Minor in Software Engineering

Jonathan Aldrich and Claire Le Goues

http://isri.cmu.edu/education/undergrad/

What is Software Engineering?

- Other CS courses
  - Core CS: Programming, Algorithms, Theory, …
  - Applications: Graphics, Robotics, OS, …

- Software Engineering
  - Technologies and approaches for making software development more effective
    - Cheaper, faster, less buggy, meet user needs
    - More fun!

Why should I care?

- Building software well is *hard!*
  - Customers can't tell you what they want
    - And you *will* guess wrong
  - When you're halfway through coding, they pull a change on you
    - Can your design accommodate the change?
  - Unpredictable problems come up
    - Did you leave time in your plan to fix them?
  - Millions of $, or human lives, may depend on your software
    - How can you gain enough confidence to sleep at night?

- There's a reason a typical job title is Software **Engineer**
  - In industry, you need to deal with all of the above
  - Programming ability alone is not enough

Cool Software Engineering Stuff

- Software architecture / design patterns
  - How do you design a million-line system?
  - Will it scale like Google and last for years?
- Software processes and teams
  - How do you organize hundreds of engineers?
  - Can you deliver the product on time and with few defects?
- Static analysis
  - Automated tools to verify your code won't crash or have security holes

So why should I care?

- Have more fun
  - Building buggy software, working 18-hour days, and delivering the wrong product late is frustrating
  - Software engineering is ***more rewarding when you do it right***
- Get a better job
  - Companies are looking for engineers, not programmers
  - The really cool jobs are team leads and software architects
    - Require management ability and/or design expertise
- Complement another field
  - Engineering – build systems with quality software components
  - Natural science – build high-quality scientific software
  - Business – manage teams that include software engineers

What you will learn

- Core computer science fundamentals
  - To the level of 15-214
- Building good software
  - Structuring of systems
    - design patterns, frameworks, architecture
  - Evaluation of systems
    - modeling, testing, analysis
- Organizing a software project
  - Teams and collaboration
  - Customers, users, and requirements
  - Process, estimation, management, and methods
- The larger context of software
  - Business, society, policy
- Engineering experience
  - Making engineering tradeoffs with limited information and resources
  - Working in teams
  - Following a process
- Communication skills
  - Written and oral

SE Minor Requirements

- Prerequisite:
  - 15-214 (Principles of Software Construction: Objects, Design, and Concurrency)
- Required core courses
  - 15-313 – Foundations of Software Engineering
  - 15-413 – Team project
- Electives – one from each of:
  - Technical elective
    - Domain-independent course focused on technical software engineering material
    - Examples: Requirements, Models, Management, Architecture, Analysis, Security
  - Engineering elective
    - Engineering focused course with significant software component
    - Examples: Wearables, Robotics, Web services, Embedded Systems
  - Business and Policy elective
    - Course that explores the business, social, legal, or policy context of computing
    - Examples: Privacy, Technology Law, Organizational Behavior, Technology Entrepreneurship
- Engineering internship requirement
  - At least 8 full-time weeks in an industrial setting, preferably 10-12 weeks
    - Must be integrated into team
    - May be in development, management, or quality assurance
    - Must be exposed to industry pressures
    - Must allow internal CMU practicum write-up
  - 17-413: Internship Reflection (required, 6 units, offered Fall)
    - Each student will write an issue-focused reflection and analysis of some personal software engineering experience, typically (but not always) based on the engineering internship above. This report must be passed by one SCS faculty member and one SE Ph.D. student, for both technical content and effective written communication.
    - 30-45 minute presentation, evaluated for communication skills and critical reflective content

Admission and Courses

- Application process
  - 15 accepted per graduating class
  - Next round of applications due Friday, Nov 7, 2014
  - Future applications due 1 week before course registration starts

- To apply:
  - email aldrich at cs *and* clegoues at cs
  - name, andrew ID, email, class, QPA, majors/minors
  - Why you want to be an SE minor
  - Proposed schedule of coursework

http://isri.cmu.edu/education/undergrad/

Questions?