MET CS 633 Online Course Syllabus (2026 Spring 1)
Software Quality, Testing and Security Management

Instructor

Alex Elentukh elentukh@bu.edu

Course Duration

Start: January 13, 2026

End: February 28, 2026

Course Credit

4 credits

Course Description

Theory and practice of security and quality assurance and testing for each step of the software development cycle. Verification vs. validation. Test case design techniques, test coverage criteria, security development and verification practices, and tools for static and dynamic analysis. Standards. Test-driven development. QA for maintenance and legacy applications. From a project management knowledge perspective, this course covers the methods, tools and techniques associated with the following processes – Plan Quality, Perform Quality Assurance, and Perform Quality Control.

Prerequisites: MET CS 520 or MET CS 521, or instructor consent

Course Learning Objectives

By successfully completing this course, you will be able to do the following:

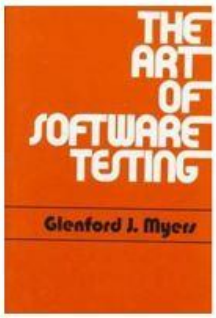
1. Manage a Software Quality Assurance function covering all phases of a global product development life cycle.
2. Effectively play the role of a Software Engineering Manager in a context of IEEE CSDP (Certified Software Development Practitioner) with a specific emphasis on organizational policies as guiding principles.
3. Solicit and define scope requirements as part of the product backlog grooming.
4. Assess common security threats and establish corresponding deterrents.
5. Select an estimation method that is appropriate for a specific phase. Oversee adoption of a consistent methodology to narrow the Cone of Uncertainty.
6. Play a role in a peer review verification and request and provide constructive and concise comments.

7. Support the Scrum delivery framework and become aware of several agile certification paths.
8. Evaluate software development tools (approved, allowed, restricted), while following the Magic Quadrant technique.
9. Articulate a strategy for system and unit tests leading to continuous integration and delivery.
10. Structure a project asset library aiming at single-click navigation to a requested artifact.
11. Provide leadership to a process program that is mapped into PMI and/or SEI CMMI as an improvement model.

Course Materials

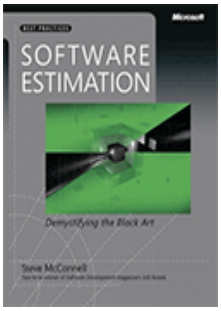
There are several books referenced throughout the course. **These books are optional**. You do not need to purchase them. Most of the course concepts are expounded through the class lecture notes, comprised of some two hundred pages.

You will be also required to search through various online resources.



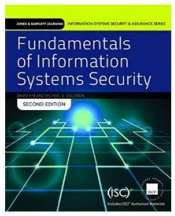
Glenford Myers is an American computer scientist, entrepreneur, and author. He founded two successful high-tech companies, authored eight textbooks in the computer sciences, and made important contributions in microprocessor architecture.

The orange book cover on the left – corresponds to the original 1978 edition. Download the 3rd edition eBook from the BU Library.

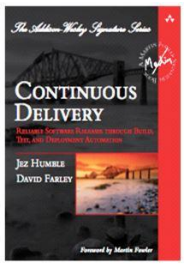


Steve McConnell, author of the 2006 book, *Software Estimation: Demystifying the Black Art* (Redmond, WA: Microsoft Press, ISBN: 0735605351). Steve McConnell is a well-known industry practitioner and Chief Software Engineer at Construx. *Software Development* magazine named Steve one of the three most influential people in the software industry along with Bill Gates and Linus Torvalds. Steve's book on estimation is available in electronic form and could be delivered instantly.

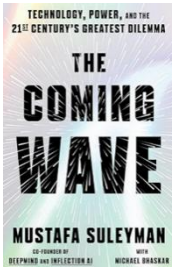
David Platt teaches User Experience Engineering at Harvard University Extension School and at companies all over the world. He's the author of 12 programming books, including *Why Software Sucks* (Addison-Wesley, 2006) and *Introducing Microsoft .NET* (Microsoft Press, 2003). Microsoft named him a Software Legend in 2002.



Fundamentals of Information Systems Security is by David Kim and Michael Solomon (ISBN-13: 978-1284031621). David Kim is the president of Security Evolution Inc (SEI) and provides consulting services around the world.



Jez Humble is a principle consultant at Thoughtworks helping organizations deliver high quality software fast and reliably. This groundbreaking book sets out the principles and technical practices enabling rapid, incremental software release. It is a part of the sea change that brought about a culture of continuous delivery.



Mustafa Suleyman is an artificial intelligence entrepreneur. He is currently the EVP of Microsoft AI division, including Copilot, Bing and Edge. He is the co-founder and former head of DeepMind, an AI company acquired by Google. After leaving DeepMind in 2022, he co-founded Inflection AI, a machine learning and generative AI company. Suleyman was named by Time among the most influential people in artificial intelligence in 2023 and in 2024.

