

ECE 435

Software Engineering

Overview

Course Catalog Description: Management and technical issues including business conduct and ethics relating to the design of large engineering projects. Student teams will address design, specification, implementation, testing, and documentation of a large hardware/software project.

Prerequisites: C- or better in ECE 331, ECE 335

Textbook: S. Lawrence, J. Atlee, *Software Engineering: Theory and Practice (3th Edition)*, Prentice-Hall Publishers, 2006.

Class Goals: To build on knowledge students have gained from previous programming and software courses so that they may learn how to apply more sophisticated techniques to the design and lifecycle of largescale software systems. To allow students to understand and/or conduct requirements analysis, system design, program design, project estimation/tracking, testing, evaluation, and maintenance. To provide a detailed understanding of different team-based management and organizational practices for large software projects.

Course Coordinator: Prof. Nasir Ghani

Table I: Objectives, Implementation, and Assessment

Objectives		Implementation	Assessment	A	B	C	D	E	F	G	H	I	J	K
O ₁	Understand basic overview of software development cycle process	9 hrs. lecture in weeks 1-3	Hw 1	X		X		X						
O ₂	Provide exposure to modern project planning, tracking, and estimation techniques	3 hrs. lecture in weeks 3 and 4	Hw 2, Exam I	X		X		X						
O ₃	Understand the requirements generation, system design, and program design process	12 hrs lectures in weeks 5-10, project assignments	Hw 3, Project, Exam II	X	X	X	X	X		X				X
O ₄	Understand testing, delivery, maintenance, and project evaluation processes	15 hrs lectures in weeks 10-16, project assignments	Hw 4, 5 Project, Exam III	X	X	X	X	X		X				X
O ₅	Understand the application of object-oriented (OO) design and programming concepts to solve real-world problems.	4.5 hrs. lecture in weeks 7-9	Hw 3, Project, Exam II			X								X
O ₆	Understand Uniform Modeling Language (UML) notations and their applicability to real-world problems	4.5 hrs lecture in weeks 6 and 8	Hw 5, Project, Exam II			X								X
O ₇	Learn to work in a high-performance team environment, improve personnel and communications skills sets	2 hrs guidance on project planning in weeks 7 and 11	Project				X		X	X				

Table II: Expectation and Assessment Outcome
Fall 2008, Dr. N. Ghani (TBD)

General expectations:

Homework Assignments:

These assignments involve two components:

1. Sharpen/practice/improve skills and understanding of software engineering concepts taught in class
2. Develop requisite skills to conduct large project design (e.g., making sure they are able to formulate requirements, specify design, conduct program development and testing, maintenance, etc).

For (1), expect that all of the students will be able to answer at least 70% of the problems assigned. For (2), expect 100% of the students to be able to complete the assignments, otherwise then cannot have success in the course project.

Class Project:

Project involves complete design of a software product from inception to delivery, i.e., full life-cycle development. Students are required to work in teams of 4-5 each to define a challenging project (approved by instructor), conduct necessary background reading and survey, full requirements elicitation, system design, program design and coding, testing, and evaluation. The teamwork aspect will emphasize improvement in inter-personal and communications skill sets. A series of design and final reports will be submitted along with an in-class presentation.

Exams: Expect 75% of the students to score 70% or better on all exams.

Assessment to be completed by Dr. Ghani for Spring 2009 class

ABET Assessment for Outcome D: Ability to function on multi-disciplinary teams

This outcome will be assessed using the class project. Specifically, the students will be given a detailed survey questionnaire after completion of the in-class project presentations. A sample list of possible queries includes the following:

- Did the project improve your overall teamwork and team-building skills?
- Did the project improve your ability to organize project tasks and see them to completion?
- Did the project provide you an opportunity to work with people with varying multi-disciplinary skills sets?
- What were some of the challenges encountered in the team environment?

Sample Course Schedule

Week	Date	Lect.	Topic	Assignment
1	26 Aug	1	Introduction & Course Administration	
	28 Aug	2	Overview of software engineering, systems approach	Ch. 1
2	2 Sep	3	SW eng evolutions, Wasserman's approach, examples	Ch. 1
	4 Sep	4	Modeling process/lifecycle, models (Waterfall, V models)	Ch. 2
3	9 Sep	5	Models (prototyping, phased, agile methods), Lai notations	Ch. 2
	11 Sep	6	Project planning/measurement, schedules/activity graphs	Ch. 3
4	16 Sep	7	Critical path analysis, tracking, personnel/communication	Ch. 3
	18 Sep	8	Effort estimation (Wolverton, expert model, COCOMO)	Ch. 3
5	23 Sep	9	Risk management, accountability, examples	Ch. 3
	25 Sep	10	Requirements analysis, elicitation techniques, testability	Ch. 4
6	30 Sep	11	ER diagrams, UML, state machines, Petri nets, functional	Ch. 4
	2 Oct	12	SDL overview, prototyping, validation/verification	Ch. 4
7	7 Oct	13	System design, conceptual/technical, OO design, layering	Ch. 5
	9 Oct	14	Re-use, coupling/cohesion, complexity, project overview	Ch. 5
8	14 Oct	–	Exam I	
	16 Oct	15	Objects, classes/hierarchies, uses cases, UML notations	Ch. 6
9	21 Oct	16	State diagrams, message sequence charts, OO measurement	Ch. 6
	23 Oct	–	<i>Fall Break</i>	
10	28 Oct	17	Program design, coding/documentation, pair programming	Ch. 7
	30 Oct	18	Program testing, fault types, test objects, unit testing	Ch. 8
11	4 Nov	19	Integration testing, test planning, examples, project issues	Ch. 8
	6 Nov	20	System testing, regression testing, cause-effect graphs	Ch. 9
12	11 Nov	–	Exam II	
	13 Nov	21	Reliability models, acceptance tests, test documentation,	Ch. 9
13	18 Nov	22	Testing safety-critical systems, testing tools	Ch. 9
	20 Nov	23	System delivery, training techniques, trouble-shooting	Ch. 10
14	25 Nov	24	Systems maintenance, Lehman/S-/P-/E-systems	Ch. 11
	27 Nov	–	<i>Thanksgiving Holiday</i>	
15	2 Dec	25	Maintenance models, rejuvenation, reverse/re-engineering	Ch. 11
	4 Dec	26	Project evaluation, case studies, prediction techniques	Ch. 12
16	9 Dec	27	Quality models, re-use, capability maturity model	Ch. 12
	11 Dec	28	Future of software engineering, course conclusions	Ch. 13
17	15 Dec	–	Exam III	