

COMP 523: Software Engineering Laboratory

Bulletin Description

Organization and scheduling of software engineering projects, structured programming, and design. Each team meets with a client to develop requirements, and then designs, codes, and debugs program components and synthesizes them into a tested, documented program product.

General Course Info

Term: Spring 2023
Course/section: COMP 523 section 001
Time: MWF 8:45am to 10:00am
There will also be weekly meetings with the instructor (coach), many team meetings, and some client meetings
Location: FB 009 (in-person instruction)
Website: <http://www.cs.unc.edu/~stotts/COMP523-s23/>
Email: [help-comp523 @ cs . unc . edu](mailto:help-comp523@cs.unc.edu) (*send all class email here*)

Instructor Info

Name: Dr. David Stotts
Office: Sitterson 319
Phone: 919 . 590 . 6133
Web: <http://www.cs.unc.edu/~stotts>
Office Hours: Tue. 1:00 to 2:30, or by appt.
Until further notice, office hours will be held in Zoom
Zoom room: <https://unc.zoom.us/j/92807648791>

Any email sent to the instructor's personal account risks getting lost in the huge pile received every day, so please use the class account. The class email account will be read regularly by the instructor and TA. The instructor will communicate with students via the email addresses given in ConnectCarolina for the class. Please get in the habit of scanning your email for information from the instructor about the class. Message titles will contain "COMP 523".

Prerequisites

COMP 401, 410 and 411 plus two additional COMP courses (chosen from the list COMP 426, 431, 433, 520, 521, 530, 535, 560, 562, 575, 580, 585). The additional two courses will ideally cover software systems, 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

Class Meetings

Class meetings will be on Mondays and Wednesdays (and perhaps Fridays) from 8:45 am to 10:00 am. We will have all class meetings in person in FB 009. Attendance is required at class meetings (and at all coach meetings). We will use class to cover development processes, issue that developers should consider, technical issues in software development, and teams will present information as well in class meetings. I expect all students to give other teams the same attendance respect they want for their own presentations.

Other Meetings

Teams will conduct several other forms of meeting during the week and semester. Each week a team will meet with the coach to report on progress and get help solving problems. At the beginning of the semester, a team will meet with the client to develop requirements for the software system. And of course, multiple times each week the team will meet as a team to make progress on the project. These meetings may be face to face or in zoom as the needs of each team determine. The times for these meetings will be established for each team after the team is formed and the project begins. Your coach will have a checkbox list at each meeting to record your attendance, and progress towards your goals.

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 (if any... we may have no need for Sakai). Individual assignments (if any) 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. We will have (as required for APPLES) two writing assignments individually.

The team aspect of this class will vary from team to team, as each team will work with a different client to develop a different software product. The projects will all be software that the clients want and will find useful, but no team will be working on software that is critical to the well-being of any client or client's business. While you will not fail, we do not allow you to work in an environment where team failure causes harm to anyone.

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.

Course Requirements and Grading

The essence of the course is the faculty-coached team project. Teams of 3-4 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 in a traditional sense; there may 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 30-40 minute presentation to the class.

Project grades are based on code, documentation, ambition, effort, teamwork, and accomplishment. This includes attendance at meetings and the fulfillment of milestones for those meetings.

The final exam is a presentation of the end product.

Grading Criteria

Overall breakdown

- Project 85%
- Technology talk: 15%

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

- Process 20%
- Code 45%
- Documentation 20%
- Midterm presentation 5%
- Final Presentation 10%

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 evaluating 3 process phases: the requirements phase, the design and implementation phase, and the completion. 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 code portion also covers automated unit testing... was it complete and thorough?
- 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 midterm and final presentations, your grade will include both content and style. Midterm will be primarily an overview of your design and technological platform decisions. Final will be a working demo, design review, and lessons learned.

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.

Course Schedule and Key Dates

Lecture dates and topics, as well as dates of exams and any assignments are found in the class calendar: <http://www.cs.unc.edu/~stotts/COMP523-s23/calendar.html>

This is also accessible from the menu at the top of the class homepage.

Course Policies

Attendance: I expect student attendance at all class meetings, peer presentations and guest presentations. Specifically, I expect you to be at all technology talk and demo presentations, and to attend outside speakers. We hope to have a number of outstanding not-to-be-missed outside speakers during the semester. There is no textbook in this class because the content is not available in any simple form. If you are going to learn the content, you need to attend lectures and practice the material in your team.

Attendance in class is also important because I use class to explain the processes your team must follow and perform. I want to explain these once... to all... and not 15 separate times in meetings.

You must also attend a weekly meeting with the team coach.

Finally, I expect your attendance at class because the seat you occupy is in high demand. If you do not wish to be in class, there are many who do wish to be there.

Final Exam

The final exam in this class is the final presentation each team gives demonstrating the project results. All students are required to attend all presentation to be counted as present at the final exam.

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

Intellectual Property Policy

In this course you are developing a software product, with team mates, for a client, as a student of a state university (UNC) so therefore, using resources paid for by taxpayers of North Carolina. Your work may or may not work well when done. It may, or may not, have some real or potential monetary value. One way or the other, it is *intellectual property*. Who owns all this work? Who owns the intellectual property?

The policy we work with was developed by UNC and has been in force for a couple decades. First we start with the baseline situation that any information and property the client brings to the course is not under any Non Disclosure Agreement (NDA). We do not take client project ideas that require a team to sign any legal agreement to keep anything secret, or that would put any team into a legal situation that could lead to problematic legal issues.

That said, the basic policy is that the three parties involved in a project – team of students, client, and UNC – each own all of the project completely. This means that any party is free to use the project (in part or in whole) as they see fit to do so. It means that no party can *prevent* any other party from using the project as they see fit. It means that no party can *a priori* make a claim on the proceeds another party might make from use of the project. In more specific terms, it means the client own the code at the end of the semester, and it also means the team owns the code as well. And of course, it means UNC owns the products from the project as well.

This policy does *not* mean that it is wrong for a team to have the code repo for the project be private (not public). A client may wish to request a protected non-public repo. One team's project is owned by the team, but other teams do not own it. Other persons in the general public do not own it. And, it is an ethics violation for a team to lock out a client from code access. A private repo should be open to members of the team, and to the client, and to the instructor.

A team owning a project implies that each member of that team each owns the project. A client owning a project implies that the client (and representatives of the client) shall have access to all code and components of the project needed to execute the project and make normal use of it. This access must remain for the client (and the instructor) for a reasonable time (two months) after the end of the class to give the client time to access and fully possess the code.

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.

Disclaimer

The professor reserves to right to make changes to the syllabus, including assignment and project due dates, as well as percentages for assignments and exams towards final course grade. 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.