CS 673 Software Engineering

Spring 2021

Department of Computer Science
Metropolitan College
Boston University

Instructor Information

Name: Yuting Zhang
Office: 1010 Commonwealth Ave., Room 322
Email: danazh at bu dot edu
URL: http://people.bu.edu/danazh

Office Hours:

Tue 4:00-5:00PM, Thu 11AM-12PM Or by appointment (through Zoom)

Feel free to ask me any questions before or after class. You can always contact me by email. **Please always add "CS673 (or cs673)" in the subject of your email**.

Course Information

Lecture Time & Place
Wednesday 6:00-8:45 PM CAS B06B

Prerequisites
At least two 500 level or above programming intensive courses. Or instructor's consent.
(This course is not about programming. However the programming skill is the prerequisite. Students should be **familiar** with **OO concepts** and **proficient** in at least **one high-level programming language** before taking this course.)

Reference Books:

Preferred SE Textbook:

- Eric Braude, Michael E. Bernstein. ***Software Engineering: Modern Approaches (2rd Edition)***. Waveland Press, Inc. *(ISDN:9781478632306)*

Other Reference SE Textbooks:

- Robert C. Martin. Agile Software Development, Principles, Patterns, and Practices.
- Bernd Bruegge and Allen H. Dutoit. Object-Oriented Software Engineering: Using UML, Patterns and Java.
- Shari Lawrence Pfleeger, Joanne M. Atlee. Software Engineering: Theory and Practice
- Roger S. Pressman. Software Engineering: A Practitioner's Approach.
- Hans Van Vliet. Software Engineering: Principles and Practice.
- Ian Sommerville. Software Engineering
- Ian Sommerville. Engineering Software Products: An Introduction to Modern Software Engineering

Other Classical Books for Software Engineers

- Frederick P. Brooks, Jr. The Mythical Man Month.
- Elisabeth Freeman, Eric Freeman, Bert Bates, and Kathy Sierra. Head First Design Patterns.
- Martin Fowler, Kent Beck, Don Roberts. Refactoring: Improving the Design of Existing Code.
- Steve McConnell. Code Complete: A Practical Handbook of Software Construction.
- Robert C. Martin. Clean Code: A Handbook of Agile Software Craftsmanship.

Other Reading Materials

- Microsoft Security Development Lifecycle: https://www.microsoft.com/en-us/sdl/
- OWASP
    - SAMM project: https://www.owasp.org/index.php/OWASP_SAMM_Project
    - TOP 10: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project
    - Developer Guide: https://www.owasp.org/index.php/Category:OWASP_Guide_Project
    - Testing Guide: https://www.owasp.org/index.php/Category:OWASP_Testing_Project
- SourceMaking (Design Patterns, Refactoring, UML): https://sourcemaking.com/

Spring 2021 COVID-19 Policies

Compliance: All students returning to campus will be required, through a digital agreement, to commit to a set of Health Commitments and Expectations including face coverings, symptom attestation, testing, contact tracing, quarantine, and isolation. The agreement makes clear that compliance is a condition of being a member of our on-campus community.

You have a critical role to play in minimizing transmission of COVID-19 within the University community, so the University is requiring that you make your own health and safety

commitments. Additionally, **if you will be attending this class in person, you will be asked to show your Healthway badge on your mobile device to the instructor in the classroom prior to starting class, and wear your face mask over your mouth and nose at all times.** If you do not comply with these rules you will be asked to leave the classroom. If you refuse to leave the class, the instructor will inform the class that they will not proceed with instruction until you leave the room. If you still refuse to leave the room, the instructor will dismiss the class and will contact the academic Dean's office for follow up.

Boston University is committed to offering the best learning environment for you, but to succeed, we need your help. We all must be responsible and respectful. If you do not want to follow these guidelines, you must participate in class remotely, so that you do not put your classmates or others at undue risk. We are counting on all members of our community to be courteous and collegial, whether they are with classmates and colleagues on campus, in the classroom, or engaging with us remotely, as we work together this fall semester.

