RATIONAL UNIFIED PROCESS

What is Rational Unified Process?
- Rational Unified Process® (RUP) is a software engineering process
- Goal: to ensure the production of high-quality software that meets the needs of its end-users within a predictable schedule and budget
- Developed and maintained by Rational® Software

What is RUP (cont)?
- Enhances team productivity by providing team members with easy access to a knowledge base
  - Guidelines
  - Templates
  - Tools
- Creates and maintains models – semantically rich representations of the software system under development
- Guides the team how to effectively use UML
- Is supported by tools which automate large parts of the process
- Is a configurable process that can adapt to large and small teams and projects
- Captures many of the industry best practices

Best Practices
- RUP effectively deployment of six best practices, among others
  - Develop software iteratively
  - Manage requirements
  - Use component-based architectures
  - Model software visually
  - Verify software quality
  - Control change of software

Overview of RUP
- Two dimensions, time and content
  - The time dimension represents the progress of the process as it is enacted, and is expressed in terms of cycles, phases, iterations, and milestones.
  - The content dimension represents what is being done along the respective time dimension.

The Time Dimension
- The software life-cycle is broken into cycles, each of which involves a new generation of the product
- Each cycle is divided into four phases in RUP
  - Inception phase
  - Elaboration phase
  - Construction phase
  - Transition phase
- Each phase is concluded with a well-defined milestone – a point in time at which certain critical decisions must be made and key goals must have been achieved
Two Dimensions of RUP

The Content Dimension

- Core process workflows
  - Business model
  - Requirements
  - Analysis & design
  - Implementation
  - Test
  - Deployment

The Content Dimension (cont)

- Core supporting workflows
  - Configuration & change management
  - Project management
  - Environment

Some Details of Time Dimension

- Inception phase: establish the business case for the system and delimit the project scope
  - Identify all external entities with which the system will interact
  - Identify all use cases
  - Specify success criteria
  - Assess risks
  - Estimate resources needed
  - Develop a plan showing dates of major milestones

Inception Phase (cont)

- Outcome of the inception phase (milestone)
  - A vision document: a general vision of the core project’s requirements, key features, and main constraints
  - A initial use-case model (10%-20% complete)
  - An initial project glossary
  - An initial business case which includes business context, success criteria (revenue projection, market recognition ...), and financial forecast
  - An initial risk assessment
  - A project plan showing phases and iterations
  - A business model if necessary
  - One or several prototypes

Inception Phase (cont)

- Project milestone: the Lifecycle Objectives Milestone
- Evaluation criteria
  - Stakeholder concurrence on scope definition and cost/schedule estimates
  - Requirements understanding as evidenced by the fidelity of the primary use cases
  - Credibility of the cost/schedule estimates, priorities, risks, and development process
  - Depth and breadth of any architectural prototype that was developed
  - Actual expenditures versus planned expenditures
Elaboration Phase

- Analyze the problem domain
- Establish a sound architectural foundation
- Develop the project plan
- Eliminate the highest risk elements of the project
- "mile wide and inch deep" view of the system
- An executable architecture prototype is built in one or more iterations

Elaboration Phase (cont)

- The outcome of the elaboration phase
  - A use-case model (at least 80% complete) – all use cases and actors have been identified, most use case descriptions have been developed
  - Supplementary requirements capturing the non-functional requirements and any requirements that are not associated with a specific use case
  - A software architecture description
  - An executable architecture prototype
  - A revised risk list and a revised business case
  - A development plan for the overall project, including the coarse-grained project plan, showing iterations and evaluation criteria for each iteration
  - An updated development case specifying the process to be used
  - A preliminary user manual (optional)

Elaboration Phase (cont)

- Project milestone: the Lifecycle Architecture Milestone
- Evaluation criteria
  - Stability of the vision of the product
  - Stability of the architecture
  - Demonstration of the executable showing that the major risk elements have been addressed and credibly resolved
  - Sufficiently detailed and accurate construction phase
  - Agreement of the stakeholders that the current vision is achievable in the context of the current architecture
  - Actual resource expenditure versus planned expenditure acceptable

