Please Pick Up a Handout
Either End of the Front Row

Software Engineering

COEN 174

Ron Danielson

Getting Started

Overview

- Administrative info
- Introduction (what the course is about)
- Overview of software engineering

Course Objectives

- Understand that the design and implementation of a successful software product requires making intelligent choices with a consistent rationale.
- Work with team members of diverse backgrounds and skill levels, and structure teams accordingly.
- Meet stated project requirements.
- Understand software engineering as a profession

Course Learning Outcomes

- Know the phases of the traditional software engineering development process, including the various levels of design and testing.
- Compare and contrast different software development processes, specifically heavy processes versus light processes, and choose the best process given a scenario.
- Specify, design, and/or implement a software product.
- Document and formally present artifacts of the software engineering process.
- Compare and contrast different software architectures, and choose the best architecture given a scenario.
- Define terms, constraints, and patterns of an object-oriented methodology.
- Compare and contrast different team structures.
- Know the IEEE/CS code of ethics for software engineers.

Unofficial Learning Outcome

- Dry-run the entire Senior Design Experience
  - Problem statement
  - Design document and review
  - Implementing a system
  - Presenting and documenting the system

Contact Information

- Professor
  - Ron Danielson
  - EC 324E
  - rdanielson@scu.edu; 554-6813
  - MW 10:45 am - noon, TR 12:30 pm – 1:30 pm, or by appointment
- TAs
  - Alex Adranly
    - aadranly@scu.edu
  - Ken Wakaba
    - kwakaba@scu.edu

About Ron

- Involved with computing for since 1968
- At SCU 42 years
  - 20 years as FT faculty
  - 22 years as FT administrator

Course Rules

- Professionalism
- Electronic devices
- Attendance
- Participation
- Integrity
  - It is ok to help other students
  - It is not ok to share code
- By submitting anything for grading, you certify it is solely your work, or that of your group for group projects

Grading

• Midterm exam:	20% (tentatively 10/25)
• Final exam:	35% (12/5)
• Project – technical	35%
• Project – contribution	10%

Project Grading

• Problem statement:	10% (week of 9/24)
• Design document:	20% (week of 10/8)
• **Design review**	10% (week of 10/15)
• Initial operational system	20% (week of 10/29)
• **Final presentation**	10% (week of 11/12)
• Final report	15% (week of 11/26)
• Final system	15% (week of 11/26)

Presentations

- All team members talk
- School of Engineering slide template
  - Suggestions about slides coming later
- Appropriate dress
  - Business casual for design review
  - Business formal for final presentation

Problem Statement

- Emphasis on problem and solution
  - Description of problem
  - Scenario(s) of use
  - Solution
- Possible contents
  - Motivation
  - Background including stakeholders
  - Business case
  - User Scenario(s)
  - Evaluation criteria

Determining the Problem

Book Matching System
Selecting a book to read for a course depends on a number of factors (reading level, issues addressed, setting, characteristics of main characters, genre, …). This system would allow professors working with K-12 students to enter factors and possible values for those factors for each of a number of books. A student could then rank their interests on each of the factors, and a weight for each factor, and the system would present a set of possible matches based on the student's preference.