Description (from Catalog)

Overview of techniques and tools to develop high quality software. Topics include software development life cycle such as Agile and DevOps, requirements analysis, software design, programming techniques, refactoring, testing, as well as software management issues. An overview of secure software development processes and techniques will also be introduced. This course also features a semester-long group project where students will design and develop a real world software system in groups using Agile methodology and various SE tools, including UML tools, project management tools, programming frameworks, unit and system testing tools , integration tools and version control tools.

Learning Outcomes

At the end of the semester, students are expected to

1. Describe and compare major software process models and activities in the software process.
2. Explain the agile methodology and techniques, and apply these in a real-world team-based project to develop a high-quality software system on time.
3. Be aware of some security risks in the software project and techniques to enhance the software security.
4. Use proficiently various SE tools including the UML tool, the project management tool, programming tools, testing tools, the version control tool, etc.
5. Communicate effectively with team members and customers.
6. Present clearly the software project in both the oral and written form.

Learning Outcomes Assessment

- Class participation
- Reading and study
- 3 Labs
    - Lab1: Set up Git (LO2, LO4)
    - Lab2: Requirement Analysis (LO2, LO4)
    - Lab3: Refactoring and TDD (LO2, LO4)
- Semester-long project: (All LOs)
- 3 Quizzes (LO1, LO2, LO3, LO4)
- The Final Exam (LO1, LO2, LO3, LO4)

This course is featured with a semester-long team-based project. Each team should have about 4-6 students. Every member of the team is expected to contribute a roughly equal share to the project.

Course Policies

Grading Policy

The grade that a student receives in this class will be based on the class participation, in-class exercises or quizzes, the project and the final exam. The grade is breakdown as below. All percentages are approximate and the instructor reserves the right to make necessary changes.

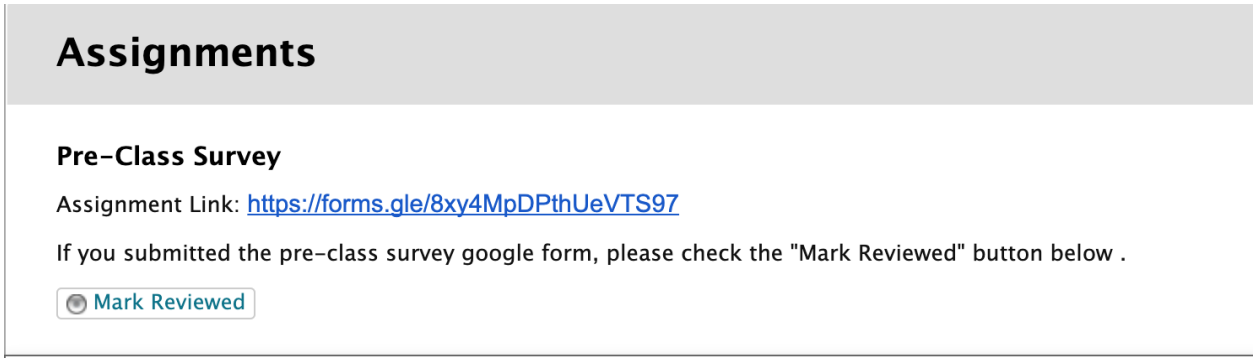
- 5% on class participation
- 9% on lab assignments (3 small labs)
- 6% on quizzes (3 quizzes)
- 50% on the semester-long project
- 30% on the final exam

Letter grade/numerical grade conversion is shown below:

A (95-100)	A- (90-94)	
B+ (85-89)	B (80-84)	B- (79-77)
C+ (74-76)	C (70-73)	C- (65-70)
D (60-65)	F (0 – 59)	

Assignment Submission

All labs should be submitted directly on blackboard. The project assignments are mostly done on the github and the google drive. To help track the progress, when you finish an assignment on the github and the google drive, please check the “Marked Reviewed” button on the assignment page as shown below.



