Rational Unified Process
- an overview
Uppsala, 2009-02-16

Niclas Abramsson

» Consultant
» BSc Applied Statistics LU, BSc SU
» 15+ years in Software Development
» Certified Senior Project Manager (IPMA-B)
» Area responsible in Scania’s Adoption of RUP
» Mentor for Project Managers
» Teaches RUP
» Applying more and more of LEAN mindset and techniques into my work

» e-mail: niclas.abramsson@it-arkitekterna.se
Agenda

- Why a process for developing software?
- What is RUP?
- What is the “RUP-way” of developing software?
- How is RUP packaged?
- Advantages by using RUP?
- Challenges when using RUP?
- Relation to other processes?

A good project

- delivers the right solution:
  - satisfied Users
  - satisfied Sponsor
- delivers in time
- delivers on budget
Challenges

Common challenges in software development projects:
- Capture the "right" requirements.
- Requirement changes.
- Different parts of the system do not fit together.
- Small changes causes big problems.
- Low quality and poor performance.
- Major problems identified late in projects.
- Lack of communication.
- Difficult to estimate time and cost.
- Management of environment, platforms and tools.

It’s all about (lack of) communication
Agenda

- Why a process for developing software?
- **What is RUP?**
- What is the “RUP-way” of developing software?
- How is RUP packaged?
- Advantages by using RUP?
- Challenges when using RUP?
- Relation to other processes?

What is RUP and UML?

- **RUP** is a process for development of software that describes
  - what should be performed (activities)
  - who should do it (roles)
  - what the result should be (artifacts).

- **UML** is a notation with defined symbols that
  - is worked out by OMG (Object Management Group)
  - is used to make drawings (for example in software development).
Agenda

- Why a process for developing software?
- What is RUP?
- What is the “RUP-way” of developing software?
- How is RUP packaged?
- Advantages by using RUP?
- Challenges when using RUP?
- Relation to other processes?

RUP’s Six Best practices

1. Develop the systems in an iterative way. Driven by risk.
2. Manage the requirements with use cases.
3. Focus on the architecture. Build the system with well delimited components.
4. Model visually
5. Continuously review and test the system.
6. Manage changes in a controlled way
Why Iterations?

- Risk exposure
- Accuracy of estimates
- Stable Architecture

Time span till stable Architecture increases due to:
- Lack of experience: Software Development & Process knowledge
- New technology
- Magnitude of technical risks

Risk profile

- Iterative
- Waterfall
- Crew size RUP

Iterations
Use Cases

Use case outline

1. "Withdraw Money" starts when the Customer inserts a card in the ATM.
2. The System checks the card and asks for a pin-code.
3. The Customer enters a pin-code.
4. The System asks the Customer what to do next...
Always question the requirements

How large portion of functionality is used?

Source: Butler Group

Architecture

An architecture is

- an interrelationship of the parts which forms a greater whole
- the most important drawings for the system
- the bearing idea in the system solution
- something that really works/executes!
Architecture (cont.)

An architecture is
- a definition of the shape in big
- a shape that makes functions possible
- a shape that makes properties possible like
  - performance
  - load balancing
  - security
  - availability
  - adaptability
  - extensibility.

What do they have in common?

Models

Use-case model
- Actor
- Use case

Analysis/Design model
- Package
- Class
- Use case realization

Implementation model
- Component

Deployment model
- Node

...with

UNIFIED MODELING LANGUAGE
Models for visualization

Visual modeling

UC realization

Classes in the Use Case

Classes are grouped in packages

Use Case

IDE

Build

Source code

Components

Use visualization
Test continuously

Test levels in RUP:
- Unit test – done by developers
- Integration Test – done by developers
- System Test – done by testers, based on Use Cases
- Acceptance Test – done with user representatives, customer

Question: When should test work begin?
- What is the purpose of test?

Manage change in a controlled way

- **Change Request** of two types:
  - Requirement changes
  - Defects = Bugs

- Deal with changes in a controlled way involves:
  - Considering consequences before saying Yes/No!

- Configuration Management:
  - Version control
  - Enables creation of builds
Keep control of changes

Agenda

- Why a process for developing software?
- What is RUP?
- What is the “RUP-way” of developing software?
- **How is RUP packaged?**
- Advantages by using RUP?
- Challenges when using RUP?
- Relation to other processes?
RUP Overview

Inception

Objective:
- To agree on project scope
Elaboration

Objective:
- To make sure there is a technical solution

Construction

Objective:
- Finalize the software by iteratively coding and testing in every iteration
Transition

Objective:
- Deploy the software in the user’s environment

Benefits of using RUP
- Well known model – good for experience exchange
- Clearer and more correct requirements
- Increased predictability
- More efficient use of resources
- Increased visibility into project status
Challenges when using RUP

- RUP is large – not easy to find the goodies
- May lead to over-documenting
- Risk of over doing!

Relations to others

- Variants:
  - Agile Unified Process
  - Essential Unified Process
  - OpenUP
  - ..........

- What about RUP vs Agile?
Reference list