CS 5704
Software Engineering
Na Meng
Virginia Tech

Overview
• A bit about me
• Course goals
• Organization

About Me
• PhD—The University of Texas at Austin
• Research area
  – Software Engineering, Programming Languages

Research Goals
• To improve software quality and programmer productivity
  – Empirical studies: to understand characteristics of software or developers, e.g., bug patterns, code complexity, software maintenance activities …
  – Tools: to detect and fix bugs, to locate failure-inducing changes, and to automate program transformation …

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, program comprehension, automatic bug detection and fix
• Introduction of frequent techniques used in SE research/software development
  – program representation, and program differencing
Course Website

• http://courses.cs.vt.edu/cs5704/spring18/

Class Discussion

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

Introduction to Software Engineering

Overview

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

Software is ubiquitous

• System software
  – OS, compilers, device drivers
• Business software
  – Payroll, accounting
• Engineering/scientific software
  – Computer-aided design, simulation
• Embedded software
  – GPS navigation, Flight control, Toaster

Software is ubiquitous

• Product-line software (PC-like based)
  – Spreadsheets, word processing, games
• Web-based software
  – Gmail, Facebook, Youtube
• Artificial intelligence software
  – Robotics, artificial neural networks, theorem proving
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

- Hardware is difficult or impossible to modify
- Software is routinely modified and upgraded
- Hiring more people causes more work done
- This is not always true
- Costs are more concentrated on products
- Costs are more concentrated on design

Hardware does “wear out”

- Failure curve of hardware—“bathtub curve”

Software does “deteriorate”

- Failure curve of software

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) towards 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 was 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