Rational Unified Process

By: Julie Johnson
November 6, 2002
Outline

- Introduction
- Best Practices
- Phases
- Workflows
- Conclusion
- References
What is RUP?

- It is a software development process, aimed at guiding software development organizations in their endeavor.
- It is a process product developed and maintained by Rational.
- Documented using UML.
Software Best Practices

- Commercially proven approaches to software development
- RUP incorporates the 6 SE best practices
  - Develop Software iteratively
  - Manage Requirements
  - Use Component-based architectures
  - Visually Model software
  - Verify Software Quality
  - Control Changes to software
Develop software iteratively
Advantages of Iterative Development

- Easier to change requirements
- Breaks down the project into smaller increments
- Early mitigation of risks
- Initial iterations enable user feedback
- Continuous testing and integration
- Easy to assess progress and project status
Incorporates the dynamic nature of requirements

RUP uses a use-case-driven approach

- Use cases defined for the system serve as the foundation for the rest of development process
- Use cases play a major role in design, testing, UI design and project management
Component-based architecture

- RUP’s design activities center on the architecture
- Architecture is validated in early iterations
- Iterative approach allows developer to identify components to develop, reuse or buy
- Uses the concept of packages, subsystems, and layers during analysis and design
- Testing is organized around components
Visually Model Software

- Uses UML- graphical notation to model throughout the SDLC
- Creates several visual artifacts
  - Domain Model
  - Use-Case Model
  - Design Model
  - Data Model
  - Implementation Model
  - Test Model
Verify software Quality

- Focuses on Product and Process Quality
- Product Quality – quality of the product and its elements
- Process Quality – Degree to which an acceptable process was implemented during manufacturing of the product
- RUP focuses on verifying and objectively assessing product and process quality
Control Changes to Software

- Changes are managed through repeatable workflows
- Manages changes across iterations—better allocation of resources
- Iterative development also allows continuous monitoring for changes
RUP Architecture

Workflows
- Business Modeling
- Requirements
- Analysis & Design
- Implementation
- Test
- Deployment
- Configuration & Change Mgmt
- Project Management
- Environment

Phases
- Inception
- Elaboration
- Construction
- Transition

Iterations
- Initial
- Elab #1
- Elab #2
- Const #1
- Const #2
- Const #N
- Tran #1
- Tran #2
A software lifecycle in RUP is broken into cycles – a cycle produces a new generation of the product.

A cycle consists of 4 phases with well-defined milestones:
- Inception - Lifecycle Objective
- Elaboration - Lifecycle Architecture
- Construction – Initial Operational Capability
- Transition - Product Release
Inception

- Establish business case and delimit project scope
  - Initial risk assessment
  - Estimate required resources
  - Success criteria
- Specifies the vision and scope
- Initial use case and domain model 10-20% complete
Elaboration

- Analyze the problem domain
- Establish sound architectural foundation
- Address elements of highest risk
- Develop Project development plan
- Use case & domain model (80% complete)
- Revised business case with risk assessment
Construction

- Incrementally develop a complete software product
- Use case and design model
- User documentation & deployment documentation
- Evaluation criteria for each iteration
- Release descriptions, including QA results
- Updated development plan
Transition

- Transition product into user community
- Produce executable results
- Updated system models, user manuals, deployment documentation, evaluation criteria for iterations
- “Post mortem” analysis of project performance
Workflows

- Sequence of activities that produces an observable result
- RUP has 9 core process workflows
  - 6 core engineering workflows
  - 3 core supporting workflows
Business Modeling

- Maintains traceability between business and software models
- Document business processes using business use cases
- Business use cases are analyzed to understand how the business should support the business processes
Requirements

- Describe *what* the system should do and come to common agreement on it
- Vision document is created – stakeholder needs
- Use cases are identified – system behavior
- Non functional requirements are documented in Supplementary specifications
- Same use case model is used during requirements, analysis and design and testing
Analysis & Design

- Shows *how* the system will be realized in the implementation phase.
- Build a system that performs desired tasks and fulfills all requirements and is robust
- Results in design model (abstraction of source code) and analysis model.
Implementation

- Define the organization of the code, in terms of implementation subsystems organized in layers
- Implement classes and objects in terms of components
- Test the developed components as units
- Integrate the results produced by individuals or teams into an executable system
Test

- Verify the interaction between objects
- Verify the proper integration of all components of the software
- Verify that all requirements have been correctly implemented
- Identify and ensure defects are addressed prior to the deployment of the software
Deployment

- Successfully produce product releases
- Deliver software to end users
- Includes activities such as
  - Producing external releases
  - Packaging, distributing and installing of software
  - Providing assistance to users
Project Management

- Risk management – risk is a driver for planning
- Develop iteration and phase plans
- Each iteration scope varies from phase to phase
- For each phase, assess trade-offs between staff, schedule and project scope
- Control project through measurements
Configuration & Change Management

- Track and maintain integrity of the project assets as they evolve in the presence of changes
- Identifies product elements, versions, and provides valid configuration of elements and workspaces
- Modify artifacts in a consistent fashion
- Status and metrics extracted from C& C management info to assist in status assessment
Environment

- Provide SDO with the software development environment – process and tools
- Focus on activities to configure the process in the context of the project
- Description on how to implement the process in an org
Conclusion

- RUP addresses many of the problems identified in the software crisis
- Incorporates software engineering best practices
- Develops software iteratively
- Provides better product and process quality
References

- Rational Whitepapers
  - What is the Rational Unified Process?
- The Rational Unified Process: An Introduction by Phillipppe Krutchen