Rational Unified Process
(RUP)

Process?
• Delivering of a software involves many steps, ranging from understanding of the problem to design, testing and implementation. A software development process takes you through these steps in a methodical way.

How should we develop a software?
Can you identify the steps involved?

Traditional Structured Analysis
• Described by W. W. Royce, 1970, IEEE WESCON, Managing the development of large software systems.
• Decomposition in terms of Function and Data
• Waterfall Method of Analysis and Design
• Part of your SE course

Waterfall Method
• Requirements Analysis
  – Analysis Specification
• Design Specification
• Coding from Design Specification
• Testing
  – Unit Testing
  – System Testing
  – UAT Testing
  – Ship It (????)
• Measuring rod is in the form of formal documents (specifications).

Waterfall Process Assumptions
• Requirements are known up front before design i.e. Users or management exactly know what they want
• Requirements rarely change
• Design can be conducted in a purely abstract space, or trial rarely leads to error
• The technology will all fit nicely into place when the time comes (the apocalypse)
• …….
Waterfall Process Limitations

- **Big Bang Delivery Theory**
- The proof of the concept is relegated to the very end of a long singular cycle. Before final integration, only documents have been produced.
- Late deployment hides many lurking risks:
  - technological (well, I thought they would work together...)
  - conceptual (well, I thought that’s what they wanted...)
  - personnel (took so long, half the team left)
  - User doesn’t get to see anything real until the very end, and they always hate it.
  - System Testing doesn’t get involved until later in the process.

The Rational Unified Process

- RUP is a method of managing OO Software Development
- It can be viewed as a Software Development Framework which is extensible and features:
  - Iterative Development
  - Requirements Management
  - Component-Based Architectural Vision
  - Visual Modeling of Systems
  - Quality Management
  - Change Control Management

RUP Features

- Online Repository of Process Information and Description in HTML format
- Templates for all major artifacts, including:
  - RequisitePro templates (requirements tracking)
  - Word Templates for Use Cases
  - Project Templates for Project Management
- Process Manuals describing key processes

The Phases

<table>
<thead>
<tr>
<th>Core Process Workflows</th>
<th>Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Modeling</td>
<td>Iterations</td>
</tr>
<tr>
<td>Requirements</td>
<td>Iterations</td>
</tr>
<tr>
<td>Analytical Design</td>
<td>Iterations</td>
</tr>
<tr>
<td>Implementation</td>
<td>Iterations</td>
</tr>
<tr>
<td>Test</td>
<td>Iterations</td>
</tr>
<tr>
<td>Deployment</td>
<td>Iterations</td>
</tr>
</tbody>
</table>

Goals and Features of Each Iteration

- The primary goal of each iteration is to slowly chip away at the risk facing the project, namely:
  - performance risks
  - integration risks (different vendors, tools, etc.)
  - conceptual risks (ferret out analysis and design flaws)
- Perform a “minewaterfall” project that ends with a delivery of something tangible in code, available for scrutiny by the interested parties, which produces validation or correctives
Risk Management

- Identification of the risks
- Iterative/Incremental Development
- The prototype or pilot project
- Early testing and deployment as opposed to late testing in traditional methods

The Development Phases

- Inception Phase
- Elaboration Phase
- Construction Phase
- Transition Phase

Inception Phase

- Overriding goal is obtaining buy-in from all interested parties
- Initial requirements capture (What user wants?)
- Cost Benefit Analysis (Making a business case)
- Initial Risk Analysis
- Project scope definition (What is included and what not?)
- Defining a candidate architecture (J2EE, .NET etc)
- Development of a disposable prototype (may be html screens only)
- Initial Use Case Model (10% - 20% complete)
- First pass at a Domain Model

Elaboration Phase

- Requirements Analysis and Capture
  - Use Case Analysis
  - Use Case (80% written and reviewed by end of phase)
  - Use Case Model (80% done)
  - Scenarios
    - Sequence and Collaboration Diagrams
    - Class, Activity, Component, State Diagrams
  - Glossary (so users and developers can speak common vocabulary)
  - Domain Model
  - To understand the problem: the system’s requirements as they exist within the context of the problem domain
  - Risk Assessment Plan revised
  - Architecture Document

Construction Phase

- Focus is on implementation of the design:
  - cumulative increase in functionality
  - greater depth of implementation (stubs flushed out)
  - greater stability begins to appear
  - implement all details, not only those of central architectural value
  - analysis continues, but design and coding predominate

Transition Phase

- The transition phase consists of the transfer of the system to the user community
- It includes manufacturing, shipping, installation, training, technical support and maintenance
- Development team begins to shrink
- Control is moved to maintenance team
- Alpha, Beta, and final releases
- Software updates
- Integration with existing systems (legacy, existing versions, etc.)