MET CS673 A1

Software Engineering

Spring 2022

Course Information:	Instructor:	Ronald Czik
	e-mail:	rec@bu.edu
	Course Website:	https://learn.bu.edu/ultra/courses/_82020_1/cl/outline

Office Hours: Either before or after class and by prior arrangement.

Location CAS 213

Days: Wednesdays 6:00 pm – 8:45 pm

Prerequisites: MET CS 342 and at least two 500 level or above computer programming-intensive courses or consent of instructor. Students should be familiar with object-oriented design concepts.

Note: This is a live document. The current version presents a tentative class schedule. It is subjected to change according to the progress of the class and the feedback of the student. Please constantly check the updated version online on blackboard.

Texts:

- *Software Engineering Products – An Introduction to Modern Software Engineering* by Ian Sommerville

Additional References:

- *Software Engineering: Modern Approaches* by Eric J. Braude and Michael E. Bernstein, Wiley; second edition
- *Software Engineering* by Ian Sommerville; tenth edition
- *Being Agile* by Mario E. Moreia
- Various other references as reviewed in class

Course Overview: Focus on techniques for the construction of reliable, efficient, and cost-effective software including requirements analysis, software design, programming methodologies (especially agile), testing procedures, continuous integration, software development tools and selected management issues.

To reinforce the central concepts in software engineering, students will work in teams building a product. Specialization within groups may be permitted, but all members must know all parts. Teams will give a mid-semester as well as a final presentation on the last day of class. Additionally, teams will be asked for interim, informal verbal status updates during the semester.

Each week, each team is required to submit:

- A project status report (PSR), which describes the state of the project as a whole
- All notes from weekly team meetings

Each week each student is required to submit:

- An individual status reports describing their individual work

Besides the book chapters, the additional reading material may be assigned for each topic. Reading before and after class is required and essential to succeed in this course. Students are responsible for ALL the materials covered in the lectures.

Course Goals: Students will be able to plan software application projects, gather requirements, create architectures, create a design, implement the code, experiment with agile approaches and continuous integration, and test software products.

At the end of the semester, students are expected to:

- Have a fundamental understanding of major software process models and activities in software process.
- Be competent in applying the software development process and best practices in real world team-based project to produce high-quality software on time.
- Be competent in effectively communicating with team members and customers.
- Be comfortable with formal project presentation.

Grades: The grade that a student receives in this class will be based on class participation (including status reports and presentations), project artifacts (code, documentation, etc.), and individual work. The grade breakdown is shown below. All percentages are approximate and the instructor reserves the right to make necessary changes.

Artifact	Weight
Individual contributions, PSRs, class participation	20%
Project Proposal	5%
Mid-semester submission	20%
Final project submission	30%
Final exam	25%

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)

PROJECTS

This course is featured with a semester-long, team-based project. Each team will have 5-7 students. Every member of the team is expected to contribute an equal share to the project.

Presentation: At the end of iteration 1 each team will present a mid-project presentation including a demo of working software to measure progress. At the last class each team will present their final work. Every student is required to participate in the presentations.

Management/Team work: Since this is a team work project. Management, communication and collaboration between members are very important. This activity is evaluated based on meeting minutes, status reports, and other documentation based on the development process used.

Implementation: the correctness, complexity and quality of the source code source will be evaluated particular for this activity. Refactoring is an important practice to improve the source code quality.

Test: both unit testing and system testing should be performed. The test code (e.g. junit test code and/or selenium test scripts) are the main deliverables for this activity.

Deployment: The final software should be easily deployed on the customer environment. If it is a desktop/mobile application, an executable file should be generated in order for the customer to easily install it. If it is web-based application, it should be deployed on a web server in order for the customer to easily access it through common browser such as Chrome, Firefox, IE etc.

The grade for each individual member will depend on your group project grade and his/her own contribution to the project, as well as type of activities based on your role in the group.

Tools:

- Project management tool: Trello (https://trello.com/) or PivotalTracker (https://www.pivotaltracker.com/)
- Version control tool: https://github.com
- Document collaboration and sharing: http://drive.google.com
- Communication: https://slack.com
- Testing tool: http://www.seleniumhq.org/ http://junit.org
- Each team must provide access to the grader and instructor to the above tools used in your project.

Academic Integrity: Academic conduct in general, and MET College, 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: https://www.bu.edu/academics/policies/academic-conduct-code/

Attendance: Class is held Thursday evenings from 18:00 to 20:45 eastern US time. Attendance is required. You are responsible for all material discussed in class. In general, no makeup will be given unless an extremely good, verifiable reason, is given in advance.

Please respect your classmates by silencing your cell phones and other electronic devices before class begins.

Calendar

Approximately 30 mins will be allotted at the end of each class for team meetings and team questions for the instructor. The schedule is subject to change.

Class	Date	Topics	Readings	Assignments
1	1/26/2022	• Introduction – course outline • Software Products	Ch. 1	• Project assigned
2	2/2/2022	• Agile software engineering • Team project proposals	Ch. 2	
3	2/9/2022	• Features, Scenarios, and Stories	Ch. 3	• Project proposal due • Project status report due
4	2/16/2022	• Software Architecture Part 1	Ch. 4	
5	2/23/2022	• Software Architecture Part 2	Ch. 4	• Project status report due
6	3/2/2022	• Mid semester team presentations		• Mid semester submission
	3/9/2022	• SPRING RECESS		
7	3/16/2022	• Cloud-Based Software	Ch. 5	• Project status report due
8	3/23/2022	• Microservices Architecture	Ch. 6	•
9	3/30/2022	• Security and Privacy	Ch. 7	• Project status report due
10	4/6/2022	• Reliable Programming	Ch. 8	
11	4/13/2022	• Testing	Ch. 9	• Project status report due
	4/20/2022	• NO CLASS – SUBSTITUTE MONDAY		•
12	4/27/2022	• DevOps and Code Management	Ch. 10	•
13	5/4/2022	• Team final presentations		• Final project due
14	5/11/2022	• Final exam		

Revised 1/18/2022 1:00 pm