Construction Phase

- All components are developed and integrated into the product
- All features are thoroughly tested
- Parallel constructions are possible, depending on the size of the project
- The outcome of the construction phase
  - The software product integrated on the adequate platforms
  - The user manuals
  - A description of the current release

Construction Phase (cont)

- Project milestone: Initial Operational Capability Milestone
- Evaluation criteria
  - Is this product release stable and mature enough to be deployed in the user community
  - Are all stakeholders ready for the transition into the user community
  - Are the actual resource expenditures versus planned expenditures still acceptable

Transition Phase

- Transition the software product to the user community
- Issues usually arise that require new releases, correct problems, or finish the features that were postponed
- Primary objectives
  - Achieving user self-supportability
  - Achieving stakeholder concurrence that deployment baselines are complete and consistent with the evaluation criteria of the vision
  - Achieving final product baseline as rapidly and cost effectively as practical
Transition Phase (cont)

- Expected results
  - “beta testing” to validate the new system against user expectations
  - Parallel operation with a legacy system that it is replacing
  - Conversion of the operational databases
  - Training of users and maintainers
  - Roll-out the product to the marketing, distribution, and sales teams

- Project milestone: the Product Release Milestone
- Evaluation criteria:
  - Is the user satisfied
  - Are the actual resource expenditure versus planned expenditure still acceptable

Iterations

- Each phase in RUP can be further divided into iterations
- An iteration is a complete development loop resulting in a release (internal or external) of an executable product
- A subset of the final product, which grows incrementally from iteration to become the final system

Time Distribution

- Although no projects are the same, time distribution is roughly the following
  - Inception: effort 5% schedule 10%
    - Business modeling
  - Requirement
  - Elaboration: effort 20% schedule 30%
    - Analysis
    - Design
  - Construction: effort 65% schedule 50%
    - Implementation
    - Testing
  - Transition: effort 10% schedule 10%
    - Deployment

Workers and Activities

- Worker: a worker defines the behavior and responsibilities of an individual, or a group of individuals working together as a team
  - One person can be in multiple roles
- Activity: an activity of a specific worker is a unit of work that an individual in that role may be asked to perform
- Artifact: an artifact is a piece of information that is produced, modified, or used by a process

Life-cycle Artifacts

- RUP is not document-driven
- Main artifact should be software product itself
- Documentation should remain lean and limited to the few that bring real value
- Two main categories of artifacts besides the software
- Management artifacts and technical artifacts
Life-cycle Artifacts (cont)

- Management artifacts
  - An Organizational Policy document which is the codification of the organization's process
  - A Vision document which describes the system level requirements, qualities and priorities
  - A Business Case document which contains in particular the overall iteration plan, and plan for the current and upcoming iteration
  - An Evaluation Criteria document which contains the requirements, acceptance criteria and other specific technical objectives
  - Release Description documents for each release
  - Deployment document which contains additional information useful for transition, training, installation, sales, and others
  - Status Assessment documents which contains periodic snapshots of project status with metrics of progress, staffing, expenditure, results, critical risks, actions items and others

- Technical artifacts
  - User’s Manual: developed early in the life-cycle
  - Software Documentation: preferably in the form of self-documenting source code
  - A Software Architecture document: extracted from the software documentation describing the overall structure of the software, its decomposition in major elements, and rationale for the key design decisions

Workflows – the content dimension

- Core process workflows
  - Business model
  - Requirements
  - Analysis & design
  - Implementation
  - Test
  - Deployment
- Core supporting workflows
  - Configuration & change management
  - Project management
  - Environment

Workflows (cont)

- Business modeling:
  - Common language (vocabulary) between business section and technical section
  - Business use cases
- Requirements:
  - What the system should do
  - A vision document
  - Actors are identified, entities that may interact with the system being developed
  - Use cases are identified, representing the behaviors of the system

Workflows (cont)

- Analysis and design:
  - Show how the system will be realized in the implementation phase
  - Result in a design model and optionally an analysis model
  - The design model consists of design classes structured into packages with well defined interfaces
  - An architecture is formed as the result of design

Workflows (cont)

- Implementation:
  - Define the organization of the code
  - Implement classes and objects
  - Test developed components as units
  - Integrate tested components into the system
Workflows (cont)

