CMSC 838Z
Research in Software Engineering

Adam Porter

Goals

- Learn about software engineering (SE)
  - What is it?
  - Why do we care about it?
  - What specific research topics does it cover?
- Learn about conducting research in SE
  - What have others done?
  - What am I or should I be doing?
  - How do I show that what I'm doing improves on what other's have done?

Logistics

- Course meets
  - MW 3:30 – 4:45
- Counts for all comps
- This is an introductory course
  - No SE background is assumed/required

Ground Rules

- Will survey a variety of SE topics
  - One topic a week
  - One foundational paper, 2-3 current papers
- Reading assigned research papers is critical
  - Read the papers - BEFORE class
  - Submit written reviews - BEFORE class
  - Reviews will be graded
- Reading list is available at:
  - http://www.cs.umd.edu/class/spring2010/cmsc838z/

Course Organization

- In-class activities (15%)
  - Discussion
  - Lectures
  - Hands-on assignments
- Exams (50%)
  - Midterm (20%)
  - Final (30%)
- Individual research project (35%)
  - Background research
  - Develop, perform, evaluate research idea
  - Document and present results

1st Software Engineering Conference

- Software Engineering: Report of a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 Oct. 1968
- Some quotes:
  - David: "It has been said, too, that development costs for software equal the development costs for hardware in establishing a new machine line."
  - Kolence: "The basic problem is that certain classes of systems are placing demands on us which are beyond our capabilities and our theories and methods of design and production at this time."
  - Dijkstra: "The dissemination of knowledge is of obvious value — the massive dissemination of error-loaded software is frightening."

Historical Aside

- DeRemer & Kron paper was published in 1976
- What was the SE environment like back then?
  - Software begins to overtake hardware as cost center
  - Software market: few customers, custom products
    - Apple II (with 4K RAM) introduced in 1976
    - Microsoft founded in 1976
  - Poor development tools/computing environments
    - C, Unix, and Internet still research projects
  - Very-low level research topics:
    - Abstract Data Type, PL semantics, structured programming, proofs of correctness, flowcharts

Historical Aside

- Fred Brooks managed the IBM OS/360 project
  - Wrote "The Mythical Man-Month"
  - Won the Turing award for graphics research
- In 1987, reading about SE technologies was often like watching late night infomercials
  - Lose 30 pounds a day while eating all the pizza and drinking all the beer you want!!!!!!
- This paper says that no quick fix is likely, but some areas might have more payoff than others

Historical Aside

- 20 years ago
  - Software was distributed on tape through the mail
  - End-user machines were slow & had only 1 CPU
  - Networked applications were relatively uncommon
  - Virtual machine technology was immature
- How will the SE environment change in the next 20 years?

NY Times, 1/24/2010

- Radiation Offers New Cures, and Ways to Do Harm

"As Scott Jerome-Parks lay dying, he clung to this wish: that his fatal radiation overdose — which left him deaf, struggling to see, unable to swallow, burned, with his teeth falling out, with ulcers in his mouth and throat, nauseated, in severe pain and finally unable to breathe — be studied and talked about publicly so that others might not have to live his nightmare.

…

A New York City hospital treating him for tongue cancer had failed to detect a computer error that directed a linear accelerator to blast his brain stem and neck with errant beams of radiation. Not once, but on three consecutive days."

This Week's Papers

- No reviews required this week
- Start on next week's papers right away
- Today
  - How to Read an Engineering Research Paper, Bill Griswold
- Wednesday
  - F. DeRemer, and H.H. Kron. Programming-in-the-large Versus Programming-in-the-small. IEEE Transactions on Software Engineering, Vol. SE-2, No. 2, June 1976, pages 80-86.
  - F. Brooks. No Silver Bullet - Essence and Accidents of Software Engineering. IEEE Computer, April 1987

Typical Paper Format

- Introduction
  - Motivation
  - Outline of solution
- Body
  - Solution
  - Evaluation
- Wrap-up
  - Related work
  - Summary of results

Reading a Paper

- What problem are they solving?
  - People problem
  - Technical problem
- What's the proposed solution?
  - Hypothesis or idea
  - How well is it connected to original problem
- Does the idea work? Is the hypothesis true?
  - How strong was the evaluation?

Reading a Paper (cont.)

- How much of the stated problem was solved?
  - Sometimes authors exaggerate
- What are the next steps?
  - If you were the author's, what would you do next?
- How does this work relate to other work in the field?

Research Project

- One-page proposal for a small-scale software engineering research project due Feb. 3.
- Each student will present his/her research to the class
- Write-ups (like a workshop or conference paper) due on May11.

Research Project (cont.)

- But we haven't:
  - read much software engineering research yet
  - done any software engineering research before
  - thought about how to evaluate software engineering research
  - <insert more>

Choosing a Project

- Boehm: More software systems fail because they don't meet user needs than because they aren't implemented properly.
- Notkin: More software engineering research is uninteresting because the problem addressed is uninteresting rather than because the solution doesn't address the problem
- Interesting research problems often
  - Populate the world with a potentially powerful new approach
  - Make progress on an existing approach that is likely to have value
  - Carefully consider conventional wisdom
- At the same time, time is very limited

Ask Yourself

- "What is it about software engineering that sucks, but doesn't have you?"
  - Tedious, error-prone activities
  - Proposed techniques that are currently impractical
  - Difficulties caused by missing information
  - Unrecognized problems
  - Incorrect assumptions

Some of My Interests

- Configurable systems
- Mining software repositories
- Lightweight in-the-field inst. & analysis