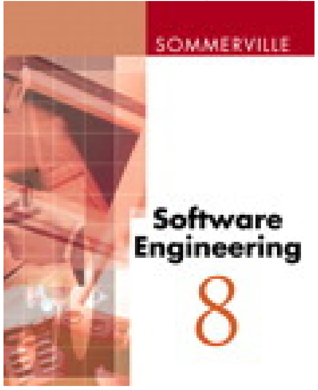
CS487 - Software Engineering Last Updated - 04/11/02
Course Manager – Dr. Ilene Burnstein, Associate Professor
3 credit hours; required for CS & CPE; 150 min. lecture each week

Current Catalog Description - Study of the principles and practices of software engineering. Topics include software quality concepts, process models, software requirements analysis, design methodologies, software testing, and software maintenance. Hands-on experience building a software system using the waterfall life cycle model. Students working in teams develop all life cycle deliverables: requirements document, specification and design documents, system code, test plan, and user manuals. Prerequisite: CS 331 or CS 401 or CS 403. (3-0-3) (C)

Textbook

- Software Engineering: (Update), 8/E Ian Sommerville, University of St. Andrews, United Kingdom ISBN-10: 0321313798 ISBN-13: 9780321313799 Publisher: Addison-Wesley. Copyright: 2006



References - other textbooks or materials

- none

Course Goals - Students should be able to:

- Understand and explain software development as a series of engineering activities, and processes.
- Demonstrate software development team-working skills.
- Analyze client/user needs.
- Select an appropriate life cycle and process model for development of a software product.
- Explain the importance of software quality evaluation activities.
- Develop a series of software life-cycle deliverables.
- Develop representations/models and descriptions of an evolving software product for inclusion in a requirements specification document.
- Build a multi-level design model and evaluate software design alternatives
- Design, execute, and log multi-level software tests.

- Describe the role that tools can play in the software life cycle.
- Communicate, verbally and in writing, the deliverables of a software development project.

Prerequisites by Topic

- Experience in developing basic programs in any computer language
- Have an understanding of, and be able to apply, the essential data structures and algorithms used in computer science.

Major Topics Covered in Course

1. The problem statement, developer-client interactions. Overview of software engineering – life cycle models, software deliverables. 3 hours
2. Software development team concepts, team organization, team structures. Project management, the project plan. 3 hours
3. Requirements analysis, methods, models. For example, structured analysis with use of data flow diagrams, data dictionary, entity-relationship diagrams. 7 hours
4. Software specification, methods, and models. For example, structured analysis with use of process specifications, state transition diagrams. 3.5 hours
5. Preliminary design concepts, methods, and models. For example, structured analysis with use of structure charts, procedural abstractions. Concepts of top down decomposition, bottom-up composition, abstraction, coupling, cohesion, modularity, information hiding, reuse, architectural styles. 6.5 hours
6. Detailed design concepts, methods and models. For example, structured analysis with use of PDL 2.5 hours
7. (Program Design Language. Algorithm, and data structure design. Object concepts. Object-oriented analysis, nature of the approach, models. For example, Coad/Yourdon analysis model with use of class diagrams, class hierarchies, attribute, and service specifications. Role of use cases. Use of modeling languages such as UML. Object-oriented design approaches, for example Coad/Yourdon's 4-layer object-oriented design model. 4.5 hours
8. Software implementation, transition from design to code. 1 hour
9. Software testing and evaluation. Black and white box test design strategies and related techniques, testing at multiple levels, regression test. 6.5 hours
10. Software quality, reviews, and metrics. 3 hours
11. Software maintenance and re-engineering. Types of maintenance, role of configuration management, legacy code, tool support for maintenance. 1.5 hours
12. Selected Topics 1.5 hours
13. Midterm Exam 1.5 hours
14. Final Exam – 2 hours
15. 45 hours

Laboratory projects (specify number of weeks on each)

- A software development project of sufficient size and scope is assigned requiring the preparation of the following deliverables submitted as a team (3-4 person) effort:
  - Requirements/specification document
  - Design document
  - Code, Test plan, Test logs
  - Demonstration of Project - Acceptance Test.
  - The project requires a team effort of approximately 175 hours spread over a 13 week period

Estimate CSAB Category Content in Credit Hours
CORE ADVANCED CORE ADVANCED
Data Structures Computer Organization and Architecture
Algorithms .5 Concepts of Programming Languages .5
Software Design 2

Oral and Written Communications - Every student is required to submit at least __3___ written reports (not including exams, tests, quizzes, or commented programs) of typically __20-30___ pages and to make __1___ oral presentations of typically __10___ minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

- Requirements specification document
- Design document
- Test plan, and log
- Oral project demonstration/acceptance test

Social and Ethical Issues - Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)?

- Types of user communities - impact on user interface design, 0.4 hours, design document
- Distribution of Software Engineering code of ethics, 0.1 hour, no grading
- Social responsibilities of software engineers, licensing and certification, 0.1 hours, no grading

Theoretical Foundations - Please list the types of theoretical material covered, and estimate the time devoted to such coverage in contact (lecture and lab) hours.

- Software Architecture - theoretical aspects, 1 hour
- Testing theory, 1 hour
- Object orientation, 1 hour
- Decomposition/ Abstraction, 1.5 hours

Problem Analysis - Please describe the problem analysis experiences common to all course sections.

- Entire course is devoted to the problems involved in developing a medium- sized software system. This includes:
  - analysis of clients needs
  - requirements/specification analysis modeling
  - design analysis
  - analysis for test design

Solution Design - Please describe the design experiences common to all course sections.

- A good fraction of course time is spent in design. Preparation of design documents requires students to develop an overall architectural design, and a detailed design for their project. Graphical notations are used to represent the architecture; a pseudo code-like model is used to represent the detailed structure of the individual modules.
- Students spend time in test design using both white and black box design strategies. They also design tests at several levels of abstraction; for example unit, integration, system and acceptance tests.

Other Course Information

- Additional Suggested Course Assignments

o none

- Planned Course Enhancements

o none