RUP, Agile, XP
Outline

• The problems of waterfall
  – How to solve them

• RUP
  – Phases
  – (iterative) Activities
  – UML

• Agile
  – XP
Waterfall model

- Requirement, planning
- Design
- Implementation
- Testing
- Maintenance
What are the problems?

• 1. not friendly to changes

• 2. take long time for code delivery

• 3. difficult to do accurate planning → missing deadline
How to deliver faster?
Incremental process

- Produce core products first
- Produce further refinements in follow-up releases
Incremental process
Example

• Text editor
  – What functionality will you include in the 1\textsuperscript{st} release?
  – What about the 2\textsuperscript{nd} and 3\textsuperscript{rd} release?
How to handle changes better?
Evolutionary process

- Spiral model
Rational Unified Process

1990’
Rational Unified Process

- Basic idea: incremental/iterative
- Phases + workflows

<table>
<thead>
<tr>
<th></th>
<th>inception</th>
<th>Elaboration</th>
<th>Construction</th>
<th>transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business req.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Req. &amp; analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>design</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>impl,.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>testing.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>deployment</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Which workflow happens at which phase?
Each workflow (activity) happens in multiple phases
RUP-Unified Modeling Lang.

- What is the product of each workflow?
  - Unified Modeling Language
- Business modeling + requirement
  → Actor and use case diagram
- Analysis & design
  → class diagram, sequence diagram, state diagram
- Implementation
- Testing
- Deployment
  → deployment diagram
UML examples

• Use case diagram
Background

• Planning planning planning
  – Airplane’s control system needs 10 years to develop

• Problems
  – Too much document
  – Too late code delivery
  – Not easy to deal with changes
  – Too much bureaucracy
  – Hard to finalize design w/o implementation
  – Hard to estimate time before design & imp.
  – Hard to finish planning (prioritize) w/o estimating time
The Agile manifesto

- http://agilemanifesto.org/
12 key practices

• planning game
• small releases
• metaphor
• simple design
• testing (customer tests and tdd)
• Refactoring
• pair programming
• collective code ownership
• continuous integration
• 40 hour week
• onsite customer
• coding standards
The eXtreme Programming process

for each release/iteration (=2 weeks)
  review & planning
  design
  implementation
Planning

- Requirement document
- User stories
  - What is it?
    - A 3”X5” card with short description
  - Customer provides: user stories, value
  - Developers provide: tasks, cost (days)
  - Selection
    - What to do for this iteration
    - Open to changes
Example user stories
Design

• Principle – KIS (keep it simple)

• Output
  – CRC Card (Class-Responsibility-Collaboration)
    • Class name, class fields/methods, a list of related classes
Example (keep it simple)

**Simplicity**
```
int getSize (Vector v){
    return v.size();
}
```

**Generality**
```
int getSize (Container c) {
    Iterator i=c.iterator();
    int size =0;
    while(i.hasNext()){
        size++;
    }
    return size;
}
```
Example (CRC Card)
Design

• What is the problem of KIS?
  – Code structure degrades over time

• Solution
refactoring

• What is refactoring?
  – Code changes that do not change the external behaviors, but make the code structure better
Implementation

• TDD (test-driven development)
  – Unit tests
  – Test suite
  – Regression testing & continuous integration

• Pair programming
How to end an iteration?

- The code passes acceptance test
  - Acceptance tests are written based on the user stories
12 key practices

- planning game
- small releases
- metaphor
- simple design
- testing (customer tests and tdd)
- Refactoring
- pair programming
- collective code ownership
- continuous integration
- 40 hour week
- onsite customer
- coding standards
Did Agile solve the problems?
Challenges for Agile
Summary

- Drawbacks of waterfall
- Good practices
  - Incremental, evolutionary
- RUP
  - Separating phases and work-flows
  - UML
- Agile, XP
  - ...
  - tdd, small releases, ...