Study Guide and Timeline of Deliverables

The following schedule is tentative and is subject to change according to the progress of the class and the feedback from students. It covers six modules with two live sessions per module.

* designates an optional activity. You will receive a feedback to an optional assignment, although it will not contribute to your final grade.

Module 1 Study Guide and Deliverables

(January 13 – January 19, 2026)

Module Theme and Topics:

- Definition and scope of Quality Assurance
- Globalization Trends
- Requirements Engineering
- Engineering Management

Reading References:

- Module 1 content notes
- Karl Wiegers
- tutorial
- GitHub tutorial

Assessments:

Trello Quiz 1: Due **Tuesday, January 20 at 6:00 AM ET**

Assignments:

- Assignment 1: Provide alternative definitions
- Assignment 2: Traverse Project Repository
- Assignment 3: Converse with Chatbot for Most Effective Learning Strategy

All Assignments due **Tuesday, January 20 at 6:00 AM ET**

Team Project:

- Register with Pivotal
- Create GIT account
- Submit initial slides of project report with team's composition, roles, policy, project scope

All Term Project deliverables due **Tuesday, January 20 at 6:00 AM ET**

Class Discussions *:

- Introduce yourself and welcome other students
- Compare Company Policies of Google and GM

Live Classrooms:

- Course Lectures: **Tuesday, January 13 at 9 PM ET**
- Assignments & Quizzes: **Thursday, January 15 at 9 PM ET**

Module 2 Study Guide and Deliverables

(January 20 – January 26, 2026)

Module Theme and Topics:

- Software Configuration Management (SCM)
- Estimation

Reading References:

- Module 2 content notes

- Steve McConnell
  - Section 4 "Cone"
  - Section 11 "Analogy"
  - Section 12.3 "Story Points"

Assessments:

Quiz 2: Due **Tuesday, January 27 at 6:00 AM ET**

Assignments:

- Assignment 1: Analyze Git's Branching and Merging
- Assignment 2: Motivation Principle
- Assignment 3: Provide consistent assessment of six requirements processes

All Assignments due **Tuesday, January 27 at 6:00 AM ET**

Team Project:

- Document Personas
- Develop Requirements in Pivotal
- Start Coding

All Term Project deliverables due **Tuesday, January 28 at 6:00 AM ET**

Class Discussions *:

- Comment on a paper, "The Art of Changing the History"
- Respond the most common question, while teaching estimation class.

Live Classrooms:

- Course Lectures: **Tuesday, January 20 at 9 PM ET**
- Assignments & Quizzes: **Thursday, January 22 at 9 PM ET**

Module 3 Study Guide and Deliverables

(January 27 – February 2, 2026)

Module Topics:

- Agile
- Static Testing

Readings References:

- Module 3 online content
- Scrum Guide (Nov. 2020)

Assessments:

Quiz 3: Due **Tuesday, February 3 at 6:00 AM ET**

Assignments:

- Assignment 1: Maintain time-trends for six scrums
- Assignment 2: Calculate all permutations of the Cost of Delay
- Assignment 3: Optimize Peer Reviews

All Assignments due **Tuesday, February 3 at 6:00 AM ET**

Team Project:

- Provide Estimation Record
- Compile CI List
- Document Tools Connectivity Diagram

All Term Project deliverables due **Tuesday, February 3 at 6:00 AM ET**

Class Discussions *:

- Give a hand to a Moderator in a contentious peer review

Live Classrooms:

- Course Lectures: **Tuesday, January 27 at 9 PM ET**
- Assignments & Quizzes: **Thursday, January 29 at 9 PM ET**

Module 4 Study Guide and Deliverables

(February 3 – February 9, 2026)

Module Topics:

- IS Security
- Elements of Software Design
- Common Tools Supporting Common Process

Readings References:

- Module 4 online content
- David Pratt
- David Kim

Assessments:

Quiz 4: Due **Tuesday, February 10 at 6:00 AM ET**

Assignments:

- Assignment 1: Tools for a Unit Test
- Assignment 2: Position 22 Software Tools into 11 Categories

All Assignments due **Tuesday, February 10 at 6:00 AM ET**

Team Project:

- Document Selected Use Cases
- Converge on a Components Interaction Diagram
- Document State Transitions

All Term Project deliverables due **Tuesday, February 10 at 6:00 AM ET**

Class Discussions *:

- Should Ed Snowden be pardoned?
- How to manage a repository of best coding practices.

