CSC 307 Intro to Software Engineering

David Janzen
Why You Should *Not* Take This Course

• Most? computer scientists become software engineers
• The two course sequence 308/309 provides a much more thorough treatment of SE
  – SE concepts and practices are best learned through hands-on practice in a team project
Why You *Should* Take This Course

- You are sure you don’t plan to become a professional software engineer, you just want to learn a bit about the field
- You like a challenge
  - We will be packing two quarters into one
- You are graduating this year and it is 307 or no SE at all
- Dr. Janzen is cool and the project will be fun
Review Syllabus

- Syllabus
- Schedule on line
Define Software Engineering

• How would you define software engineering?

• IEEE definition:
  -- (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.
  -- (2) The study of approaches as in (1).
Does Practice Match Definition?

- Do you develop software in “a systematic, disciplined, quantifiable approach”?
- Do you expect software engineers to use such an approach if they are building:
  - Facebook
  - Google Earth
  - Avionics for commercial airliners
  - Laser surgery control software
Define Systematic

- characterized by order and planning
Define Disciplined

- obeying the rules
Define Quantifiable

• capable of being quantified
Define Development

• “A process in which something passes by degrees to a different stage (especially a more advanced or mature stage)”

• Construction
  – Requirements elicitation and analysis
  – Software architecture and design
  – Software construction
  – Software testing

• Process
• Management
• Tools and Methods
Define Operation

• “A process or series of acts especially of a practical or mechanical nature involved in a particular form of work”

• Build/Compile
• Installation
• Configuration
• Training
Define Maintenance

- “Care: activity involved in maintaining something in good working order”
- Defect tracking
- Defect fixes
- Adding features
- Refactoring software
Complete Survey
Introduce Course Project

- We will form teams of four or five students
- Each team will have a wiki
- Each team will develop the same project in parallel
- Project description at http://wiki.csc.calpoly.edu/brackets
Lab Activity 1: Cocktail Party

• People on left half of room will go to person in similar seat on right half of room
  – Introduce yourself (write each other’s name down)
  – Ask and answer the following:
    • Where did you go to high school?
    • What part of programming do you like the most/least?
    • What are your first and second preferences in project roles out of the following:
      – Team lead, requirements analyst, architect, developer, quality assurance, build/environment/tool master
  – When I raise my hand, rotate three people to the left and repeat
Lab Activity 2: Team Formation

• On a 3x5 card,
  – write your name
  – Write your first and second preferences in project roles out of the following:
    • Team lead, requirements analyst, architect, developer, quality assurance, build/environment/tool master
  – Write three others whom you think you would like to work with on the course project, and what role(s) you think they would do well
Lab Activity 3: Intro to Wiki

- Read about the Trac Wiki at http://trac.edgewall.org/wiki/TracWiki
- Try it out at http://trac.edgewall.org/wiki/SandBox