The Unified Software Development Process (UP)

Dr. Cornelius Ncube
P104D
cncube@bournemouth.ac.uk
Guiding principles

- Using the Unified Software Development Process (UP) to define the activities of requirements analysis and systems design
- The UP is the industry standard software engineering process for the UML
Course materials

- Handouts
- Labs
- Solutions

ISBN: 0321321278

For easy reference, all slides are cross referenced to sections in the book "UML 2 and the Unified Process"
- There is an example cross reference icon in the top left hand corner of this slide
Goals of the lesson

- To follow the process of requirements analysis from requirements capture through to implementation using the Unified Software Engineering Process as the framework
- To have fun!
Introduction to the Unified Process
The Unified Process (UP)

- The Unified Software Development Process is an industry standard software engineering process
  - It is commonly referred to as the "Unified Process" or UP
  - It is the generic process for the UML
  - It is free - described in "The Unified Software Development Process", ISBN:0201571692"

- UP is:
  - Use case (requirements) driven
  - Risk driven
  - Architecture centric
  - Iterative and incremental

- UP is a generic software engineering process. It has to be customised (instantiated) for your project
  - In house standards, document templates, tools, databases, lifecycle modifications, ...

- Rational Unified Process (RUP) is an instantiation of UP
  - RUP is a product marketed and owned by Rational Corporation
  - RUP also has to be instantiated for your project
UP history

- **1967**: Jacobson working at Ericsson
- **1976**: Jacobson establishes Objectory AB
- **1987**: Rational acquires Objectory AB
- **1995**: Rational Unified Process (RUP)
- **1996**: UML becomes an industry standard
- **1997**: Rational acquires Objectory AB
- **1998**: Unified Software Development Process
- **2001**: RUP 2001
- **2004**: Ongoing RUP development
Iterations

- Iterations are the key to the UP
- Each iteration is like a mini-project including:
  - Planning
  - Analysis and design
  - Integration and test
  - An internal or external release
- We arrive at a final product release through a sequence of iterations
- Iterations can overlap - this allows parallel development and flexible working in large teams
  - Requires careful planning
- Iterations are organised into phases
Each iteration may contain all of the core workflows but with a different emphasis depending on where the iteration is in the lifecycle.

UP specifies 5 *core workflows*:

- Requirements
- Analysis
- Design
- Implementation
- Test

An iteration

- Planning
- Project specific...
- Assessment

Other workflows
Baselines and increments

- Each iteration generates a baseline
- A baseline is a set of reviewed and approved artefacts that:
  - Provide an agreed basis for further review and development
  - Can be changed only through formal procedures such as configuration and change management
- An *increment* is the difference between the baseline generated by one iteration and the baseline generated by the next iteration
  - This is why the UP is called “iterative and incremental”
Each phase can include several iterations
- The exact number of iterations per phase depends on the size of the project!
  e.g. one iteration per phase for small projects
- Each phase concludes with a major milestone
Phases and Workflows

- This figure is the key to understanding UP!
- For each phase we will consider:
  - The focus in terms of the core workflows
  - The goal for the phase
  - The milestone at the end of the phase
### Inception

<table>
<thead>
<tr>
<th>Focus</th>
<th>Requirements – establish business case and scope. Capture core requirements</th>
<th>Analysis – establish feasibility</th>
<th>Design – design proof of concept or technical prototypes</th>
<th>Implementation – build proof of concept or technical prototype</th>
<th>Test – not generally applicable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goals</td>
<td>Establish feasibility of the project - create proof of concept/technical prototypes</td>
<td>Create a business case</td>
<td>Scope the system - capture key requirements</td>
<td>Identify critical risks</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Inception</th>
<th>Elaboration</th>
<th>Construction</th>
<th>Transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*amount of work in each core workflow*
Inception - milestone

- Life Cycle Objectives - conditions of satisfaction:
  - System scope has been defined
  - Key requirements for the system have been captured. These have been defined and agreed with the stakeholders
  - An architectural vision exists. This is just a sketch at this stage
  - A Risk Assessment
  - A Business Case
  - Project feasibility is confirmed
  - The stakeholders agree on the objectives of the project
## Elaboration

**Focus**

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Analysis</th>
<th>Design</th>
<th>Implementation</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>– refine system scope and requirements</td>
<td>– establish what to build</td>
<td>– create a stable architectural baseline</td>
<td>– build the architectural baseline</td>
<td>– test the architectural baseline</td>
</tr>
</tbody>
</table>

**Goals**

- Create an executable architectural baseline
- Refine Risk Assessment and define quality attributes (defect rates etc.)
- Capture use cases to 80% of the functional requirements
- Create a plan with sufficient detail for the construction phase
- Formulate a bid which includes resources, time, equipment, staff, cost

<table>
<thead>
<tr>
<th>Inception</th>
<th>Elaboration</th>
<th>Construction</th>
<th>Transition</th>
</tr>
</thead>
</table>
| ![Bar Chart](image)

---

2.9.4
Elaboration - milestone

- Lifecycle Architecture - conditions of satisfaction:
  - A resilient, robust executable architectural baseline has been created
  - The Risk Assessment has been updated
  - A project plan has been created to enable a realistic bid to be formulated
  - The business case has been verified against the plan
  - The stakeholders agree to continue
## Construction

**Focus**

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Inception</th>
<th>Elaboration</th>
<th>Construction</th>
<th>Transition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Analysis</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Design</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Implementation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Test</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Goals
- Complete use case identification, description and realization
- Finish analysis, design, implementation and test
- Maintain the integrity of the system architecture
- Revise the Risk Assessment
Construction - milestone

- Initial Operational Capability - conditions of satisfaction:
  - The product is ready for beta testing in the user environment
## Transition

<table>
<thead>
<tr>
<th>Focus</th>
<th>Requirements – not applicable</th>
<th>Analysis – not applicable</th>
<th>Design – modify the design if problems emerge in beta testing</th>
<th>Implementation – tailor the software for the user site. Fix bugs uncovered in beta testing</th>
<th>Test – perform beta testing and acceptance testing at the user site</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Inception</td>
<td>Elaboration</td>
<td>Construction</td>
<td>Transition</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Correct defects</td>
<td>Prepare the user site for the new software and tailor the software to operate at the user site</td>
<td>Modify software if unforeseen problems arise</td>
<td>Create user manuals and other documentation</td>
<td>Provide customer consultancy</td>
</tr>
</tbody>
</table>
Transition – milestone

Product Release - conditions of satisfaction:

- Beta testing, acceptance testing and defect repair are finished
- The product is released into the user community
Summary

- UP is a risk and use case driven, architecture centric, iterative and incremental software development process
- UP has four phases:
  - Inception
  - Construction
  - Elaboration
  - Transition
- Each iteration has five core workflows:
  - Requirements
  - Analysis
  - Design
  - Implementation
  - Test