OpenUP: Comparison with RUP, and Evaluation of Tools Supporting the Process

Vangel V. Ajanovski

vangel.ajanovski@finki.ukim.mk
https://staff.finki.ukim.mk/ajanovski

Faculty of Computer Science and Engineering
Saints Cyril and Methodius University
Skopje, Macedonia
The Rational Unified Process (RUP) is covered to a large extent within several courses in the Information Systems (IS) and Software Engineering (SE) programs:

- A short introductory lecture on OpenUP would be beneficial for students
- Describe the process and differences with RUP
- Investigate and evaluate tools
- Discussion about making the choice

Relevant courses:

- *IS Development Processes (1\textsuperscript{st} cycle, 2\textsuperscript{nd} cycle)*
- *Analysis and Logical Design of IS (1\textsuperscript{st} cycle)*
- *Physical Design and Implementation of IS (1\textsuperscript{st} cycle)*
Rational Unified Process

- **RUP (Rational Unified Process)**
  - It is an Iterative software development process created by Rational
  - Some people read the title as "**Rationally** unified process", but ...

- **History:**
  - 1981 – Rational Machines was established
    - 1985 – Rational Environment (IDE for ADA) for R1000
  - 1994 – merged with Verdix => Rational Software
  - 1995 – Rumbaugh, Jacobson and Booch started unification of their work and methods => UML
    - Kruchten assembled a process framework, based on the Objectory process from Erickson’s Objectory AB => Rational Objectory Process
    - Later renamed to RUP – in line with the Unification in UML
  - 2003 – IBM acquired Rational Software
OpenUP

- History continued
  - IBM created a subset of RUP for agile projects and released as an open-source method:
    - Basic Unified Process (BUP)
  - 2005 – BUP initially went under Eclipse Foundation
  - 2006 – BUP was renamed to OpenUP/Basic

- Some characteristics of the “trim”
  - OpenUP preserves the essentials from RUP
  - Less ceremony and more agility
  - A simpler process, targeting small and collocated teams
  - Small projects – teams 3-6 people, 3-6 months
OpenUP Principles

- Core principles
  - Balance competing priorities to maximize stakeholder value
  - Collaborate to align interests and share understanding
  - Focus on the architecture early to minimize risks and organize development
  - Evolve to continuously obtain feedback and improve
RUP Main Page
OpenUP Main Page

What is OpenUP?

OpenUP is a lean Unified Process that applies iterative and incremental approaches within a structured lifecycle. OpenUP embraces a pragmatic, agile philosophy that focuses on the collaborative nature of software development. It is a tools-agnostic, low-ceremony process that can be extended to address a broad variety of project types.

Micro-Increment

Iteration Lifecycle

Project Lifecycle

Getting Started Core Principles Roles Work Products Disciplines Lifecycle
OpenUP Structured Life-cycle

- Micro-Increment: Work Item, Days
- Iteration Lifecycle: Iteration Plan, Weeks
- Project Lifecycle: Project Plan, Months

Inception, Elaboration, Construction, Transition

Risk, Value

Demo-able or Shippable Build
Personal Focus, Team Focus
Stakeholder Focus
Differences in Main Structure
Project Life-cycle
Iteration Life-cycle
OpenUP: Comparison with RUP and evaluation of the tools
Elaboration Phase

OpenUP: Comparison with RUP and evaluation of the tools
Transition Phase
Tools

- Rational Method Composer
- Eclipse EPF Composer
  - Windows and Linux
  - Latest rel.: 1.5.1.7 from July 22, 2015
  - http://www.eclipse.org/epf
- Installation problems
  - Only in 32-bit format
  - Options:
    - Install a 32-bit sub-system (real or vm)
    - Compile from source
EPF Composer Overview

Welcome! Take a few minutes to become familiar with EPF Composer.

- **Eclipse Process Framework Composer Overview**
  - Read about Eclipse Process Framework Composer's purpose and its key concepts.

- **Methods**
  - The reusable building blocks
  - Read about the role and representation of method content elements in EPF Composer.

- **Authoring Processes**
  - Read about the role and representation of processes in EPF Composer.

- **Configuration and Publishing**
  - Read about configuring and publishing method content and processes in EPF Composer.
EPF Composer Tutorials

Learn how you can use Eclipse Process Framework Composer to deliver practical, customized, and consistent process guidance to your project team.

- **Explore the user interface**
  - Learn to navigate through the user interface: perspectives, views, and search.

- **Create method content**
  - Learn how to use the editors to create roles, tasks, work products, and guidance.

- **Rouse method content**
  - Learn how to extend, replace, and contribute to base content elements.

- **Working with processes**
  - Learn how to create and reuse capability patterns and delivery processes.

- **Publishing and Exporting**
  - Learn how to publish configurations to a web site and how to export processes to a planning tool.
EPF Composer - A Multi-process tool

Libraries Download

EPF Practices
This library contains all the practices included in EPF. It is intended to be used by a process engineer to assemble, customize, and publish practices.

Download library: 1.5.1.5
Build of EPF Composer used: 1.5.1.5
Release Notes

OpenUP
The following libraries contain specific extensions and versions of OpenUP.

Download library: OpenUP/DSDM 1.0, OpenUP Portuguese 1.0
Build of EPF Composer used: 1.2.0

Scrum
The following libraries contain specific extensions and versions of Scrum.

Download library: Scrum 1.5, Scrum Portuguese 1.5
Build of EPF Composer used: 1.5

XP
The following libraries contain specific extensions and versions of XP.

Download XP 0.1 library
Build of EPF Composer used: 1.2.0M4-N.20070724-2201

Download XP Portuguese 0.1 library
Build of EPF Composer used: 1.5
EPF Composer - Using a Method Library
EPF Composer - Editing Process Elements

Guidance (Template): uc_specification

General Information
Provide general information about this template.
Name: uc_specification
Presentation name: Use-Case Specification
Type: Template
Brief description: This is the informal template suggested for representing a use case specification.

Attached file(s)/URL(s):
resources/uc_specification_tpl.dot

Detail Information
Provide detailed information about this template.
Main description:

Description | Guidance | Preview
EPF Composer - Publishing the Process

Select method configuration content
Select the method configuration content to publish. You can choose to publish the entire configuration or only a few select processes in the configuration.

- Publish the entire configuration
- Publish selected processes:
  - openup
    - Capability Patterns
      - elaboration_phase_iteration
      - inception_phaseIteration
      - transition_phaseIteration
      - construction_phaseIteration
      - initiate_project
      - plan-manage_iteration
      - ongoing_tasks
      - identify_and_refine_requirements
      - develop_architecture
      - agree_technical_approach
      - develop_solution
      - test_solution
    - Delivery Processes
      - openup_lifecycle

Publish referenced base processes
Questions and Comments