Live Classrooms:

- Course Lectures: **Tuesday, February 3 at 9 PM ET**
- Assignments & Quizzes: **Thursday, February 5 at 9 PM ET**

Module 5 Study Guide and Deliverables

(February 10 – February 16, 2026)

Module Topics:

- System Test
- Unit Test
- Continuous Delivery

Readings References:

- Module 5 online content
- Glenford Myers, Chapter 1 Self Assessment Test
- MIT courseware
- Jez Humble, Chapter 5, Deployment Pipeline

Assessments:

Quiz 5: Due **Tuesday, February 17 at 6:00 AM ET**

Assignments:

- Assignment 1: Data-Driven Test Design
- Assignment 2: Review Online Courseware
- Assignment 3: Essential Test Coverage Based on State Transitions

All Assignments due **Tuesday, February 17 at 6:00 AM ET**

Team Project:

- Transform Mock-ups into Wireframes
- Complete Coding
- Demo a hosted prototype

All Term Project deliverables due **Tuesday, February 17 at 6:00 AM ET**

Class Discussions *:

- Link two diverse notions from Andy Grove and Jez Humble

Live Classrooms:

- Course Lectures: **Tuesday, February 10 at 9 PM ET**
- Assignments & Quizzes: **Thursday, February 12 at 9 PM ET**

Module 6 Study Guide and Deliverables

(February 17 – February 23, 2026)

Module Topics:

- Process Architecture
- Process Improvement
- Course review in preparation for Final Exam

Readings References:

- Module 6 online content
- CMMI, version 1.3
- ISO 9001 2008

Assessments:

Quiz 6: Due **Tuesday, February 24 at 6:00 AM ET**

- Assignment 2: Test an App

Assignments:

- Assignment 1: Learning Platform
- Assignment 2: Embeddings

All Assignments due **Tuesday, February 24 at 6:00 AM ET**

Team Project:

- Schedule online final presentation of project team
- Term Project deliverables due **Tuesday, February 24 at 6:00 AM ET**
    - Develop test cases in a standard format

  - Reduce data-driven combinations using Allpairs
- Term Project Final Report due **Wednesday, February 25 at 6:00 AM ET**

Class Discussions *:

- Adapt & Adopt – outline a path toward an effective institutionalization of a new process

Live Classrooms:

- Course Lectures: **Tuesday, February 17 at 9 PM ET**
- Assignments & Quizzes: **Thursday, February 19 at 9 PM ET**
- Preparation for the Final Exam – Q & A: **Tuesday, February 24 at 9 PM ET**

Course Evaluation:

Please complete the course evaluation once you receive an email or Blackboard notification indicating the evaluation is open. Your feedback is important to MET, as it helps us make improvements to the program and the course for future students.

Final Exam Details

The Final Exam is a proctored exam available from **Wednesday, February 25, 2026, at 6:00 AM ET to Saturday, February 28, 2026, at 11:59 PM ET**.

The Computer Science department requires that all final exams be administered using an online proctoring service that you will access via your course in Blackboard. In order to take the exam, you are required to have a working webcam and computer that meets the exam proctoring service's system requirements. A detailed list of those requirements can be found on the How to Schedule page. Detailed instructions regarding your proctored exam will be forthcoming from the Assessment Administrator. **You will be responsible for scheduling your own appointment within the defined exam window.**

The Final Exam will be close book/notes and is accessible only during the final exam period. You can access it from the Assessments section of the course. Your proctor will enter the password to start the exam.

Final Exam Duration: **3 hours**

Grading Information

The course is divided into modules, each of which begins on a Tuesday and ends on the following Monday. Please check the Study Guide in the syllabus for Live Classroom dates and specific due dates for deliverables.

Grade Weighting & Grade Conversion

The following table summarizes the four kinds of graded items along with their weight. These are four avenues for you to learn. These are four angles to illuminate the course content.

Graded Items	Percentage of Grade
Assignments	20%
Quizzes	20%
Term Project	30%
Final Exam	30%
Total	100%

The following table details the translation on a numerical grade to a letter grade.

Letter Grade	Numerical Grade
A	>=95 – <=100
A-	>=90 - <95
B+	>=85 - <90
B	>=80 - <85
B-	>=77 - <80
C+	>=74 - <77
C	>=70 - <74
C-	>=65 - <70

Class Engagement

- **Class Discussions:** A large part of the overall learning experience is gained through discussions and participation amongst the class. The intent here is to encourage meaningful participation during live sessions and through the discussion board. A typical lecture consists of two parts. The first part covers new material. The second part details each assignment and every quiz that covers new material. Class discussions are optional, but students are encouraged to post comments toward several predefined topics. All posts, along with an abundance of references, are propagated from previous classes to supplement the body of knowledge for the whole course.

- **Peer Reviews:** All in-process artifacts (e.g., requirements and test cases), as well as the final deliverable (code), shall be peer reviewed. Providing constructive and concise comments is a part of the defined process elaborated in the class. Each student actively participates in peer reviews, submits a predefined deliverable and incorporates changes suggested by peers.