Assignment Late Policy

Each assignment has a deadline. For all individual assignments, the late assignments will be penalized within three days with penalty. No assignments will be accepted three days after the deadline.

All project deadlines are firm. A deadline miss means zero for the grade of that phase.

It is the students' responsibility to keep secure backups of all assignments.

Academic Integrity

Academic conduct in general and MET College rule in particular require that all references and uses of the work of others must be clearly cited. All instances of plagiarism must be reported to the College for action. *For the full text of the academic conduct code, please check* http://www.bu.edu/met/for-students/met-policies-procedures-resources/academic-conduct-code/.

Course Outline

This course is organized into six modules of about 2 or 3 lectures each.

Module 1

Topics:

1. Introduction to Software Engineering & Software Process: SE is difficult, SE Concepts and Terminology, SE Ethics, Software Process including Waterfall, Spiral Model, Unified Process, and Agile.
2. Agile Methodology & Software Project Management: Manifesto, Agile Principles, eXtreme Programming, Scrum, AUP, DevOps, DevSecOps, Software Quality, Software Configuration Management, Risk Management

Reading: Online Lecture Notes, Braude (Part I, II and III), or related chapters in other textbooks. Additional papers provided

Assignments: Lab1, Project Assignments

Module 2

Topics:

1. Requirement Analysis and Management using User Stories: Gather requirements, User Stories, The INVEST Principle, Acceptance Tests, User Story Management, Product Backlog, Velocity and Story Points, Epics and themes, Scope Creep, Technical Debt, User Story Management Tool
2. From Requirements to Design - UML class diagrams and state transition diagrams: UI Mockups, State Transition Diagrams, Entity-Boundary-Control, Class Diagrams

Reading: Online Lecture Notes, Braude, Part IV, or related chapters in other textbooks

Assignments: Lab2, Project Assignments, Quiz 1

Module 3

Topics:

1. High Level Design: Design Goals, Software Architecture, MVC, Layered and Tiered Architecture, SOA, Microservices, REST
2. Design Principles and Design Patterns: Class diagrams, OO Reuse, Design Principles, MVC Compound Design Patterns

Reading: Online Lecture Notes, Braude, Part V, or related chapters in other textbooks

Assignments: Project Assignments

Module 4

Topics:

1. Implementation: Programming languages and Frameworks, Coding Standards, Code Smells, Refactoring Techniques
2. Testing: Regression Testing, Type of Testing, Testing Plan, Test Cases, Test Code, TDD and BDD

Reading: Online Lecture Notes, Braude, Part I, II and III, or related chapters in other textbooks

Assignments: Lab 3, Project Assignments, Quiz 2

Module 5

Topics:

1. More UML Tools in Requirement Analysis and Design: Use Case Model, Class Diagrams, State Machine Diagrams, State Pattern, Sequence Diagrams
2. Testing Techniques: Whitebox Testing, Testing Coverage, Blackbox Testing, Domain Testing, Integration Testing

Reading: Online Lecture Notes, Braude, Part I, II and III, or related chapters in other textbooks

Assignments: Project Assignments

Module 6

Topics:

1. Secure Software Development Process: CIA and IAAA, Software Security, Seven Touchpoints, Microsoft SDL, OWASP SAMM
2. Software Security Practices: Security requirements, Architecture Risk Analysis, Microsoft STRIDE, Code Review, Secure Coding Standards, Security Testing, SAST, DAST, IAST, RSP, Top Vulnerability Lists

Reading: Online Lecture Notes, Braude, Part I, II and III, or related chapters in other textbooks

Assignments: Project Assignments , Quiz 3

Project Assignments:

The project assignments are mostly done through google drive and github. To help both students and the instructor keep track of the assignments, we also create a checkpoint for each assignment on the blackboard. Make sure to submit checkpoints on the blackboard.

