Case Example using Rational Unified Process (RUP)

SIMTech
Rational Software(Singapore), IBM

6 April 2005
Contents

• What is RUP?
• Modified RUP
  – Requirements
  – Analysis & Design
  – Implementation
  – Testing
  – Deployment
• Summary
What is RUP?

- The Rational Unified Process® or RUP® is a software engineering process developed by Rational Software, IBM.

- It provides a disciplined approach to assigning tasks and responsibilities within a development organization.

- Its goal is to ensure the production of high-quality software that meets the needs of its end users within a predictable schedule and budget.
RUP Overview

- The **first** dimension represents time and shows the lifecycle aspects of the process as it unfolds.
  - it is enacted, and it is expressed in terms of phases, iterations, and milestones.

- The **second** dimension represents the static aspect of the process:
  - how it is described in terms of process components, disciplines, activities, workflows, artifacts, and roles.

- The graph shows how the emphasis varies over time.
  - For example, in early iterations, spend more time on requirements, and in later iterations, spend more time on implementation.

Source: IBM Software
Modified RUP

- **Define Mission**
  - Identify Test Motivators
    - Functionality
    - Reliability
    - Performance
    - Interoperability
  - Agree on Mission
  - Define Assessment & Traceability Needs
  - Define Test Approach
    - API Level
    - SOAP Level
  - Define Test Ideas

- **Define Test Bed**
  - Identify Test Environment
  - Prepare H/W & S/W Infrastructure
  - Prepare Test Data Sets

- **Develop, Test & Evaluate**
  - Define Test Details
  - Implement Test
    - Generate WS Client
  - Implement Test Suite
  - Execute Test Suite
  - Analyse Test Failures
  - Determine Test Results

- **Improve Test Assets**
  - Define Test Approach
  - Identify Test Ideas
  - Prepare Guidelines for Project

- **Define**
  - Structure the Implementation Model
    - Structure the Implementation Model
  - Implement Components
    - Implement Design
    - Elements
      - Wrapping into WS
  - Integrate Each WS
    - Integrate Subsystem
      - Aggregate WS for application development

- **Structure the**

- **Define a Candidate Architecture (for each WS)**
  - Architectural Analysis
    - Identify WS Signatures
    - Identify possible 3rd party WS
  - Analyse Behaviour (for each UC)
    - UC Analysis
  - Refine the Architecture (for each WS)
    - Identify Design Elements
      - Signature Mapping Translation
      - Confirm reuse of 3rd party WS
      - Identify WS to be built
    - Identify Design Mechanisms
  - Design Components
    - UC Design
    - Subsystem Design
      - WS Signature Design
      - Class Design

- **Define System**
  - Find Actors & UCs (per WS)
  - Manage Scope Of System
    - Prioritise the Ucs (per WS)
  - Refine System Definition
    - Detail a UC
    - Detail the Software Requirements

- **Analyze Problem**
  - Capture a Common Vocabulary
  - Develop Vision (across all WS)

- **Understand Stakeholder Needs**
  - Elicit Needs
    - Categorise needs into respective WSs
    - Identify WS
  - Develop Vision
    - Refine the Categorisation Based on Features
  - Manage Dependencies
    - Prioritise the WS
    - Find Actors & UCs (per WS)

- **Define**
  - Find Actors & UCs (per WS)
  - Manage Scope Of System
    - Prioritise the Ucs (per WS)
  - Refine System Definition
    - Detail a UC
    - Detail the Software Requirements

- **Requirements**
- **Analysis and Design**
- **Implementation**
- **Testing**
- **Deployment**

Copyright © 2005 Singapore Institute of Manufacturing Technology. All rights reserved.
Modified RUP - Requirements

- **Analyze Problem**
  - Capture a Common Vocabulary
  - Develop Vision (across all WS)
- **Understand Stakeholder Needs**
  - Elicit Needs
    - Categorise needs into respective WSs
    - Identify potential WS
  - Develop Vision
    - Refine the Categorisation Based on Features
- **Manage Dependencies**
  - Prioritise the WS
- **Find Actors & UCs (per WS)**
- **Define System**
  - Find Actors & UCs (per WS)
- **Manage Scope Of System**
  - Prioritise the UCs (per WS)
- **Refine System Definition**
  - Detail a UC
  - Detail the Software Requirements

**Web Services Specific Activities**

Source: IBM Software
Activities in Requirement Discipline

- Identify stakeholders of the web services
- Define boundaries of web services
- Identify (initial) constraints to be imposed on the web services
- Formulate Problem Statement (positioning why the need to develop web services)

