

# CSE 416: Software Engineering (Spring 2021)

[Scott D. Stoller](#)

---

## Course Description

This course covers fundamental topics in software engineering. The heart of the course is a capstone project in which teams of 4 students develop a complex software system following the principles and practices of software engineering. Core topics include:

- software development process, software lifecycle
- requirements capture, use cases, use case diagrams
- software design and architecture, component diagrams
- object-oriented design, class diagrams
- dynamic models: activity, communication, sequence, and state machine diagrams
- persistence, object-relational mapping
- software quality assurance

The course also covers some advanced topics, such as:

- software architecture
- advanced design patterns
- service-oriented architecture (SOA)
- cloud computing

The project requires use of a database and development of a GUI (typically Web-based), so knowledge of databases and web programming is important. While CSE216 and CSE316 provide a sufficient foundation, additional knowledge from CSE305 (Principles of Database Systems), CSE333 (User Interface Development), CSE336 (Internet Programming), or other experience is also helpful. CSE416 provides practical experience using these technologies and integrating them in more complex software systems, but teaching them is not a CSE416 course goal, because they are covered in other courses.

The [official course outcomes](#) are:

- An ability to perform project planning, requirements analysis, and system/test design.
  - An ability to work as a team to produce software systems that meet specifications while satisfying an implementation schedule.
  - An ability to produce professional quality oral/written presentations of system designs, reviews, and project demonstrations.
- 

## Readings

The Unified Modeling Language (UML) is used in this course to express software designs. I strongly recommend that you get one of the following two books on UML. I evaluated about a dozen UML books, and these are the best.

- Simon Bennett, John Skelton, and Ken Lunn, Schaum's Outline of UML, 2nd edition, McGraw-Hill, 2004. This book has thorough coverage of UML and abundant examples. It is out of print in the U.S., but inexpensive used copies are usually available, e.g., on [Amazon](#).
- Russ Miles and Kim Hamilton, [Learning UML 2.0](#), O'Reilly Media, 2006. This is a reasonably priced book with thorough coverage of UML. New and used copies are available, e.g., on [Amazon](#).

Benefits of getting one of these books: (1) It will save you time, compared to hunting on the web for information about details of each kind of diagram. (2) When grading UML diagrams, we will accept usage shown in either of these books as correct. Usage found on random web pages might be non-standard or incorrect.

In addition to a UML book, you should read: Scott W. Ambler. [Mapping Objects to Relational Databases: O/R Mapping In Detail](#).

If you have not studied UI design (i.e., usability, not just technology), for example in CSE 333, it is worth spending a few hours studying it before designing your system's GUI. A good book on this topic is: Doug Van Duyne, James Landay, and Jason Hong, *The Design of Sites*, 2nd edition, Addison-Wesley, 2007. Cheap used copies are often available on [Amazon](#) and eBay. A book that is not as good, but free for SBU students, is: Jenifer Tidwell, *Designing Interfaces*. An [online version is accessible for free via the SBU library](#). If you find other good online resources on UI design, let me know.

For other topics, the lecture notes are sufficient. You can refer to almost any software engineering textbook for additional information, if you like. The library has numerous software engineering textbooks; well-known ones include *Software Engineering: Theory and Practice* by Shari Lawrence Pfleeger, and *Software Engineering* by Ian Sommerville. Another interesting book, from which some of the lecture notes are derived, is [Software Engineering Best Practices: Lessons from Successful Projects in the Top Companies](#) by Capers Jones.

Other recommended readings:

- [Head First Object-Oriented Analysis and Design](#) by Brett D. McLaughlin, Gary Pollice, and Dave West. Its coverage of UML is incomplete, but it's a useful book on software design, with a lively writing style. It's available to Stony Brook students for free through [Safari books on-line](#).
- [Modern Debugging: The Art of Finding a Needle in a Haystack](#), by Diomidis Spinellis, CACM, 2019. You can access this for free through SBU's subscription to ACM Digital Library. From a computer connected to the campus network, the above link should work directly. From off-campus, select ACM Digital library from the [SBU Library Database List](#) and then search for this article by title.
- [What Google Learned From Its Quest to Build the Perfect Team](#), by Charles Duhigg, *The New York Times*, 2016. It's not specific to software engineering, but it's very interesting, not very long, and may help you become a better team player and manager.

