Overview

• A bit about me
• A bit about you
• Course goals
• Organization
About Me

• PhD in Computer Science from The University of Texas at Austin, 2014
• Post doc in the same department for seven months
• Assistant Professor in Computer Science of Virginia Tech since August, 2015

Research Interests

• Software Engineering
  – Empirical study
    • To understand how developers maintain software and make code changes
  – Design and implementation of new techniques
    • To assist developers maintain software by finding bugs, diagnosing root causes, and suggesting code changes
About You

• Your name?
• Master or PhD?
• Research interest?
• Why are you in graduate school?

Course Goals

• Intellectual development
  – Good understanding of problems and techniques in Software Engineering
  – Knowledge of advanced tools which can assist software development

• Practical development
  – Improve implementation and writing
  – Produce interesting research outcome
Course Organization

- Introduction of Software Engineering (3 weeks)
  - software process, Object-Oriented analysis & design, etc.
- Introduction of research topics in SE
  - empirical study, delta debugging, fault localization
- Introduction of frequent techniques used in SE research/software development
  - program differencing, clone detection, etc.

Grading Policy

- Project: 55%
- Critiques: 30%
- Attendance and class participation 15%
Class Discussion

• Ask clarifying questions or challenging questions
• Answer other people’s question based on your paper comprehension and research experience
• Deep and hard questions are highly encouraged!

Ethics

• The work you turn in must be your own
• If you copy any sentence to your critiques, you should cite the source
• Everything you write or present should be correct to the best of your knowledge
Introduction to Software Engineering

Overview

• Software in our lives
• Hardware vs. Software
• What is software engineering?
Software is ubiquitous

<table>
<thead>
<tr>
<th>Type of Software</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>System software</td>
<td>OS, compilers, device drivers</td>
</tr>
<tr>
<td>Business software</td>
<td>Payroll, accounting</td>
</tr>
<tr>
<td>Engineering/scientific software</td>
<td>Computer-aided design, simulation</td>
</tr>
<tr>
<td>Embedded software</td>
<td>GPS navigation, Flight control, Toaster</td>
</tr>
<tr>
<td>Product-line software (PC-like based)</td>
<td>Spreadsheets, word processing, games</td>
</tr>
<tr>
<td>Web-based software</td>
<td>Gmail, Facebook, Youtube</td>
</tr>
<tr>
<td>Artificial intelligence software</td>
<td>Robotics, artificial neural networks, theorem proving</td>
</tr>
</tbody>
</table>
What is Software?

• Definition [Pressman]
  – The product that software professionals build and then support over the long term

• Software encompasses:
  – Executable programs
  – Data associated with these programs
  – Documents: user requirements, design documents, user/programmer guides

Hardware vs. Software

☐ Manufactured
☐ Wear out
☐ Built using components
☐ Relatively simple

☐ Developed/ engineered
☐ Deteriorate
☐ Custom built
☐ Complex
### Manufacturing vs. Development

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hardware</strong></td>
<td><strong>Software</strong></td>
</tr>
<tr>
<td>is difficult</td>
<td>is routinely</td>
</tr>
<tr>
<td>or impossible</td>
<td>modified and</td>
</tr>
<tr>
<td>to modify</td>
<td>upgraded</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Hiring</strong></td>
<td><strong>This</strong></td>
</tr>
<tr>
<td>more people</td>
<td>is not always</td>
</tr>
<tr>
<td>causes more</td>
<td>true</td>
</tr>
<tr>
<td>work done</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Costs</strong></td>
<td><strong>Costs</strong></td>
</tr>
<tr>
<td>are more</td>
<td>are more</td>
</tr>
<tr>
<td>concentrated</td>
<td>concentrated</td>
</tr>
<tr>
<td>on products</td>
<td>on design</td>
</tr>
</tbody>
</table>

Hardware does “wear out”

- Failure rates early in its life
- “Infant mortality”
- “Wear out”
- Hardware affects of dust, abuse, temp., & environment

Failure curve of hardware—“bathtub curve”
Software does “deteriorate”

![Failure curve of software](image)

Component based vs. Custom built

- Hardware products employ many standardized design components.
- Most software is always custom built.
- The software industry does seem to be moving (slowly) toward component-based construction.
Software Crisis?

- Projects running over-budget
- Projects running over-time
- Software was very inefficient
- Software was of low quality
- Software often did not meet requirements
- Projects were unmanageable and code difficult to maintain
- Software was never delivered

What is software engineering?

Pressman’s book
A discipline that encompasses
- process of software development
- methods for software analysis, design, construction, testing, and maintenance
- tools that support the process and the methods
Process, Methods, Tools

• Various tasks required to build and maintain software
  – e.g. design, testing, etc.
• SE process: the organization and management of these tasks
  – various process models
• SE methods: ways to perform the tasks
• SE tools: assist to perform the tasks
  – UML tools, IDEs, issue tracking tools

Background Check Survey

• Java experience 83%
• Eclipse experience 75%
• Eclipse plugin development 87.5% w/o
• Expectation:
  – 71% expect basic concepts and advanced topics in SE
• Undergraduate-level Software Engineering 66.7% (including relevant courses)