Software Engineering Laboratory

Bulletin Description

Organization and scheduling of software engineering projects, structured programming, and design. Each team designs, codes, and debugs program components and synthesizes them into a tested, documented program product.

General Course Info

Term:	TERM FALL 2013
Department:	COMP
Course Number:	523
Section Number:	001
Time:	MW, 1:00-2:15 In addition, there will be weekly team meetings with the professor and the client
Location:	SN 011
Website:	http://wwwx.cs.unc.edu/Courses/comp523-f13

Instructor Info

Name:	Diane Pozefsky
Office:	FB 146
Email:	pozefsky@cs.unc.edu
Phone:	919 590-6117
Web:	http://www.cs.unc.edu/~pozefsky
Office Hours:	Open Door Policy

Textbooks and Resources

There are no required or recommended textbooks. Sakai will be used primarily for the returning of comments and grades on individual assignments. Individual assignments will be submitted through Sakai and team assignments will be submitted through the team's website. Server software resources will be available as needed. All recommended or required readings will be available or referenced on the class website.

Course Description

The goal of this course is to teach the skills necessary for building a software product as a team. The lecture portion of the class will cover the broader picture of software engineering that includes a wide range of software development projects in terms of size, complexity, and criticality. The course carries EE (experiential education) and C (communications intensive) tags and is an APPLES course.

Target Audience

This course is intended for upper class majors with an interest in building software for practical use. Students are expected to have enough experience to be able to learn new software systems on their own and to design a system using techniques and principles learned in other courses. This is an ideal course for those interested in getting real world experience in building software and communicating with others.

Prerequisites

COMP 410 and 411 plus two additional COMP courses numbered 426 or higher. The additional two courses will ideally cover software tools, techniques or principles.

Goals and Key Learning Objectives

At the end of the course, each student will have experienced all aspects of a software development project, including

- working with a client to define goals and priorities
- designing a system
- scheduling and planning a multi-person project
- effective communications
- running meetings
- writing technical documentation
- preparing web content
- writing and testing code
- deploying the system
- public presentations

Disclaimer

The professor reserves to right to make changes to the syllabus, including assignment and project due dates. These changes will be announced as early as possible and will be reflected on the course website. If there are discrepancies between this syllabus and the website, the website is considered the definitive information.

Course Requirements

The essence of the course is the faculty-coached team project. Teams of 2-4 students spend the semester negotiating, estimating, scheduling, specifying, coding, debugging, integrating, documenting and testing a substantial programming product. Each project has a real client that is expecting a completed project. Each document will be submitted to the professor in draft form and will be revised based on comments. In addition, documentation needs to be maintained to reflect changes in the product that is being produced.

There will be no written exams; there will be individual assignments given to cover the key concepts of the course that are not well reinforced through the project and to expose you to the literature in the field.

There are a lot of new things happening in the field of software engineering that you are not exposed to within the department's curriculum. Each team will present a technology from a provided list or one that they are using that has not been taught in other classes. Teams will give a 30-45 minute presentation to the class.

Project grades are based on code, documentation, ambition, effort, teamwork, and accomplishment.

The final exam is a presentation of the end product.

Key Dates

Project demos will be the weeks of September 30 and November 4. Essays will be due October 21 and November 20. Key deliverables for the projects are the functional specification, due September 16 and the manuals and design documentation, due December 9. This is also the day that the final code is due and the final presentation given.

Grading Criteria

Overall breakdown

Project	75%
Technology talk	15%
Essays	10%

Project

I compute a single grade for the project, based on the following percentages:

Process	25%
Code	25%
Documentation	20%
Final Presentation	5%

I then apply an individual contribution multiplier for each person. This value is based on my observations as well as the evaluations by your client, any consultants, and your peers. The multiplier ranges from .7 to 1.1, but a value above 1 is only used in exceptional cases. Basically, I do not believe that you should be able to get a better grade than the product you produced.

A few more details:

- Process includes whether you are interacting appropriately as a team, with me, and with your client. Are you addressing issues as they arise? It includes professionalism in your dealings with your client and your professor and whether you are usually on time with deliverables or habitually late. It includes your web site and the materials that you produce as steps to produce the other artifacts. I will be giving you 3 process grades: for the requirements phase (7%), the design and implementation phase (8%), and the completion (10%). The reason for that is to recognize that there are different processes and issues during different phases of the project.

- The code grade covers function, correctness and readability. The three components are equally important. Have you met the primary requirements? How many bugs was I able to find? I will do a random review of the code that you produce. I expect to be able to understand it. This includes web pages as well as other code that you write.
- The documentation covers the formal deliverables: the functional spec, the design document, the user manuals, and the test plan. They are weighted equally. Remember that spelling, grammar, and readability are important; unreadable good content is not sufficient.
- For the final presentation, your grade will include both content and style.

Technology Talk

Your grade will include both content and style. I will be looking to see if you understand what you are presenting and whether you are communicating well with your classmates.

Individual Assignments

There will be two essays assigned. The intent is to introduce you to the classic literature of the field. Grading will be based on the demonstration of understanding the content, reasonable proofing, and following instructions.

Course Policies

Attendance: While attendance is not taken in class, I expect student attendance at all peer presentations. Specifically, I expect you to be at all technology talk and demo presentations. For other classes, I only point out that there is no textbook in this class because the content is not available in any simple form. If you are interested in the content, you need to listen to lectures. We also have a number of outstanding not-to-be-missed outside speakers during the class.

The course final is given in compliance with UNC final exam regulations and according to the UNC Final Exam calendar.

Honor Code

Collaboration and peer-learning are necessary for team projects. Only the individual assignments are not to be done collaboratively. These are open book, open notes, and open network. My goal is to give you essays that require individual thought and reflection and the work must be that of the student. Directly taking text from other sources is not acceptable. Short excerpts from other sources may be quoted and properly cited.

Course Schedule

The following is a draft of the class schedule.

	Aug 23 Introduction
26 Client Presentations Preferences due at 9 pm	30 Client Meetings
Sept 2 Labor Day (no class) Web site due	4 Requirements Phase
9 Scheduling and Project Mgmt User stories & personas due	11 Software Development Processes Platform selection due
16 Version Control Functional spec & Tech talk topic due	18 Software Architectures and Design Initial schedule due
23 Accessibility Technology spike due this week	25 Multi-platform Web Development Architecture diagram due
30 Demos	Oct 2 Demos Requirements phase ends
7 User Interfaces	9 Agile Programming
14 Tech Talks	16 Tech Talks
21Security Essay 1 due	23 Security
28 Tech Talks	30 Tech Talks
Nov 4 Demos	6 Demos Design & implementation phase ends
11 Tech Talks	13 Tech Talks
18 Testing	20 Testing Essay 2 due
25 Mythical Man-Month First drafts of design doc & manual due	27 Thanksgiving (no class)
Dec 2 Privacy and Security	4 Ethics and IP
9 final presentation, all deliverables due Completion phase ends	

Last updated 10/6/2013