UML and the Unified Process

Renaud FLORQUIN
Isabelle LECLERCQ

FloConsult SPRL
http://www.floconsult.be
mailto:info@floconsult.be
!! OO Process ≠ UML

- Unified Process
- Decision
- Abstraction
- Process
- UML
OO SW development (1)

- Iterative/incremental
- Risk-driven
- Architecture driven
Unified process phases

Inception
- Define the scope of the project and develop business case

Elaboration
- Plan project, specify features and baseline the architecture

Construction
- Build the product

Transition
- Transition the product to users
UP practices

1. Iterative/incremental software development approach
2. Risk management
3. Use Case driven
4. Architecture Centric
5. Supported by UML
1. Iterative/Incremental

**Iterative development**
- Rework scheduling strategy to revise and improve parts of the system
- Refactoring
  - e.g. only 10% of the code should be optimized for performance

**Incremental development**
- The software is developed and delivered in stages (increments)
UP: iterations

Number of iterations
- Depends on the process phase
- Generally ranges from 0 to 3

Duration of iterations
- Depends on the size of the project
- From 2 weeks to 6 months

Time-boxing:
- Iterations are fixed in length
- If needed, reduce scope, rather than slip completion date
UP iterations: small project

Iterations: from 2 to 4 weeks
UP iterations: larger project

Iterations: from 4 to 24 weeks
2. Risk management (1)

Risk reduction
- Breaks the project into small sub-projects
- Possibility to handle riskiest sub-projects at first

Elaboration (early):
- Understand better the requirements
- Baseline the architecture
  - "executable" architecture
- Main components
- Selection of technologies
- ...

FloConsult SPRL; 2003-2006
Risks management (2)

Inception | Elaboration | Construction | Transition

High | Low

Project Risk Exposure

Project Life Cycle

Waterfall

Incremental
3. Use-case driven

- **Basis for the entire development process**
  - Help to synchronize the content of various models
  - Drive numerous activities in the process
    - Capturing for functional requirements
    - Validation of the design model
    - Definition of test cases
    - Planning of iterations

Use Cases bind these workflows together
4. Architecture centric

- Models to visualize, specify, construct and document architecture
  - 4 + 1 (now N) views model of architecture
- Successive refinement of an executable architecture
  - Architectural prototypes for validation and baselines

![Diagram showing architecture development phases: Inception, Elaboration, Construction, Transition with time progression and architecture build-up](image-url)
4+1 Views

Views of a SW Architecture:
- Logical
- Process
- Implementation
- Deployment
- Use cases

Now N+1 views
5. Supported by UML

- Visual modeling
- 13 diagrams
- Most of them Object-Oriented
  - Not Use Case
  - Activity, State, ... => depends!
!! OO Process ≠ UML
BUT... What is UML?

- Unified Modeling Language
- The UML is a graphical language for
  - Specifying
  - Visualizing
  - Constructing
  - Documenting
- Standardized by the OMG
  - www.omg.org
UML = Notation

UML: Takes the best of each of the 3 methods

- OOSE (Jacobson): Use Cases
- OMT (Rumbaugh): Analysis
- Booch: Conception, Architecture

And of many others:

- Harel
- Kruchten

But only notation, impossible to standardize a unique process
# UML static diagrams

<table>
<thead>
<tr>
<th>Static Diagram</th>
<th>Notation</th>
<th>Used in ...</th>
<th>OO</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Class</strong></td>
<td>![Class Diagram]</td>
<td><strong>Inception</strong> analysis/design (Domain model)**  &lt;br&gt; <strong>Elaboration</strong> analysis/design (Design model)</td>
<td>✔</td>
</tr>
<tr>
<td><strong>Deployment</strong></td>
<td>![Deployment Diagram]</td>
<td><strong>Elaboration</strong> analysis/design (N Views)</td>
<td></td>
</tr>
<tr>
<td><strong>Component</strong></td>
<td>![Component Diagram]</td>
<td><strong>Elaboration</strong> analysis/design (N Views)</td>
<td>✔</td>
</tr>
</tbody>
</table>
# UML dynamic diagrams

<table>
<thead>
<tr>
<th>Dynamic Model</th>
<th>Notation</th>
<th>Used in ...</th>
<th>OO</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use Case</strong></td>
<td><img src="image" alt="Use Case Diagram" /></td>
<td>Inception <em>requirements</em> (High-level, 20% detailed)</td>
<td></td>
</tr>
<tr>
<td><strong>Sequence</strong></td>
<td><img src="image" alt="Sequence Diagram" /></td>
<td>Elaboration <em>requirements</em> (System Sequence Diagrams)</td>
<td>✔</td>
</tr>
<tr>
<td><strong>Collaboration (Communication)</strong></td>
<td><img src="image" alt="Collaboration Diagram" /></td>
<td>Elaboration <em>analysis/design</em> (between objects)</td>
<td>✔</td>
</tr>
<tr>
<td><strong>Activity</strong></td>
<td><img src="image" alt="Activity Diagram" /></td>
<td>Elaboration <em>requirements</em> (to illustrate Use Cases)</td>
<td>✔</td>
</tr>
<tr>
<td><strong>State</strong></td>
<td><img src="image" alt="State Diagram" /></td>
<td>Elaboration <em>analysis/design</em> (between objects)</td>
<td>✔</td>
</tr>
</tbody>
</table>

1. Use Case
   - Notation: Actor-Use Case Diagram
   - Used in: Inception requirements (High-level, 20% detailed)

2. Sequence
   - Notation: Sequence Diagram
   - Used in: Elaboration requirements (System Sequence Diagrams)

3. Collaboration (Communication)
   - Notation: Collaboration Diagram
   - Used in: Elaboration analysis/design (between objects)

4. Activity
   - Notation: Activity Diagram
   - Used in: Elaboration requirements (to illustrate Use Cases)

5. State
   - Notation: State Diagram
   - Used in: Elaboration analysis/design (between objects)
### Bibliography

<table>
<thead>
<tr>
<th>Title</th>
<th>Author(s)</th>
<th>Pages/ISBN</th>
</tr>
</thead>
</table>