Welcome to CS220
Software Construction

January 3rd, 2017
Shan Lu

https://www.classes.cs.uchicago.edu/archive/2017/winter/22001-1/
Outline

• Technical stuff
  – What is software engineering
    • What are the goals & challenges
  – What is a software engineering process
    • Waterfall model

• Administrative stuff
  – Who I am
  – Components/tasks/schedule of this class

• A brief history of software engineering
My background

• Shan Lu
  • Ry 257-A, shanlu@cs.uchicago.edu
  • Office hours: 4:15—5pm, Tu/Th

– East China ➔ Illinois ➔ Wisconsin ➔ Illinois

– Research
  • Software reliability, software efficiency, etc.

– Teaching
  • I enjoy discussion
Our TA

• Zewei Chu
  – zeweichu@cs.uchicago.edu
  – Office hour: 3—5pm Thursday

• Hua Li
  – hli5@uchicago.edu
  – Office hour: 4—6pm Wednesday
Your background?

• How many programs have you written?
  – What are the sizes of your programs?
• What programming languages do you use?
• How familiar are you with O-O?
Engineering
Software Construction

Concepts & Practices
--- An engineering discipline about all aspects of software production
What are the aspects of S.E.?
What are the aspects of S. production?

• Gathering requirements
• Design
• Development
• Testing & debugging
• Maintenance
What is the goal of S.E.?

• What are the criteria for good programmers?

• What are the criteria for good software?

• The goal of software engineering is ...
What is the goal of S.E.?

• What are the criteria for good programmers?
  – Write good software
  – Be on time

• What are the criteria for good software?
  – Reliable/correct (few bugs)
  – Efficient (run fast)
  – Maintainnable
  – Good usability
  – Good security

• The goal of software engineering is
  – Produce good software, within time schedule, within resource budget
What are the challenges?
What are the challenges?

• Large code sizes
    – Linux Kernel 1.0.0 (1994) 100K+
    – Linux Kernel 2.2.0 (1999) ?
    – Hubble Space Telescope ?
    – Chrome? Firefox?
    – Boeing 787?
    – Mac OS X Tiger?
    – Car software
    – healthcare.gov

• Changing requirements
  – User, hardware, ...

• Large development team (at different geo locations)
Google

- 15000+ developers in 40+ offices
- 4000+ projects under active development
- 5500+ submissions per day on average
- Single monolithic code tree with mixed language code
- Development on one branch - submissions at head
- All builds from source
- 20+ sustained code changes per minute with 60+ peaks
- 50% of code changes monthly
- 75+ million test cases run per day
How to ...?

- Practices/disciplines

- Tools
Engineering
Software Construction

--- Practices and tools about design, development, and maintenance of software
S.E. process

• A sequence of activities that lead to the production of a software product

• There are many processes proposed
  – Waterfall
  – RUP (Rational Unified Process)
  – Agile
    • Extreme programming
Waterfall model

• Activities ➔ separate process phases
Waterfall model

The classic waterfall development model

- Requirements/analysis
- Design
- Coding
- Testing
- Maintenance
Waterfall model phase I

• Requirement & analysis

• Where do we obtain the requirement?
• Should we modify or refine the requirements?
  – What should we consider?

• Output
Waterfall model phase II

• Design

• What need to be designed?

• Output
Waterfall model phase II

• Design

• What need to be designed?
  – UI
  – Data structure (component design)
  – Module, API interface (architecture design)

• Output
  – Design document
Waterfall model phase III

- Implementation

- Output
Waterfall model phase IV

- Testing

- Output
Waterfall model phase IV

- Testing

- Output
Waterfall model phase V

- Maintenance

- Ratio of cost among phases
Problems with waterfall model
Problems with waterfall model

- Difficult to handle changes (not in model, high cost)
- Error fixing expensive
- Hard to estimate time
Administrative Stuff
## An overview of our schedule

