CS 6704
Software Engineering Research

Na Meng
Virginia Tech

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

3

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

4

About You

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

5

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

6

Course Organization

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

7

Course Websites

- Course syllabus and schedule
  - http://courses.cs.vt.edu/cs6704/spring17/
- Grades
  - https://canvas.vt.edu/courses/42022

8

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!

9

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

10

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

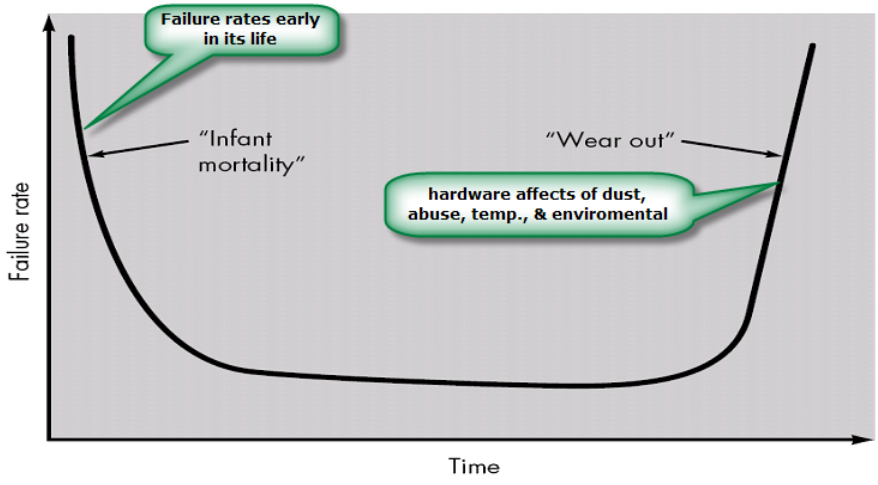
- Manufactured
- Wear out
- Built using components
- Relatively simple

- Developed/ engineered
- Deteriorate
- Custom built
- Complex

Manufacturing vs. Development

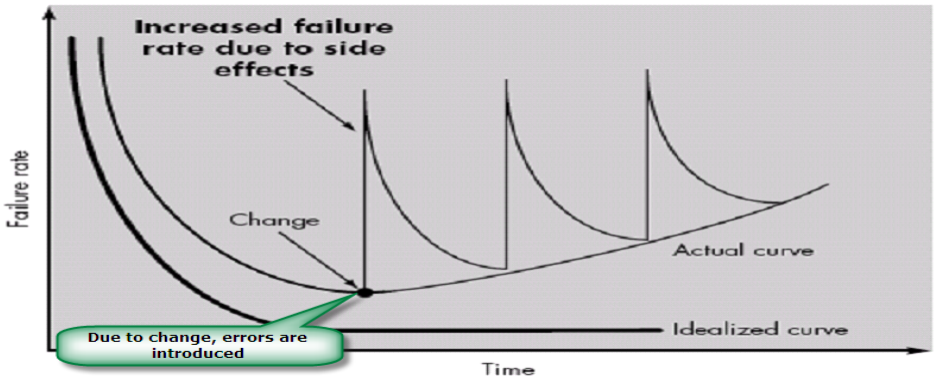
☐ Hardware is difficult or impossible to modify	☐ Software is routinely modified and upgraded
☐Hiring more people causes more work done	☐This is not always true
☐Costs are more concentrated on products	☐ Costs are more concentrated on design

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) 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