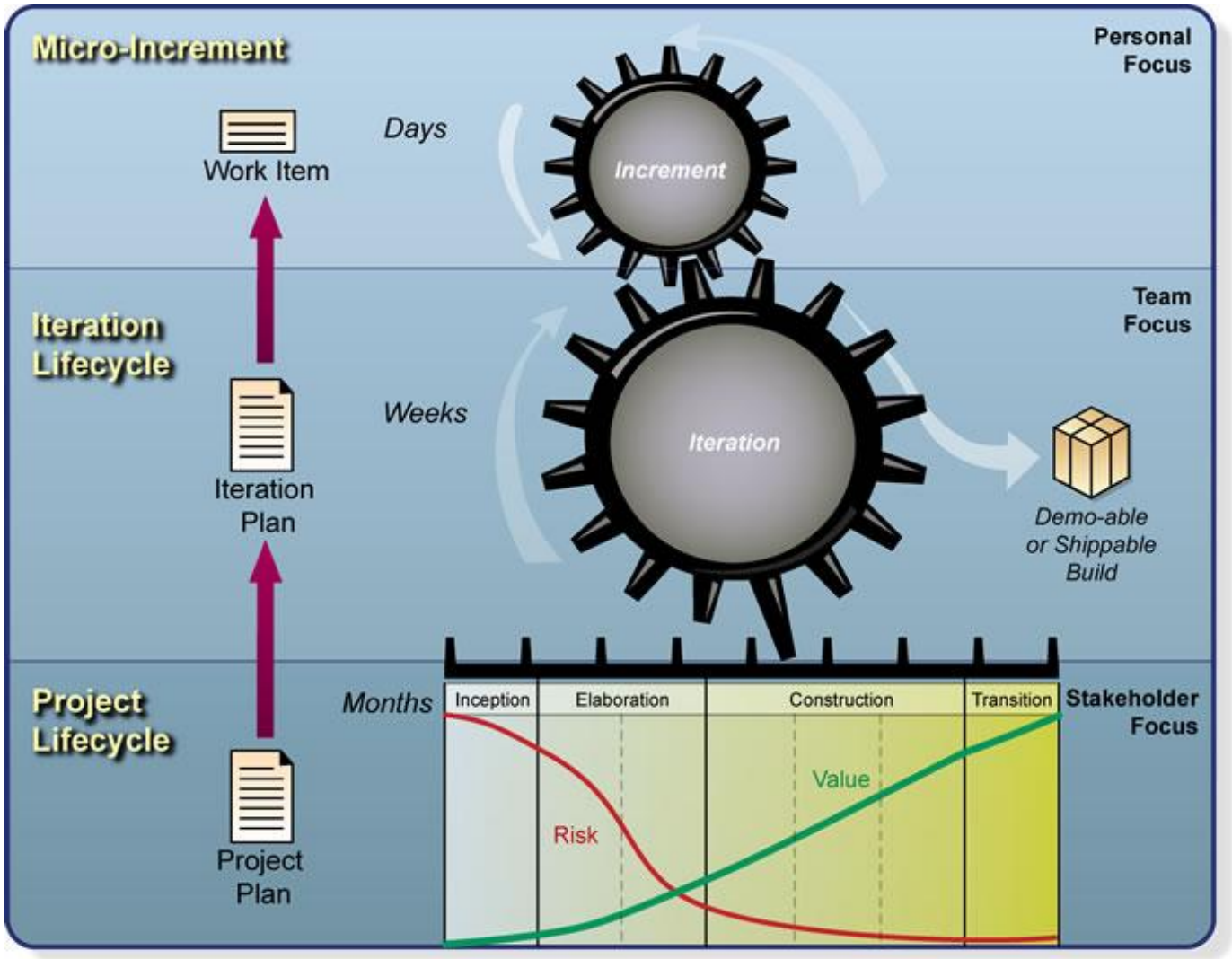


Processus : Delivery process (3)

- Décrit la vie complète du projet, mais ne remplace pas la réalité
- Composé de Patterns à itérer autant de fois que nécessaire pour chacune des 4 phases
- Le nombre d'itération dépend fort du type de projet



Micro-increment



Micro-increment

- *Résultat d'un travail pouvant aller de quelques heures à quelques jours d'une personne ou de plusieurs.*

- **BUT :**
  - Séparer le travail en plus petites unités de sorte que chacune contribue à la valeur ajoutée du projet.
  - Boucle très courte de feedback

- OpenUP ne fournit pas une liste complète des micro incréments possibles.

Influences

- **Scrum et XP** : modèle itératif court avec délivrable en fin de chaque itération
- **RUP** : reprend le concept de phase en y ajoutant la "Self-organization" et l'itération
- **Eclipse Way** : Agile et itératif

Sources

OpenUP

http://epf.eclipse.org/wikis/openup/

http://www.eclipse.org/epf/general/OpenUP.pdf

http://www.methodsandtools.com/PDF/mt200801.pdf

http://www.ibm.com/developerworks/rational/library/sep07/kroll/

EPF

http://en.wikipedia.org/wiki/Eclipse_Process_Framework

http://www.eclipse.org/epf/