• Test:
  – Test for reliability, functionality, application performance and system performance
  – Verify the interaction between objects
  – Verify the proper integration of all components
  – Verify that all requirements have been correctly implemented
  – Identify defects and ensure they are addressed before release of the software

Workflows (cont)

• Deployment:
  – Product releases, delivery of the software to the users
  – Producing external releases of the software
  – Packaging the software
  – Distributing the software
  – Installing the software
  – Providing help and assistance to users (training)
  – Planning and conducting beta tests
  – Migrating of existing software or data
  – Formal acceptance

Workflows (cont)

• Project management:
  – The art of balancing competing objectives, managing risks, overcoming constraints to deliver a product that meets the needs of customers and users
  – A framework for managing software-intensive projects
  – Practical guidelines for planning, staffing, executing, and monitoring projects
  – A framework for managing risk

Workflows (cont)

• Configuration and change management:
  – How to control the numerous artifacts produced by the many people who work on a common project
  – Issues include
    • Simultaneous update
    • Limited notification
    • Multiple versions
  – Manage parallel development, multiple sites, automate build process
  – Keep an audit trail on why, when and by whom any artifact was changed

Workflows (cont)

• Environment:
  – Provide processes and tools needed to support the development team
  – Configure the process in the context of a project
  – Manage a Development Kit providing the team with guidelines, templates, and tools necessary to customize the process

RUP Products

• RUP is not just a concept. Many products exist to support it.
  • Resources include
    – A web-enabled searchable knowledge base providing all team members with guidelines, templates, and tool mentors for all critical development activities
      • Extensive guidelines
      • Tool mentors
      • IBM Rational Rose®: examples of templates
      • IBM Rational SoDA®: templates that help automate software documentation
      • Microsoft® Word templates: templates to assist documentation in all workflows and all portions of life cycle
RUP Product (cont)

- Resources include (cont)
  - Microsoft Project Plans: templates to help jump start the creation of a project plan
  - Development Kit: description of how to customize and extend the RUP to the specific needs of the adopting organization or project, and a set of tools assisting the efforts
  - Resource center containing the latest white papers, updates, hints, and techniques

RUP Product (cont)

- Tool mentors: step-by-step guide describing how to operate a tool
  - IBM Rational Requisite Pro -- keeps the entire development team updated, and on track throughout the application development process by making requirements easy to write, communicate and change
  - IBM Rational ClearQuest -- a Windows and web-based change-request management product that enables project teams to track and manage all change activities that occur throughout the development life cycle
  - IBM Rational Rose 98 -- a visual modeling tool for business process modeling, requirement analysis, and component architecture design
  - IBM Rational SoDA -- automates the production of documentation for the entire software development process, dramatically reducing documentation time and costs
  - IBM Rational Purify -- a runtime error checking tool for application and component software developers programming in C/C++

RUP Product (cont)

- Tool mentors: step-by-step guide describing how to operate a tool
  - IBM Rational Visual Quantify – an advanced performance profiling tool for application and component software developers programming in C/C++, Visual Basic, and zara
  - IBM Rational Visual PureCoverage – automatically pinpoints areas of code not exercised in testing so developers can thoroughly, efficiently and effectively test their applications
  - IBM Rational TeamTest – creates, maintains and executes automated functional tests, allowing the developing team to thoroughly test the code and determine if the software meets requirements and performs as expected
  - IBM Rational PerformanceStudio – an easy-to-use, accurate and scalable tool that measures and predicts the performance of client/server and web systems
  - IBM Rational ClearCase – a software configuration management tool, giving project managers the power to track the evolution of every software development project

A Brief History of RUP

198x Objectory process from Ivar Jacobson, developed and marketed by Objectory AB in Sweden
1992 Simplified version of Objectory published in Ivar Jacobson’s book Object Oriented Software Engineering
Oct 1995 Rational buys Objectory AB and with it Objectory V3.8
Oct 1996 Rational Objectory Process (ROP) Version 4.0, Adds iterative development and software architecture
Sep 1997 Rational Objectory Process (ROP) Version 4.1, UML replaces Objectory notation
Nov 1998 Rational Unified Process (RUP) Version 5.0, Adds business modeling, extensions in project management
June 2003 RUP 2003 (current version)