Course Overview

Software Engineering

Stephen M. Thebaut, Ph.D.
University of Florida
Office Hours & Contact Info

- **Instructor:** Steve Thebaut
  - Office Hours: by appointment
  - e-mail: smt@cise.ufl.edu
Description

- A graduate-level survey of the fundamental concepts and principles underlying current and emerging methods, tools, and techniques for the cost-effective engineering of high-quality software systems.

- NOT a “programming” course.

- Focuses on surveying critical aspects of SE that may be less familiar to students of computer science. For example…
identifying a development process appropriate to the circumstances,
— eliciting and documenting requirements,
— indentifying appropriate design techniques,
— employing effective verification and validation strategies (including formal reviews/inspections) throughout the software development lifecycle,
— software maintenance, and
— software project management.
Prerequisites

- Familiarity with programming using a high-level language (C, C++, Java, etc.)
- Basic knowledge of algorithms, data structures (linear lists, etc.), and discrete math (symbolic logic)
Class Meeting Schedule

- Week 1: May 4-6 (Tues, Wed, Thurs)
- Week 2: May 10-12 (Mon, Tues, Wed)

------ Nineteen-day break period ------

- Week 3: June 1-3 (Tues, Wed, Thurs)
- Week 4: June 7-9 (Mon, Tues, Wed)

Class meets from 6-9 PM
Web Site

Visit the course website at:

www.cise.ufl.edu/class/cen5035/se.html

- Syllabus
- Lecture Notes
- Announcements
- Optional Exercises
- Reading assignments
Textbook and Outside Readings

- **SOFTWARE ENGINEERING, 8th ed.,** by Ian Sommerville, Addison-Wesley, 2007.
  - See “Readings” at website for assigned parts of Chapters 18, 19, 21, 22, 28, 31, and 32.

- Suggested *supplemental readings* are also listed on the course website, but these will NOT be the basis for exam questions.
Outline of Topics

- Introduction to SE and FAQs
- Software Processes
- Project Management
- Software Requirements
- Requirements Engineering Processes
- Prototyping/Rapid Development
- Formal Specification
- Architectural Design
- Distributed & Service-Oriented Systems
Outline of Topics (cont’d)

- Object- and Aspect-Oriented Design
- Software Reuse
- Verification and Validation
- Proofs of Correctness
- Software Testing
- Software Evolution
- Process Improvement
Examinations, Project, and Grades

- Course grades will be based **SOLELY** on an individual term project (20% of final grade), an open-book/open-notes take-home exam (20% of final grade), and a 3-hour comprehensive final exam (60% of final grade) – date TBD.

- The course grading scale used will be as prescribed by UWI:

  A: 70-100%
  B+: 60-69%
  B: 50-59%
  Failing: 0-49%
Take-Home Exam

- Will reflect the types of questions and problems you can expect on the final exam.
- Primarily intended to facilitate collaborative learning as opposed to individual assessment, so students are encouraged to work together in small groups.
- **Completed exams are due at the beginning of class on Tuesday, June 8.** *(NOTE: It is *strongly* recommended that students attend class on the due date...)*
Term Project *Musts*

- Must represent a significant (8-10 hours *plus* report writing time), *individual*, creative effort.
- Must be directly related to the course, tailored to your personal and/or professional interests.
- Topic must be approved in advance. Discuss your idea informally with me before submitting a written project proposal (up to 500 words) **via e-mail**.
- Proposals must be received by Monday, May 17.
- Final project reports (4-6 typed pages) are due at the beginning of class on Thursday, June 3.
Term Project Sample Topics

● Conduct in-depth research/study based on outside sources (articles, books, etc.) of a specific SE topic of interest to you that is NOT covered by the course in depth.

● Analyze how some specific SE process or development approach introduced in the course could be adapted or applied to a problem or situation of concern or interest to you. (The adaptation required should be non-trivial.)
Term Project Sample Topics (cont’d)

- Conduct research in your workplace or a local organization related to some current SE practice with the aim of identifying strengths, weaknesses, and feasible paths to process improvement.

- Read and provide an in-depth, critical review of some recent SE-related book or scholarly paper of interest to you. (E.g., Fred Brooks’ new book, The Design of Design.)
Term Project Sample Topics (cont’d)

- Consider the non-trivial research- and/or analysis-related exercises at the end of each Sommerville chapter. (They are usually among the latter exercises given in each set.)
Term project proposal Info

● Describe the proposed topic *in detail*.
  — Is the topic also covered in the course? If so, in what chapters and/or lecturers?
  — How will your work compliment or add to what is covered in the class?

● Describe the *PURPOSE* of your project.
  — Are there specific questions or issues that you will address?
  — How will your effort or results be of benefit to yourself or others?

● Explain *WHY* you wish to work on this particular topic.
  — Why is it of interest to you?
  — Have you studies or worked on this or a related topic before -- perhaps in a different class or at work? If so, explain.
Optional Homework Exercises

- Three optional problem sets (together with solution notes) related to Formal Specification, Software Testing, and Proofs of Correctness are available on the course website.
Questions?
Course Overview

Software Engineering

Stephen M. Thebaut, Ph.D.
University of Florida