MET CS-473/673 D1

Introduction to Software Engineering

Summer 2016

Course Information:	Instructor:	Ronald Czik
	e-mail:	rec@bu.edu
	Course Website	https://learn.bu.edu/webapps/blackboard/execute/launcher?type=ourse&id=_32972_1&url=

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

Location: EPC 201

Days: Mondays 6:00 - 9:30 p.m.

Prerequisites: MET CS 342 and at least one 500 level computer programming-intensive course of consent by the instructor.

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 students. Please constantly check the updated version online on blackboard.

Texts:

- *Software Engineering: Modern Approaches* by Eric J. Braude and Michael E. Bernstein Wiley; 2 edition (April 5, 2010) ISBN-10: 0471692085ISBN-13: 978-0471692089

Additional References:

- *Software Engineering 10th Edition by Ian Sommerville (ISBN-13: 978-0133943030, ISBN-10: 0133943038)*
- *Software Engineering: A Practitioner's Approach by R. Pressman* (McGraw Hill)
- *The Mythical Man-Month 20th anniversary edition by F. Brooks et alA Discipline for Software Engineering* by W. Humphries ("the PSP book")
- *Introduction to the Team Software Process* by W. Humphries ("the TSP book")
- *Being Agile* by Mario E. Moreia

Course Overview:

Techniques for the construction of reliable, efficient, and cost-effective software.

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 on most project parts. Specialization within groups may be permitted, but all members must know all parts. Teams will give presentations on the last day of class, and will be called upon to give in-progress reviews.

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 and lab sessions including any topics not in the textbooks.

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 comfortable with CASE (Computer Aided Software Engineering) tools such as UML tool, testing tool, version control tool etc.
- 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, in-class exercise, quizzes, project and the exam. The grade is breakdown as shown below. All percentages are approximate and the instructor reserves the right to make necessary changes.

	Weight
Class Participation	5%
Weekly hand-ins	10%
Final	25%
Project	60%

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 should have at about 4-6 students. Every member of the team is expected to contribute a roughly equal share to the project.

Presentation: At the end of each iteration a presentation by each team is required. Except the first planning phase, a demo is required in each iteration representation. Every student is required to participate the final (iteration 3) presentation, and any one of the previous 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, weekly report, 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. STD and the test code (e.g. junit test code and 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:

- Requirement management tool: Pivotaltracker (https://www.pivotaltracker.com/) or Trello (https://trello.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/ or http://junit.org

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: http://www.bu.edu/met/for-students/met-policies-procedures-resources/academic-conduct-code/

Attendance: Class is held Monday evenings from 6:00 to 9:30 p.m. in EPC 201. Attendance is required. You are responsible for all material discussed in class. In general, no makeup quizzes and exam 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

Both the lecture and project use iterative approaches. The lecture includes two iterations. The project includes planning 3 iterations. The schedule is subject to change according to the progress of the class and the feedback of the students.

Lecture	Date	Topic	Readings	Project Deadlines
1	05/23/16	Overview of software engineering	Part I	Project is assigned, initial planning starts
2	06/03/16	NOTE: FRIDAY CLASS. Project Management: Requirements analysis (User Stories) and scheduling, quality	Part II	Proposals Due Iteration 1 start
3	06/06/16	Project Management (requirements to design – UML)	Part II	
4	06/13/16	Project Management from requirements to design (UML diagrams)	Part II	Iteration 1 done Demo presented Iteration 2 starts
5	06/2016	Software Design –Patterns (MVC)	Part III	
6	06/27/16	Software Design – Test driven design, refactoring	Part IV	
7	07/09/16	NOTE: FRIDAY CLASS. More on Patterns, Quality metrics	Part IV	Iteration 2 done Demo presented Iteration 3 starts
8	07/11/16	Implementation	Part VI	
9	07/18/16	More on implementation	Part VI	
10	07/25/16	Testing	Part VII	
11	08/01/16	Presentations		Iteration 3 done Final presentation
12	08/08/16	Final exam		

Last modified 05/23/2016 10:00 rec