---

## **[Blackboard](#)**

Announcements, lecture notes, assignments, etc., are posted in Blackboard. I recommend enabling email notifications (or your other preferred form of notification) for announcements, content changes, etc. To do this, click the down arrow next to your name in the right part of the page header, select Settings in the resulting menu, and then select Edit Notifications Settings. More detailed instructions are here: [Blackboard Help: Notifications](#).

Blackboard discussion forums are an important communication tool in this course. I recommend subscribing to each discussion forum, by clicking the Subscribe button in the header row above the list of threads in the forum; you will then receive an email informing you of each post.

---

## Teaching Assistants

TBD.

---

## Class

Tue and Thu, 11:30AM-12:50PM. Zoom link is available in Blackboard. You must be logged into your SBU Zoom account to join the Zoom meeting.

---

## Office Hours

**Instructor:** 9:30am-10:15am on Tue, Wed, Thu, and Fri. During office hours, I will be available on [Google Chat](#). Login to Google Chat using your @stonybrook.edu google account, and enter my name in the search box. This will create a private (one-on-one) chat message thread for us. (If Chat says my status is "away", don't always believe it; chat a message to me anyway.) If you'd like to talk, create a Zoom meeting or Google Meet meeting, and send me the link in Google Chat; if you'd like to talk with me together with some teammates, send them the link, too. I'll join the meeting, and we can talk. If I'm currently talking with other students, I'll mention this in the chat and tell you when I'm ready to join your meeting.

**TA:** Hanke Kimm, 10:10am-10:55am on Monday and 10:40am-11:25am on Friday. Held online in the same way as the instructor's office hours.

---

## Policies

**Academic Integrity.** If your team submits anything that includes any material created by other people, your team's submission must clearly identify that material and indicate its source; otherwise, it is plagiarism. Discussing assignments with other people is fine. However, each team must write its own code and documentation independently. Showing your team's written work to other students, giving it to them, or making it accessible to them (e.g., by putting it in a public repository) will be treated as a suspected academic integrity violation.

**Code Repository.** Code must be stored in a repository that (1) is private, i.e., not accessible to students on other teams, and (2) maintains a commit history. If a team's code is in a public

repository during the semester, and another team copies it or is suspected of copying it, both teams will be charged with suspected academic integrity violations for code sharing. Teams may be required to give the instructor and TAs access to the team's repository for examination of the commit history.

**Course Grades.** Each assignment is graded relative to some maximum number of points. The score is normalized into a number between zero and one (e.g.,  $19/20 = 0.95$ ) and then multiplied by the assignment weight. Course grades are based primarily on the sum of these weighted scores. The tentative weights are:

hw1-team	0
hw2-requirements	6
hw3-design1	10
hw4-design2	10
hw5-design-review	6
hw6-code1	11
hw7-code2	11
hw8-code-review	6
hw9-code3	40
TOTAL	100

**Electronic Communication.** (1) It is your responsibility to read email in your official University email account. For most students that is [Google Apps for Education](#), but you may [verify your official Electronic Post Office \(EPO\) address](#). (2) It is your responsibility to read the Blackboard discussion forums for this course. I post announcements, clarifications, etc., in them. Questions that do not require confidentiality should be posted in them (rather than sent by email), so everyone can benefit from the answers.

**Helpful Postings.** As an additional incentive for helpful postings (e.g., answering other students' questions), at the end of the semester, the two students with the largest numbers of helpful postings (provided it is at least three postings) will get +1% on their sum of weighted scores. The instructor will determine which postings count.

