CS 5704
Software Engineering

Na Meng
Virginia Tech

Overview

- A bit about me
- A bit about you
- Course goals
- Organization

2

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, delta debugging, fault localization
- Introduction of frequent techniques used in SE research/software development
  - program differencing, clone detection, etc

7

Grading Policy

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

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

N. Meng, B. Ryder

15

Hardware vs. Software

☐ Manufactured   ☐ Developed/ engineered

☐ Wear out   ☐ Deteriorate

☐ Built using components   ☐ Custom built

☐ Relatively simple   ☐ Complex

N. Meng, B. Ryder

16

Manufacturing vs. Development

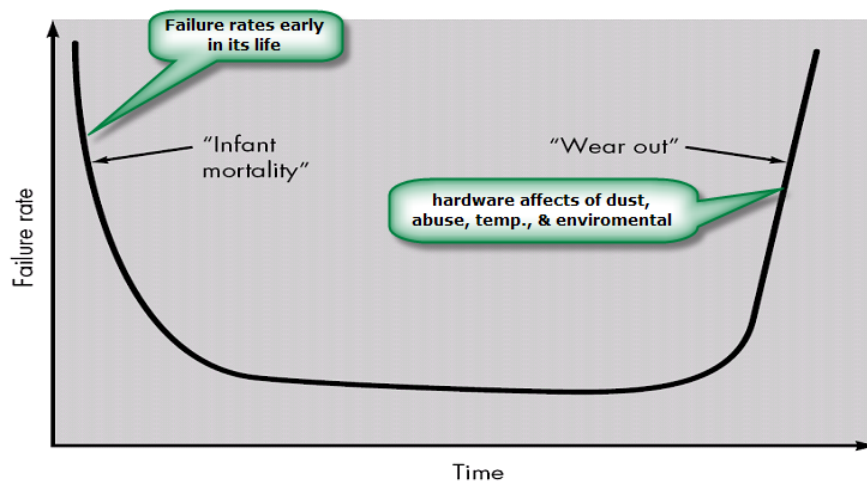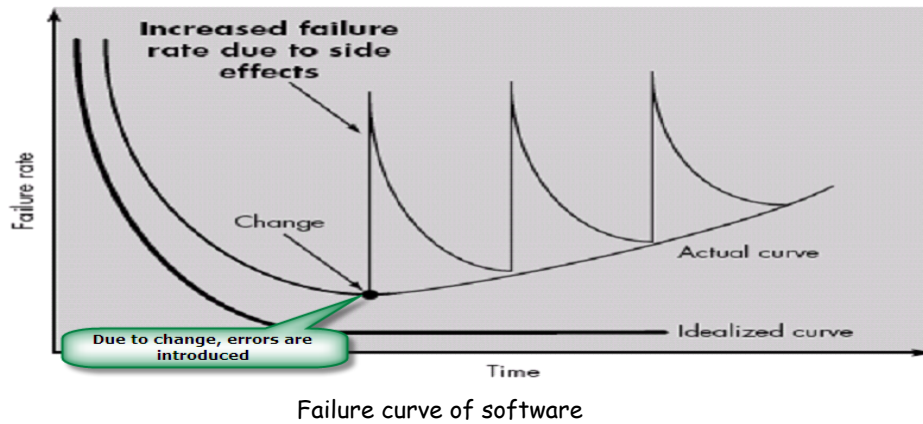|	
☐ Hardware is difficult or impossible to modify	☐ Software is routinely modified and upgraded
☐Hiring more people causes more work done	☐This is not always true
☐Costs are more concentrated on products	☐ Costs are more concentrated on design

N. Meng, B. Ryder 17

Hardware does "wear out"

Failure rates early in its life

"Infant mortality"

"Wear out"

hardware affects of dust, abuse, temp., & enviromental

Failure rate

Time

Failure curve of hardware—"bathtub curve"

N. Meng, B. Ryder

18

9

Software does "deteriorate"

Failure curve of software

N. Meng, B. Ryder

19

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.

N. Meng, B. Ryder

20

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

N. Meng, B. Ryder   21

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

N. Meng, B. Ryder   22

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

N. Meng, B. Ryder   23

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)

24