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:	FALL 2016
Department:	COMP
Course Number:	523
Section Number:	001
Time:	MWF, 1:25-2:40 In addition, there will be weekly team meetings with the professor and the client
Location:	SN011
Website:	http://wwwx.cs.unc.edu/Courses/comp523-f16

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 and for submitting individual essays. Weekly comments will be on a google document shared with the team. All team deliverables will be submitted through the team's website. All recommended or required readings will be available or referenced on the class website. All required external resources will be available without cost.

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 CI (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 programming COMP courses numbered 426 or higher. The most useful courses to already have are COMP 426 (web applications are the most common distributed system today) and COMP 521 (everyone needs a database).

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. Specific plan is to assign weekly readings and ask you to

respond to a question about the readings on Sakai. These readings will begin after Labor Day, once teams are assigned and running.

There is a lot of material that needs to be covered in the first couple of weeks. Therefore, the course will meet on Fridays for the first two weeks. After that, this slot is open for team or client meetings.

Project grades are based on code, documentation, ambition, effort, teamwork, and accomplishment. There will be 6 grades given for the project; they are equally weighted.

The final exam is a presentation of the end product and is scheduled according to the university calendar for Monday, December 12 at noon. It will be held in SN 011.

Key Dates

In class project demos will be October 5 and November 9. Completion of the 6 phases are

- Requirements phase, September 17
- Sprint 1, October 1
- Sprint 2, October 22 (yes, I know that is fall break)
- Sprint 3, November 5
- Sprint 4, November 19
- Completion, December 10

Required documentation for each phase is identified on the class website. Note that final code and documentation is due BEFORE the final presentations.

I will be scheduling a time when I am doing the final testing of each project. At least one team member will be present when I do the testing. This avoids the problem where a small bug prevents me from testing the bulk of the function or a poor user interface leaves me befuddled.

Grading Criteria

Overall breakdown

Project	80%
Readings	15%
Final Presentation	5%

Project

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

Process	30%
Code	35%
Documentation	35%

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 and taking on more than your share is not always a sign of a good team member.

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 process grades every week. You will receive individual and team grades. Key considerations for the grade are whether you meet your milestones, you are adapting your process as you learn, and whether you are working well as a team. You will be graded on a weekly basis, so the final grade will not be a surprise.
- 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. Interim deliverables are part of the process grade.
- For the final presentation, your grade will include both content and style.

Individual Reading Assignments

There will be an assigned paper each week after Labor Day. The paper and the prompt will be posted on Sakai and the response will be posted inline on Sakai. Answers are not to 4 sentences. The intent is to introduce you to the classic literature of the field. Grading will be on a 3-point scale:

0: didn't post a response
1: clearly didn't read
2: probably read but didn't absorb
3: got it

No late responses will be accepted.

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 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. The up-to-date schedule is posted on the web site. This schedule only covers class lectures, not assignment deliverables, which are detailed on the web site.

	Aug 24 Introduction	26 Concepts, Requirements
29 Client Presentations	31 Communicating with Clients	Sep 2 Client Meetings
Sept 5 Labor Day	7 Personas, User Stories, Use Cases	
12 Functional Specs	14 Project Management	17 end requirements
19 Frameworks	21 Framework Examples	
26 Teamwork and Collaborative Tools	28 Development Tools	Oct 1 end sprint 1
3 Virtual Machines	5 DEMOS	
10 UIs	12 No class (Yom Kippur)	
17 HCI	19 Accessibility	22 end sprint 2
24 Design	26 Models	
31 Architectures	Nov 2 QA	5 end sprint 3
7 Testing	9 DEMOS	
14 Security	16 Privacy	19 end sprint 4
21 Ethics	23 Thanksgiving	
28 Agile and CD/CI	30 Extreme Programming	
Dec 5 SCRUM	7 Guest Speaker	10 end development
12 FINAL (noon)		

Last updated 8/24/2016