
COURSE SYLLABUS



CEN 4020 **Software Engineering I**

Fall 2007

Prerequisites (COP 4530) Strickly Enforced

COURSE POLICIES:

Assignments/Responsibilities:

Exams 40%
Final Exam 5%
Software Engineering Homework Assignments 10%
Quizzes 10%
Software Engineering Project Assignments 35%
 PROTOTYPE - 4% working prototype and 1% presentation
 SRS - 4% final SRS document and 1% presentation
 Other deliverables - 25%

Grading/Evaluation:

94-100 A	87-89 B+	77-79 C+	67-69 D+	50-59-F+
90-93 A-	84-86 B	74-76 C	64-66 D	0-49 - F
	80-83 B-	70-73 C-	60-63 D-	

Attendance:

For on-campus students, attendance is expected for each class meeting. Distance students are required to participate in alternative Internet-based modes of instruction, including reading the on-line lecture notes, asking questions of the instructors via e-mail, and making regular use of the course Discussion Board (if available) to discuss the topics posted by the instructor.

Make-Up Tests:

All homework assignments missed with university approved excuses must be made up within one week or the original assignment. Missed exam with university approved excuses will be made up by substitution of a comprehensive make up exam at the end of the semester.

Projects:

This course is a project intensive course. It allows students to have two distinct experiences with projects. The first experience is for a homework project. Students individually read written specifications, and interpret these specifications to produce individual homework assignments which assure each student has knowledge of the Unified Modeling Language for performing software specification and code generation. The second experience is to work on a software development team using the Unified Process and UML to produce IEEE Software Requirements Specifications. Exams test ability to produce project deliverables.

NOTE: “Except for changes that substantially affect implementation of the evaluation (grading) statement, this syllabus is a guide for the course and is subject to change with advance notice.”

COURSE MATERIALS:

Required Textbook: Systems Analysis and Design with UML Version 2.0

Author: Alan Dennis, Barbar Wixom, David Tegarden

Publisher: John Wiley and Sons, Inc

ISBN-10: 0471348066

ISBN-13: 9780471348061

Unified Process Reference:

[Agile Unified Process](#) (AUP), a lightweight variation by Scott W. Ambler

[Basic Unified Process](#) (BUP), a lightweight variation developed by [IBM](#)

[Enterprise Unified Process](#) (EUP), an extension of the RUP

Essential Unified Process (EssUP), a lightweight variation by [Ivar Jacobson](#)

[Open Unified Process](#) (OpenUP), the Eclipse Process Framework

[Rational Unified Process](#) (RUP), the [IBM](#) / [Rational Software](#)

Rational Unified Process-System Engineering (RUP-SE), a version of RUP

Unified Modeling Language Reference: <http://www.uml.org/>

COURSE RATIONAL:

This course serves as the first course in a two semester project series for students who have completed all elementary programming courses. It introduces students to software engineering requirements engineering techniques and specification documents according to IEEE standards.

COURSE DESCRIPTION:

This course is the first semester of a two semester software engineering sequence integrating theory and practice with a project experience. Topics include theory, tools, requirements elicitation, software requirements, specifications, requirements review, software development, ethics, software development life cycle terms and project management.

COURSE OBJECTIVES:

A student who has completed this course with a passing grade should be able to:

Be able to recognize and use various life cycle models

Define and use the concepts of the software development processes

Define and work using a team structure.

Define, implement, and use a software development project plan.

Investigate and define ethical issues in software development

Recognize and use requirements engineering processes

Define requirements using RUP workflows processes

Specify requirements using the Unified Modeling Language

Build and review supporting data dictionaries for UML

Elicit requirements using defined elicitation techniques

Define functional and non-functional requirements

Build and review a data model

Use review techniques for walkthroughs, formal reviews and audits

TOPICS:

- Week 1: Introduction to software engineering, Software engineering process models, Inputs and Functionality Specifications
- Week 2: The Unified Process, Workflows and phases, Introduction to UML, Use Case Diagrams. Ethical and social implications of designing use cases and business process reengineering.

- Week 3: Intro to Requirements workflow, requirements elicitation and elicitation techniques, functional and nonfunctional requirements, shall lists. IEEE Software Requirements Specification Document, Social implications of testable requirements and the IEEE SRS.
- Week 4: Analysis workflow, data dictionary definitions, requirements analysis, Behavioral Specifications.
- Week 5: Object-oriented specifications, Structural Specifications, Data dictionary for structural specifications, ethical and social implications of good specifications.
- Week 6: Requirements and analysis workflows revisited with inputs, output, major activities, techniques and tools, UML Diagrams created, CASE tools. Ethical and social implications of the use of CASE in the work environment.
- Week 7: Team organization, team roles, communication skills needed including both written and oral, project management documentation, social implications of team building.
- Week 8: Spring Break
- Week 9: Planning phase of software development, project initiation, cost benefit analysis, risk analysis, feasibility analysis, ethical considerations of truth in feasibility analysis.
- Week 10: Project management, PERT, Gantt, project plans, change manage, configuration management, requirements scope management, ethical considerations in scope management and truth in project plans, presentation of ethical case study.
- Week 11: Testing workflow, verification and validation, functional testing, unit testing, system testing, integration testing, testing approaches. Ethical and social considerations for user testing and acceptance testing plans.
- Week 12: Moving to design – an introduction to design workflow, design considerations.
- Week 13: User Interface design, prototyping, screen shots, reports, user interface navigation matrix
- Week 14: Team Presentations and Review, discussion of ethical considerations is software specification presentations to users.

ACADEMIC HONOR CODE:

Students are expected to uphold the Academic Honor Code published in The Florida State University Bulletin and the Student Handbook. The Academic Honor System of The Florida State University is based on the premise that each student has the responsibility (1) to uphold the highest standards of academic integrity in the student's own work, (2) to refuse to tolerate violations of academic integrity in the university community, and (3) to foster a high sense of integrity and social responsibility on the part of the university community.

Please see the following web site for a complete explanation of the Academic Honor Code.

<http://www.fsu.edu/Books/Student-Handbook/codes/honor.html>

<http://www.fsu.edu/Books/Student-Handbook/>

AMERICANS WITH DISABILITIES ACT:

Students with disabilities needing academic accommodation should: (1) register with and provide documentation to the Student Disability Resource Center; (2) bring a letter to the instructor indicating the need for accommodation and what type. This should be done during the first week of class.

For more information about services available to FSU students with disabilities, contact:

Student Disability Resource Center
 Dean of Students Department
 08 Kellum Hall
 Florida State University
 Tallahassee, FL 32306-4400
 (850) 644-9566 (voice)
 (850) 644-8504 (TDD)

SDRC@admin.fsu.edu

<http://www.fsu.edu/~staffair/dean/StudentDisability/>