RUP, Agile, XP
Administrative stuff

• TA office hours
  – 4—6pm Th @ CSIL3
  – 3:30—5:30pm Fri @ CSIL3

• Class enrollment
Outline

• The problems of waterfall
  – How to improve waterfall?
• RUP
  – Phases
  – (iterative) Activities
  – UML
• Agile
  – XP
Waterfall model

1. Requirement
2. Design
3. Implementation
4. Testing
5. Maintenance
What are the problems?

1. difficult to handle changes
2. take long time to deliver
3. expensive to fix errors
4. difficult to estimate/planning
How to deliver faster?
Incremental process

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

- Text editor
- Class registration system
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>Phase</th>
<th>Business req.</th>
<th>Req.</th>
<th>Design</th>
<th>Impl.</th>
<th>Test</th>
<th>Deployment</th>
</tr>
</thead>
</table>

Which workflow happens at which phase?
RUP

Inception

Elaboration

Construction

Transition

Business Modelling

Requirements

Analysis & Design

Implementation

Test

Deployment
RUP

• 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
Agile

2001
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 XP process

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

- Requirement document
- User stories
  - What is it?
  - Customer provides ...
  - Developers provide ...
Example user stories
Design

• Principle – KIS (keep it simple)

• Output
  – CRC Card (Class-Responsibility-Collaboration)
Example (keep it simple)

Simplicity  Generality
Example (CRC Card)
Design

• What is the problem of KIS?

• Solution
refactoring

• What is refactoring?
Implementation

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

• Pair programming
Implementation

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

• Pair programming
How to end an iteration?
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, ...
A few project example

- https://github.com/catherinemoresco/PDFProject
- https://github.com/courageousillumination/deckr
- https://github.com/dyxh/cs220