Welcome to CS220

September 30<sup>st</sup>, 2014
Shan Lu

http://www.classes.cs.uchicago.edu/archive/2014/fall/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: TBA
  
  – East China → Illinois → Wisconsin → Illinois

  – Research
    • Software reliability, concurrency, etc.

  – Teaching
    • I enjoy discussion
Our TAs

• Zhixuan Zhou
  – zhixuanzhou@...
  – Friday 2pm—4pm @ CSIL 4

• Sean Laguna
  – slaguna@...
  – Monday 8pm—10pm @ CSIL 4
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?
• How familiar are you with STL?

• Please fill the survey
Software Engineering
Concepts & Practices
--- An engineering discipline about all aspects of software production
What are the aspects of S. production?

• Gathering requirements
• Design
  – Planning & delegating task
• Development
• Testing & debugging
• Maintenance
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)
  – Maintainable
  – Good usability
  – Good security

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

• Large code sizes
  – Linux Kernel 1.0.0 (1994) 100K+
  – Linux Kernel 2.2.0 (1999) 2 million
  – Hubble Space Telescope 2 million
  – Chrome? Firefox? 5 million
  – Boeing 787? 10 million
  – Mac OS X Tiger?
  – Car software 100 million
  – healthcare.gov 500 million

• 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
--- 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?
  – Client

• Should we modify or refine the requirements?
  – What should we consider?

• Output
  – Requirement document
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
  - Code
Waterfall model phase IV

- Testing
- Output
Waterfall model phase IV

- Testing

- Output
Waterfall model phase V

- Maintenance

- Ratio of cost among phases
  - Requirement + design + coding : testing is about 3:2
Problems with waterfall model
Administrative Stuff
An overview of our schedule

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>9/30</td>
<td>Intro</td>
<td>10/02</td>
<td>Agile, XP</td>
</tr>
<tr>
<td>10/07</td>
<td>Req. &amp; Modeling I</td>
<td>10/09</td>
<td>Req. &amp; Modeling II</td>
</tr>
<tr>
<td>10/07</td>
<td>Mini-Project Due</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/14</td>
<td>Planning</td>
<td>10/16</td>
<td>Arch. Design</td>
</tr>
<tr>
<td>10/21</td>
<td>Testing I</td>
<td>10/23</td>
<td>Midterm</td>
</tr>
<tr>
<td>10/28</td>
<td>Testing II</td>
<td>10/30</td>
<td>OO Design Pattern I</td>
</tr>
<tr>
<td>11/04</td>
<td>OO Design Pattern II</td>
<td>11/06</td>
<td>Quality &amp; Metrics</td>
</tr>
<tr>
<td>11/11</td>
<td>Refactoring</td>
<td>11/13</td>
<td>Bugs &amp; Debugging</td>
</tr>
<tr>
<td>11/18</td>
<td>Advanced Topics I</td>
<td>11/20</td>
<td>Advanced Topics II</td>
</tr>
<tr>
<td>11/25</td>
<td>no class</td>
<td>11/27</td>
<td>no class</td>
</tr>
<tr>
<td>12/02</td>
<td>Maintenance</td>
<td>12/04</td>
<td>no class</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Project M.S. 6</td>
</tr>
</tbody>
</table>

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

- Class
- 1 or 2 mini projects 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
What you need to do 2: Quizzes

• ~10 minutes @ every Tuesday lecture
• The 1\textsuperscript{st} quiz is on Oct. 7\textsuperscript{th} (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>Milestone</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/15</td>
<td>1</td>
<td>Proposal (2—3 students)</td>
</tr>
<tr>
<td>10/29</td>
<td>2</td>
<td>Planning (6—8 students)</td>
</tr>
<tr>
<td>11/05</td>
<td>3.a</td>
<td>Testing of 1st iteration</td>
</tr>
<tr>
<td>11/12</td>
<td>3.b</td>
<td>End of 1st iteration</td>
</tr>
<tr>
<td>11/19</td>
<td>4.a</td>
<td>Testing of 2nd iteration</td>
</tr>
<tr>
<td>11/25</td>
<td>4.b</td>
<td>End of 2nd iteration</td>
</tr>
<tr>
<td>12/03</td>
<td>5</td>
<td>System testing &amp; documentation</td>
</tr>
<tr>
<td>12/07</td>
<td>6</td>
<td>Acceptance testing &amp; debugging</td>
</tr>
</tbody>
</table>

– 45 % of your final grade
What you need to do 4: mini projects

• One or two mini-projects

• The 1\textsuperscript{st} mini-project will be due on 10/8\textsuperscript{th}
What you need to do 5: Exams

• Midterm exam
  – In the lecture on 10/23
  – 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 office hours
  – Sean Laguna, slaguna@cs.uchicago.edu
    • Monday, 8pm – 10pm @ CSIL 4
  – Zhixuan Zhou, zhixuanzhou@cs.uchicago.edu
    • Friday, 2pm—4pm @ CSIL 4

• Piazza!!

• 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**
Current S.E. research
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