**Submission of Homework.** All homework is due by 11:59pm on the due date specified in the schedule posted on Blackboard. hw1 is submitted as a Google Form. All other assignments are submitted on Blackboard, using the appropriate "View/Complete Assignment" link (do not put any comments in the Comment box; we don't read it). Upload a single .zip or .tar.gz file containing all of the files for the assignment (unless the instructions for the assignment specify otherwise). The name of every file uploaded to Blackboard should start with the assignment name followed by the team name, e.g., hw2-Dream-Team.zip. Any team member can submit the electronic version on Blackboard, but please coordinate so that exactly one team member submits it.

**Late Submissions.** For homework assignments that involve presentations (design review, code review, and demo), lateness penalties are 10% per day and strictly enforced (except in extenuating circumstances), because rescheduling them is disruptive and inconvenient. For

other homework assignments, assignments submitted after the deadline receive a -1% penalty per 6 hours late (rounded up to the next multiple of 6).

**Team Communication.** Teams are encouraged to create and use a discussion group or messaging channel (e.g., a Google group or Slack channel) dedicated to the course project, rather than relying on email, SMS, etc.

**Teamwork.** Each team member is responsible for ensuring that he or she contributes equally, and that the Contributions section of each assignment documents their contributions accurately and in sufficient detail, in case any dispute arises later about whether they contributed a fair share to the project. **If you believe that your teammates are preventing you from contributing a fair share, or that one of your teammates is not contributing a fair share, discuss the situation with the instructor promptly. If you don't discuss such a situation with the instructor in a timely fashion, you are not taking the situation seriously. Team members not contributing their fair share may receive individual project scores lower than the team's scores, or they may, with consent from the instructor, be removed from the team and would then need to work on the project on their own.**

---

## University Policies

**Missed work due to extenuating circumstances.** Students are expected to attend every class, report for examinations and submit major graded coursework as scheduled. If a student is unable to attend lecture(s), report for any exams or complete major graded coursework as scheduled due to extenuating circumstances, the student must contact the instructor as soon as possible. Students may be requested to provide documentation to support their absence and/or may be referred to the Student Support Team for assistance. Students will be provided reasonable accommodations for missed exams, assignments or projects due to significant illness, tragedy or other personal emergencies. Please note, all students must follow Stony Brook, local, state and Centers for Disease Control and Prevention (CDC) guidelines to reduce the risk of transmission of COVID. For questions or more information click [here](#).

**Academic Integrity Statement.** Each student must pursue his or her academic goals honestly and be personally accountable for all submitted work. Representing another person's work as your own is always wrong. Faculty is required to report any suspected instances of academic dishonesty to the Academic Judiciary. Faculty in the Health Sciences Center (School of Health Technology & Management, Nursing, Social Welfare, Dental Medicine) and School of Medicine are required to follow their school-specific procedures. For more comprehensive information on academic integrity, including categories of academic dishonesty please refer to the academic judiciary website at [http://www.stonybrook.edu/commcms/academic\\_integrity/](http://www.stonybrook.edu/commcms/academic_integrity/).

**Critical Incident Management.** Stony Brook University expects students to respect the rights, privileges, and property of other people. Faculty are required to report to the Office of University Community Standards any disruptive behavior that interrupts their ability to teach, compromises the safety of the learning environment, or inhibits students' ability to learn. Faculty in the HSC Schools and the School of Medicine are required to follow their school-specific procedures. Further information about most academic matters can be found

in the Undergraduate Bulletin, the Undergraduate Class Schedule, and the Faculty-Employee Handbook.

**Student Accessibility Support Center Statement.** If you have a physical, psychological, medical or learning disability that may impact your course work, please contact Student Accessibility Support Center, Stony Brook Union Suite 107, (631) 632-6748, or at [sasc@stonybrook.edu](mailto:sasc@stonybrook.edu). They will determine with you what accommodations, if any, are necessary and appropriate. All information and documentation are confidential. Students who require assistance during emergency evacuation are encouraged to discuss their needs with their professors and Student Accessibility Support Center. For procedures and information go to the following website: <http://www.stonybrook.edu/ehs/fire/disabilities>.