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>01/03</td>
<td>Introduction, Software Processes [notes]</td>
</tr>
<tr>
<td>01/10</td>
<td>Requirement Engineering &amp; System Modeling I [notes]</td>
</tr>
<tr>
<td>01/17</td>
<td>System Modeling III [notes]</td>
</tr>
<tr>
<td>01/24</td>
<td>Testing 1 [notes]</td>
</tr>
<tr>
<td>01/31</td>
<td>Code Smell</td>
</tr>
<tr>
<td>02/07</td>
<td>Midterm</td>
</tr>
<tr>
<td>02/14</td>
<td>Design Patterns II (Composite, Interpreter, Strategy) [notes]</td>
</tr>
<tr>
<td>02/21</td>
<td>OO and Design Patterns IV [notes]</td>
</tr>
<tr>
<td>02/28</td>
<td>Parallel &amp; Distributed Software II</td>
</tr>
<tr>
<td>03/07</td>
<td>Bugs &amp; Debugging II</td>
</tr>
<tr>
<td>01/05</td>
<td>Agile, Extreme Programming, Project Discussion [notes]</td>
</tr>
<tr>
<td>01/12</td>
<td>Requirement Engineering &amp; System Modeling II [notes]</td>
</tr>
<tr>
<td>01/19</td>
<td>Architectural Design [notes]</td>
</tr>
<tr>
<td>01/26</td>
<td>Testing 2 [notes]</td>
</tr>
<tr>
<td>02/02</td>
<td>Refactoring [notes] &amp; Software Maintenance [notes]</td>
</tr>
<tr>
<td>02/09</td>
<td>OO and Design Patterns I (Observer) [notes]</td>
</tr>
<tr>
<td>02/16</td>
<td>Design Patterns III [notes]</td>
</tr>
<tr>
<td>02/23</td>
<td>Parallel &amp; Distributed Software I</td>
</tr>
<tr>
<td>03/02</td>
<td>Bugs &amp; Debugging I [notes]</td>
</tr>
<tr>
<td>03/09</td>
<td>No Class (Reading Period)</td>
</tr>
</tbody>
</table>

---

**Any student graduating at the end of this quarter?**
There are a **lot** of work to do

- Class
- 1 mini project (due 1/26) 8%
- 1 big programming project 45%
  - Many milestones/checkpoints
- Weekly Quiz 7%
- Two exams 40%

*If you are going to drop this course, do it soon.*
What you need to do 1: lectures & reading

• Lectures
  – Tu/Th 3:00—4:20 pm
What you need to do 2: Quizzes

• ~10 minutes @ every Tuesday lecture
• The 1st quiz is on January 10th (next Tuesday)
• Close-book, close-note
• Cover lectures and project content

• 1 point for each quiz, 7% of your overall grades

*: unless ...
What you need to do 3: Project

- Course project
  - 6—8 people a group
  - The whole process
  - 6+ milestones

<table>
<thead>
<tr>
<th>Date</th>
<th>Number</th>
<th>Milestone</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/17</td>
<td>1</td>
<td>Proposal (2—3 students)</td>
</tr>
<tr>
<td>2/02</td>
<td>2</td>
<td>Planning (6—8 students)</td>
</tr>
<tr>
<td>2/10</td>
<td>3.a</td>
<td>Testing of 1st iteration</td>
</tr>
<tr>
<td>2/17</td>
<td>3.b</td>
<td>End of 1st iteration</td>
</tr>
<tr>
<td>2/24</td>
<td>4.a</td>
<td>Testing of 2nd iteration</td>
</tr>
<tr>
<td>3/03</td>
<td>4.b</td>
<td>End of 2nd iteration</td>
</tr>
<tr>
<td>3/08</td>
<td>5</td>
<td>System testing &amp; documentation</td>
</tr>
<tr>
<td>3/13</td>
<td>6</td>
<td>Acceptance testing &amp; debugging</td>
</tr>
</tbody>
</table>

- 45 % of your final grade
What you need to do 4: warm-up project

- One warm-up project
- It is due on 1/26th
What you need to do 5: Exams

• Midterm exam
  – In the lecture on 02/07
  – 20% of your final grades

• Final exam
  – During the exam week
  – 20% of your final grades

• Cover material from class and the projects
Resources

• CSIL Labs

• TA
  – Zewei Chu, zeweichu@cs.uchicago.edu
    • Thursday, 3pm – 5pm @ CSIL??
  – Hua Li, hli5@uchicago.edu
    • Wednesday, 4pm—6pm @ CSIL??
  – Starting from next week

• Piazza!! (will start by the end of this week)
• Feel free to ask me questions in&off class
A brief history I

• The pioneering era
  – No S.E.
  – No way to estimate s/w development time
  – s.w. is free

• Starting 1960s

• The Software Crisis 1965--1985
  – Therac 25 1985—1987
  – Morris worm 1988
A brief history II

• 1985 – 2000
  – No silver bullet
  – **OO, design patterns**, formal methods, **process**

• 2000 – present
  – Agile
  – Model-driven design

  – Tools, Program synthesis, ...
Current S.E. research

Belvin's Law 3

YESSS!

If the code works the first time, that just means that the bug is more carefully hidden.

Gee, thanks for that.
Summary

• What we discussed
  – What is software engineering
  – What is s.e. process
  – Waterfall model

• What you should do/prepare to do
  – Submit your survey
  – Check course webpage
  – Check piazza
  – Quiz
  – Mini-project to be released in two days
  – Project proposal