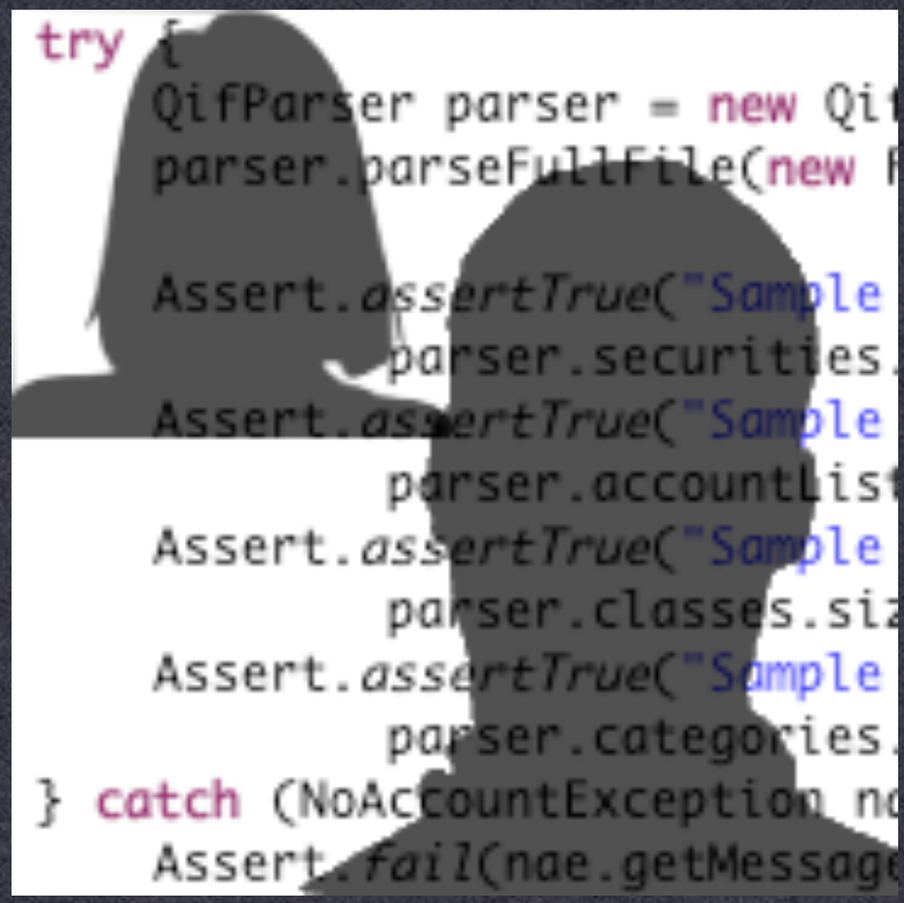
507

SE



CS 507: Advanced Software Engineering ‘Human Aspects of SE’

Reid Holmes

What is Programming?

"The process of transforming a mental plan of desired actions for a computer into a representation that can be understood by the computer"

-- Jean-Michel Hoc and Anh Nguyen-Xuan

What is Software Engineering?

> The establishment and application of scientific, economic, social, and practical knowledge in order to invent, design, build, maintain, research, and improve software that is reliable and works efficiently on real machines.
>
> — WIKIPEDIA MASHUP

Topic List

- Program comprehension
  - information needs, code navigation, working sets, code search
- Software evolution
  - refactoring, program differencing, reverse engineering
- Development tools & environments
  - team awareness, delta debugging, visualization, DOI models, task-centric development
- Quantitative and qualitative means of evaluating software engineering research

Topic List

- Program comprehension
  - information needs, code navigation, working sets, code search
- Software evolution
  - refactoring, program differencing, reverse engineering
- Development tools & environments
  - team awareness, delta debugging, visualization, DOI models, task-centric development
- Quantitative and qualitative means of evaluating software engineering research

Assessment

- In Class:
  - Paper presentation(s): 15% [1 class]
  - Paper reviews: 15% [Every class]
  - Class participation: 15% [Every class]
- Project:
  - Project reviews: 15% [April 4]
  - Project: 40%
    - Proposal [January 19]
    - Pitch [January 28]
    - Paper [March 21]
    - Talk [March 29]

Presentations (15%)

- 1 presentation (or 2, if class size is small)
  - ~30 minute talk
  - ~30 minute discussion
- Since this is a long talk slot it is fine to pull in content from other papers for context, to add a demo, or to look at a relevant video, if it can increase our understanding of the paper.
- I will go first (next week)

Presentations

- Select your top two papers and enter into Google Doc (link on course page).
  - Do not choose a paper someone else has already selected (motivation to choose soon!).
- Paper list provided on course page
  - If you want to present another SE paper, send me email and I will see if it is appropriate.
- Paper selections due Jan 12.
- I will organize the course schedule by grouping papers into the most appropriate order.

Paper Reviews (15%)

- Every lecture a short (300-500 word) paper summary / review will be submitted via email.
- The review should summarize:
  - The contributions of the paper.
  - The main positive points of the paper.
  - The main negative points of the paper.
  - What you learned from the paper.
- Grade directly related to the proportion of papers you review.
  - First review due on Thursday.

Class Participation (15%)

- Everyone should contribute to every paper we discuss.
- Grade directly related to the proportion of discussions you contribute to.
  - This is intended to be easy marks, given you will have already sent in a paper review.

Project Reviews (15%)

- Assess projects like a program committee
  - Everyone will read and review several papers
  - Reviews organized via Easychair
    - http://easychair.org
- Program committee meeting in the last class
  - Up to you whether we ‘accept’ papers
    - ‘acceptance’ has no bearing on your grade

Project (40%)

- Building software is hard. Your project will be to build a development tool that solves a problem you have encountered in your own experience.
- While you will be building a tool, a paper (6-8 pages) will be the primary artifact of the project.
- Groups are encouraged (up to 3 people)

Project Deliverables

- Project Proposal [Jan 19]
  - 1 page description, I will give feedback.
    - What problem are you trying to solve? How? How will you evaluate your tool?
- Project Pitch [Jan 26]
  - Quick 5 minute pitch to the class.
- Project Paper [Mar 21]
  - This is the main component of your project.
- Project Talk [Mar 29 / Mar 31]
  - 10 minute conference-style talk.

Software Tool

- Identify a real problem faced by developers
- Model a solution
- Implement the tool that addresses your model of the problem
- Evaluate (preliminary) the tool
  - Users would be great here, but given time constraints qualitative scenarios would work

To Do

- 1) Get an easychair account (free)
- 2) Choose 2 papers you would like to present
  - Insert into Google doc (by Sept 12 @ 0800)
- 3) Start thinking about projects

- I am happy to meet with you to talk about #2 and #3 at any time, if you have questions or want to talk about different ideas.

Thursday

- Fred Brooks Jr., No Silver Bullet. IEEE Computer, 1987.

Next Tuesday

- W. Wayt Gibbs, Software's Chronic Crisis. Scientific American, 1994.