- Determine sources for requirements (who and where can you gather the requirements of web services)
- Gather information (based on stakeholders identified)

- Assign attributes to the features of the web services

**Requirements**

- **Analyze Problem**
  - Capture a Common Vocabulary
  - Develop Vision (across all WS)
- **Understand Stakeholder Needs**
  - Elicit Needs
    - Categorise needs into respective WSs
    - Identify potential WS
  - Develop Vision
    - Refine the Categorisation Based on Features
- **Manage Dependencies**
  - Prioritise the WS
- **Find Actors & UCs (per WS)**
- **Define System**
  - Find Actors & UCs (per WS)
- **Manage Scope Of System**
  - Prioritise the UCs (per WS)
- **Refine System Definition**
  - Detail a UC
  - Detail the Software Requirements
**Modified RUP – Analysis & Design**

**Analysis and Design**

- **Define a Candidate Architecture (for each WS)**
  - Architectural Analysis
    - Identify WS Signatures
    - Identify possible 3rd party WS
  - Analyse Behaviour (for each UC)
    - UC Analysis
- **Refine the Architecture (for each WS)**
  - Identify Design Elements
    - Signature Mapping Translation
    - Confirm reuse of 3rd party WS
    - Identify WS to be built
  - Identify Design Mechanisms
- **Design Components**
  - UC Design
  - Subsystem Design
    - WS Signature Design
  - Class Design

---

Web Services Specific Activities

---

Source: IBM Software
Activities in Analysis & Design

Analysis and Design

- Define a Candidate Architecture (for each WS)
  - Architectural Analysis
    - Identify WS Signatures
    - Identify possible 3rd party WS
- Analyse Behaviour (for each UC)
  - UC Analysis
- Refine the Architecture (for each WS)
  - Identify Design Elements
    - Confirm reuse of 3rd party WS
    - Identify WS to be built
    - Signature Mapping Translation
  - Identify Design Mechanisms
- Design Components
  - UC Design
  - Subsystem Design
    - WS Signature Design
  - Class Design

Web Services Specific Activities

- Define the high-level organization
- Functionality of Service
  - Derived from Use Case Models
- Interoperability
  - Follow best practices
- Support of Industry Standards
  - SOAP, WSDL, JAX-RPC
- Identify the use case realization for the current iteration
- Interfaces well matched with external available web services
- Identify reuse opportunities of subsystems
- Update the organization of the design model
- Decisions on signature mappings
Modified RUP - Implementation

- Structure the Implementation Model
  - Structure the Implementation Model

- Implement Components
  - Implement Design Elements
    - Wrapping into WS

- Integrate Each WS
  - Integrate Subsystem
    - Aggregate WS for Application Development

Web Services Specific Activities

Source: IBM Software
Modified RUP - Testing

• Define Mission
  • Identify Test Motivators
    • Functionality
    • Reliability
    • Performance
    • Interoperability
  • Agree on Mission
  • Define Assessment & Traceability Needs
  • Define Test Approach
    • API Level
    • SOAP Level
  • Identify Test ideas

• Define Test Bed
  • Identify Test Environment
  • Prepare H/W & S/W Infrastructure
  • Prepare Test Data Sets

• Develop, Test & Evaluate
  • Define Test Details
  • Implement Test
    • Generate WS Client
    • Implement Test Suite
    • Execute Test Suite
    • Analyse Test Failures
    • Determine Test Results

• Improve Test Assets
  • Define Test Approach
  • Identify Test Ideas
  • Prepare Guidelines for Project

Source: IBM Software
Modified RUP - Deployment

• Plan Deployment
  • Develop Deployment Plan
• Develop Support Material
  • Develop Training Material
  • Develop Support Material
• Produce Deployment Unit (WS)
  • Write Release Notes
  • Develop Installation Artifacts
  • Create Deployment Unit (WS)
  • Deploy WS to identified app servers
  • Publish WS (optional)

Web Services Specific Activities

Source: IBM Software
Summary of Web Services Specific Tasks

- Requirements Identification & Analysis
  - Categorise needs into respective web services
  - Identify potential web services
  - Use case modeling is per web service
  - Prioritise web services

- Web Services Architecting
  - Identify Web services signatures
  - Identify possible 3rd party web services
  - Identify web services to be built

- Web Services Design
  - Consider signature mapping translation
  - Refine signature

- Web Services Implementation
  - Wrap into web services
  - Aggregate Web Services

- Web Services Testing & Deployment
  - Test in both API and SOAP level
  - Deploy into App/web servers
Thank You

http://www.oasis-open.org/apps/org/workgroup/fws-i-msc/

ycheng@simtech.a-star.edu.sg