---

## Web-Based Drawing Tools

[Lucidchart](#) and [draw.io](#) are good web-based diagramming tool with a library of shapes for UML diagrams (type "UML" in the search box near the top of the Shapes panel on the left). For Lucidchart, I suggest that you select "login with Google" and login with your @stonybrook.edu email address. Unlike Visual Paradigm for UML, which performs UML-specific error checking, these general-purpose diagramming tools will happily let you draw invalid diagrams. LucidChart (and perhaps also draw.io) is also good for creating sketches of UI screens.

[PowerPoint](#) (online version) via SBU's license for Office 365. PowerPoint is OK but less convenient, because it lacks a library of UML shapes.

---

## Offline UML Tools

[Visual Paradigm for UML \(Community Edition\)](#). *Under normal circumstances, I recommend using this tool, but during the pandemic, you probably will be better off using a tool that supports online collaborative editing.* This tool supports all of the types of UML diagrams used in this course, with comprehensive support for their features. It is relatively well documented and free for non-commercial use. It runs on Windows, Linux, Mac OS X, etc. It is installed on the computers in the Windows Lab. It puts a "Community Edition" watermark on the diagrams; that's fine for this course. There is a free online course, [Visual Paradigm Essential](#), that teaches the essentials.

**UMLet:** a lightweight open-source UML tool. It supports class diagrams, use case diagrams, sequence diagrams, state diagrams, deployment diagrams, activity diagrams, and component diagrams (I think), but not communication diagrams. I don't know whether it supports all features of those diagrams (e.g., loops in sequence diagrams).

**Violet.** Not recommended. It does not support component diagrams, communication diagrams, or deployment diagrams, and it has limited support for sequence diagrams (lacks support for actor icons, loops, alternatives, conditions, etc.).

---

## Code Repository



You may use any code repository satisfying the above [policy](#). Here are some suggestions.

**Git:** Free private Git repositories on [GitHub](#) and [GitLab](#) allow unlimited collaborators. Free private Git repositories on [Bitbucket](#) allow 4 or fewer team members (this is not ideal, since you might need to share the repo with the instructor or TA).

**Subversion:** The Computer Science Department runs a [Subversion server](#). If your team would like a repository on it, send a request for a repository, mentioning that it is for CSE 416, and specifying the names and SBU ID numbers of all team members, to [rt@cs.stonybrook.edu](mailto:rt@cs.stonybrook.edu). A repository will be created within a few days. You can then access your team's repository using your Computer Science Department Windows Lab username with your Stony Brook ID as the password.

---

## Software Development Tools

Every team is encouraged to use an Integrated Development Environment (IDE) for software development and debugging. Many teams from prior years recommend also using a build system such as [Jenkins continuous integration server](#) or [Maven](#).

---

## Screen Recording

You will submit videos of your system in action. Screen recording software produces much better videos than your cell phone camera. Here are some suggestions. If you find better tools, let me know.

**Game Bar:** On Windows, the built-in [Game Bar](#) can capture screen video and audio. I think it captures only a selected window, not the entire screen.

**QuickTime Player in macOS:** The QuickTime Player has built-in screen recording functionality, but with the default settings, it produces huge video files. Please change the settings!

**Skype for Business** is free for all SBU students. It is available on Windows (in the Start menu, it might be at top level or in the Microsoft Office folder) and Mac. It can record a selected window or the entire screen. Here are [screen recording instructions](#) (note: in step 2, you need to click on the small downward-facing triangle, not the settings gear next to it). To access the recording after closing the meeting window, click on that downward-facing triangle again and then select Tools then Recording Manager. Before recording, you can set the desired image resolution by clicking on the settings gear next to that downward-facing triangle, selecting Recording, and then setting the resolution.

**Open Broadcaster Software** is an [open-source](#) software for streaming and recording video. Available on Windows, Linux, and Mac. I have used it successfully on Windows.

---

## Database Server

You may use any database server. One option is the [Computer Science Department's database server](#). To get an account on it, send a request, mentioning that it is for CSE 416,

and specifying the desired DBMS (MySQL or Oracle) and the names and SBU ID numbers of all team members, to [rt@cs.stonybrook.edu](mailto:rt@cs.stonybrook.edu). An account will be created within a few days.