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: right after each lecture

  – East China → Illinois → Wisconsin→Illinois

  – Research
    • Software reliability, software efficiency, etc.

  – Teaching
    • I enjoy discussion
Our TA

• Maria Hyun
  – mhyun@uchicago.edu
  – Office hour
    • This week: Friday 1—3pm @ CSIL4
    • Remaining weeks: TBA
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?
Engineering

Software Construction

Concepts & Practices
Software Engineering

--- 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)
  – Maintainable
  – 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
  - [http://www.informationisbeautiful.net/visualizations/million-lines-of-code/](http://www.informationisbeautiful.net/visualizations/million-lines-of-code/)
  - 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
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
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>Event</th>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>03/29</td>
<td>Intro</td>
<td>03/31</td>
<td>Agile, XP</td>
</tr>
<tr>
<td>04/05</td>
<td>Project discussion</td>
<td>04/07</td>
<td>Req. &amp; Modeling I</td>
</tr>
<tr>
<td>04/12</td>
<td>Req. &amp; Modeling II</td>
<td>04/14</td>
<td>Modeling III</td>
</tr>
<tr>
<td>04/19</td>
<td>Arch. Design</td>
<td>04/21</td>
<td>Midterm</td>
</tr>
<tr>
<td>04/26</td>
<td>Testing I</td>
<td>04/28</td>
<td>Testing II</td>
</tr>
<tr>
<td>05/03</td>
<td>OO Design Pattern</td>
<td>05/05</td>
<td>OO Design Pattern</td>
</tr>
<tr>
<td>05/10</td>
<td>OO Design Pattern</td>
<td>05/12</td>
<td>OO Design Pattern</td>
</tr>
<tr>
<td>05/17</td>
<td>Quality &amp; Metrics</td>
<td>05/19</td>
<td>Refactoring</td>
</tr>
<tr>
<td>05/24</td>
<td>Bugs &amp; Debugging</td>
<td>05/26</td>
<td>Maintenance</td>
</tr>
<tr>
<td>05/31</td>
<td>TBA</td>
<td>06/02</td>
<td>no class</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>05/05</td>
<td>Project M.S. 3.a</td>
</tr>
<tr>
<td>05/12</td>
<td>Project M.S. 3.b</td>
</tr>
<tr>
<td>05/19</td>
<td>Project M.S. 4.a</td>
</tr>
<tr>
<td>05/26</td>
<td>Project M.S. 4.b</td>
</tr>
<tr>
<td>05/31</td>
<td>Project M.S. 5</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 next week) 8%
  • Many milestones/checkpoints

• 1 big programming project 45%

• 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 1\textsuperscript{st} quiz is on April 5\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>4/13</td>
<td>1</td>
<td>Proposal (2—3 students)</td>
</tr>
<tr>
<td>4/27</td>
<td>2</td>
<td>Planning (6—8 students)</td>
</tr>
<tr>
<td>5/04</td>
<td>3.a</td>
<td>Testing of 1st iteration</td>
</tr>
<tr>
<td>5/11</td>
<td>3.b</td>
<td>End of 1st iteration</td>
</tr>
<tr>
<td>5/18</td>
<td>4.a</td>
<td>Testing of 2nd iteration</td>
</tr>
<tr>
<td>5/24</td>
<td>4.b</td>
<td>End of 2nd iteration</td>
</tr>
<tr>
<td>5/30</td>
<td>5</td>
<td>System testing &amp; documentation</td>
</tr>
<tr>
<td>6/05</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 4/7\textsuperscript{th}
What you need to do 5: Exams

• Midterm exam
  – In the lecture on 04/21
  – 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
  – Maria, mhyun@uchicago.edu
    • Friday, 1pm – 3pm @ CSIL4

• 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