Welcome to CS220
Software Construction

October 2^{nd}, 2018
Shan Lu

https://www.classes.cs.uchicago.edu/archive/2018/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
  - JCL 343, shanlu@cs.uchicago.edu
  - Office hours: after-class—4:15pm, Tu/Th

- East China → Illinois → Wisconsin → Illinois

- Research
  - Software reliability, software efficiency, etc.

- Teaching
  - I enjoy discussion
Our TA / Grader

• Yuxi Chen
  – chenyuxi@uchicago.edu
  – Office hour: 4—6pm Tu/Th @ CSIL1

• Hussein Elkhesheen
  – husseinelkheshen@uchicago.edu
  – TBD
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
--- An engineering discipline about all aspects of software production
What do you do to produce software?
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
  – 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
  – Module, API interface (architecture design)
  – Data structure (component 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

The classic waterfall development model

- Requirements/analysis
- Design
- Coding
- Testing
- Maintenance
Problems with waterfall model

- Difficult to handle changes (not in model, high cost)
- Error fixing expensive
- Hard to estimate time
More information at ...

- Chapter 2.0, 2.1, 2.2.0, 4.1.0, 4.1.1
- Chapter 1.1.1, 2.1.0, 2.1.1
Administrative Stuff
An overview of our schedule

10/02 Introduction, Software Processes [notes]
10/09 Project Discussion, Requirement Engineering & System Modeling I [notes]
10/16 System Modeling III [notes]
10/23 Testing 1 [notes]
10/30 Code Smell [notes]
11/06 Midterm
11/13 Design Patterns II (Composite, Interpreter) [notes]
11/20 (No Class) Happy Thanksgiving!
11/27 Bugs and Bug Detection [notes]
12/04 Project Presentation

10/04 Agile, Extreme Programming [notes]
10/11 Requirement Engineering & System Modeling II [notes]
10/18 Architectural Design [notes]
10/25 Testing 2 [notes]
11/01 Refactoring [notes]
11/08 Software Maintenance and Design Patterns I (Observer) [notes]
11/15 Design Patterns III [notes]
11/22 (No Class) Happy Thanksgiving!
11/29 Parallel Software Construction [notes]
12/06 No Class (Reading Period)

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

- Class

- 1 mini project (due 10/15)  8%
- 1 big programming project  40%
  - Many milestones/checkpoints
  - Proposal due 10/17
- Weekly Quiz  7%
  - First one on 10/9
- Two exams  45%

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

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

• ~10 minutes @ every Tuesday lecture
• The 1\textsuperscript{st} quiz is on October 9\textsuperscript{th} (next Tuesday)
• Close-book, close-note
• Cover lectures and project content

• 1 point for each quiz, 7% of your overall grades
What you need to do 3: Project

- Course project
  - 7—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>10/17</td>
<td>1</td>
<td>Proposal (2—3 students)</td>
</tr>
<tr>
<td>10/30</td>
<td>2</td>
<td>Planning (7—8 students)</td>
</tr>
<tr>
<td>11/07</td>
<td>3.a</td>
<td>Testing of 1\textsuperscript{st} iteration</td>
</tr>
<tr>
<td>11/13</td>
<td>3.b</td>
<td>End of 1\textsuperscript{st} iteration</td>
</tr>
<tr>
<td>11/20</td>
<td>4.a</td>
<td>Testing of 2\textsuperscript{nd} iteration</td>
</tr>
<tr>
<td>11/29</td>
<td>4.b</td>
<td>End of 2\textsuperscript{nd} iteration</td>
</tr>
<tr>
<td>12/04</td>
<td>5</td>
<td>System testing &amp; documentation</td>
</tr>
<tr>
<td>12/09</td>
<td>6</td>
<td>Acceptance testing &amp; debugging</td>
</tr>
</tbody>
</table>

- 40 % of your final grade
- Grading criteria: 75% group + 25% individual
- There will be peer evaluation
What you need to do 4: warm-up project

• One warm-up project
  – Will be released today or tomorrow
  – Do it in a group of two people

• It is due on 10/15th
What you need to do 5: Exams

• Midterm exam
  – In the lecture on 11/06
  – 20% of your final grades

• Final exam
  – During the exam week
  – 25% of your final grades

• Cover material from class and the projects
Overall Grade

- Curved

- 2018 winter
  - A* 25; B* 5; C* 2

- 2017 winter
  - A* 19; B* 8; C* 5

- 2014 Fall
  - A* 22; B* 14; C* 4
Resources

• CSIL Labs

• TA
  – Yuxi Chen, chenyuxi@uchicago.edu

• 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, verification, ...
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
  – Check course webpage
  – Check piazza
  – Quiz
  – Mini-project to be released soon
  – Project proposal