There are two types of project assignments:

1. Individual assignments: each student should submit his/her own assignment through google drive and blackboard.
    a. Weekly report: fill a row in your own sheet each week in the group weekly report on google doc. After you are done, submit the check question assignment on the blackboard.
    b. Post-iteration self and peer review: fill the iteration review survey form on google form. After you are done, submit the check question assignment on the blackboard.

2. Group assignments: each group only needs to submit one copy of the whole group work on github and blackboard. Students will work on the group documents collaboratively on google doc. At each iteration release, the group leader or the configuration leader or some designated member will archive the documents on the github, together with the source code to create a release. After it is done, the group leader (or the designated member) should submit the checkpoints on the blackboard.
    a. Iteration 0 Release including
        i. Readme.md
        ii. Doc/ProjX_SPPP
        iii. Doc/ProjX_meetingminutes
        iv. Doc/ProjX_weeklyreport
        v. Doc/ProjX_presentation_iter0
    b. Iteration 1 Release including
        i. README.md (updated)

ii. Doc/ProjX_SPPP (updated or v2)
iii. Doc/ProjX_meetingminutes (updated)
iv. Doc/ProjX_weeklyreport (updated)
v. Doc/ProjX_userstories
vi. Doc/ProjX_SDD
vii. Doc/ProjX_Presentation_iter1
viii. Code/… : runnable source

c. Iteration 2 Release including
   i. README.md (updated)
   ii. Doc/ProjX_SPPP (updated)
   iii. Doc/ProjX_meetingminutes (updated)
   iv. Doc/ProjX_weeklyreport (updated)
   v. Doc/ProjX_userstories (updated)
   vi. Doc/Proj1_SDD (updated)
   vii. Doc/ProjX_STD
   viii. Doc/ProjX_Presentation_iter2
   ix. Code/… : runnable source

d. Iteration 3 Release (from the master branch) including
   i. README.md (updated)
   ii. Doc/ProjX_SPPP (updated)
   iii. Doc/ProjX_meetingminutes (updated)
   iv. Doc/ProjX_weeklyreport (updated)
   v. Doc/ProjX_userstories (updated)
   vi. Doc/Proj1_SDD (updated)
   vii. Doc/ProjX_STD (updated)
   viii. Doc/ProjX_Deployment (if applied)
   ix. Doc/Proj1_Presentation_final
   x. Code/… : runnable source code

Course Schedule

(This is a tentative schedule. It is subject to change based on the class progress and students' feedback**)**

Both the lecture and the project use iterative approaches. The lecture includes two iterations. The project includes an initial planning and then three mini-project iterations.

Class #	Date	Module #	Topics	Course Assignments	Project Assignments
1	01/27	Module 1	Topic 1		Iteration 0 Starts (01/27-02/17)
2	02/03	Module1	Topic 2	Lab 1 is Due	
3	02/10	Module 2	Topic 1		
4	02/17	Module 2	Topic 2	Lab 2 is Due	Iteration 0 Presentation Iteration 1 Starts (02/17-03/10)
5	02/24	Module 3	Topic 1		
6	03/03	Module 3	Topic 2	Quiz 1 is Due	
7	03/10	Module 4	Topic1		Iteration 1 Presentation Iteration 2 Starts (03/10-04/07)
8	03/17	Module 4	Topic 2		
9	03/24	Module 5	Topic 1	Quiz 2 is Due Lab 3 is Due	
	03/31	Wellness Day			
10	04/07	Module 5	Topic 2		Iteration 2 Presentation Iteration 3 Starts (04/07- 04/28)
11	04/14	Module 6	Topic 1, 2		
	04/21	Substitute Monday's Class		Quiz 3 is Due	
12	04/28	Final Project Representation			Iteration 3 Ends Final Presentation
13	05/05	Final Exam			