- **Attendance Policy:** Attendance is **optional** but encouraged at Live Classroom sessions. You are responsible for all material discussed in class.

Graded Items

- **Quizzes:** Quizzes in this course are an important method for learning new material. Quizzes complement other methods, assignments, discussions, and your term project. Quizzes offer a slightly different path and angle toward the same course concepts. You will have multiple attempts on each quiz and the mean score will go toward the grade. Yes & No answers bear no judgment. They are impersonal, which is quite different from assignments where, in many cases, there is no right or wrong answer.

- **Assignments:** Assignments encourage you to extrapolate from the course material and from your own experiences. Researching a topic and responding to questions about grey-area-scenarios will force you to think independently. You should timebox your research. These assignments illustrate key points, and exploring too many details is bound to deviate from the big picture focus.

- **Term Project:** The purpose of the term project is to follow key phases of software product development. It is a chance to pilot and adopt the best practices covered in the course. In today's environment, collaboration is imperative as it drives the overall effectiveness of a software project. Peer review is the critical activity that enables team members to collaborate through finding defects in each other's work products. After participating in a term project in a controlled environment, you will be able to apply your skills later to a real situation. You are encouraged to benefit from several dozen projects completed for previous classes.
    - Check out the detailed Description of Term Project on the course site. Regular project submissions correspond to the lecture material for that module. For example, the first module covers requirements. So, the deliverable for the second module is to write project requirements in Pivotal. It is advisable to start coding as early as possible. A project team is expected to meet at least weekly. There is a demo expected at a weekly meeting of a project team of a partially-working system with coded parts and prototypes – as early as second module. The last module is dedicated to testing and bug fixing; hence coding should be pretty much done by the module that is previous to last.
    - Here is the testimony from a student.
      "I recently completed my Master's degree in 2017, and have very fond memories of CS633 – my final class before graduation. As a Principal Software Developer at Liberty Mutual, CS633 covers a plethora of topics relevant to my everyday work, providing me with the best-practice tools I need to succeed. Most specifically, the term project, which very accurately simulates a real-life product life-cycle, was exceptionally rewarding. My team and I were able to build a cloud-based, production-ready application, all while fine-tuning and

> reinforcing the learning that coincided in the classroom sessions. I most highly recommend this class, among the many others here at BU Master of Science in Computer Information Systems!"

- **Proctored Final Exam:** You will have three hours to complete the final exam; there should be plenty of time. The final exam will be open-book, open-notes.
    - A combination of multiple choice and answers based on quizzes from course modules.
    - **Essays:** Short essays/responses which will focus on in-depth lecture discussions and on practical knowledge gained through the Term Project.

You should expect no surprises on your final exam, as it includes no questions that are not covered during the class.

The last question on your final exam is as follows: *One of the goals of this course is for you to adopt at your day job some techniques we covered in the class. Which of the best practices from the course do you plan to adopt?*

Here is one of the responses.

> "While there are many elements of this course that I expect to directly apply to my current job, there are two that stand out more than others. Those two are peer reviews and estimation techniques. My company performs peer reviews for functional specifications as well as technical designs, but we do not conduct code reviews. I think if we were to implement a static analysis of team code, we would detect potential defects sooner. We would also have a chance to suggest more uniform coding best practices so there would be a greater consistency of code among developers. It would also aid in cross-training, in which my team is severely lacking."

Here is another response.

> The tool evaluation matrix and magic quadrant will be extremely valuable to me in the future as I find that the organization I work for tends to acquire a multitude of tools which do the same exact thing and it needs to be revised in order to standardize their use to become more efficient and save money at the same time. Thank you for the toolkit you have provided throughout the semester!

Ungraded Items

- **Ungraded Discussion Forums:** There are ungraded discussion forums throughout the course. You are encouraged to share your knowledge and learn from your peers. Discussions forums are provided for your benefit. Some discussion forums involve the teaching team members; others are among students.

- **Live Classroom Sessions:** Live Classroom sessions will be offered during this course in Modules 1 through 6. In each module, students have a Live Classroom session with the instructor and another Live Classroom (or problem-solving session) with the facilitator. Days/times will be posted in the Study Guide and the "Announcements" area.
    - Your participation, while not mandatory, will be valuable to you and the class. To participate in the Live Classroom discussion, you will need to go to the "Live Classroom/Offices" area.
    - Live Classroom sessions will be recorded and archived for further viewing. You can go to the "Live Classroom Recordings" area to view the recordings.

Evaluation Criteria and Grading Rubric

In general, the explanation for letter grades is provided by the Boston University Office of the University Registrar.

	D	C–	C+	B–	B+	A
Clarity	Disorganized or hard to understand		Satisfactory, but some parts of the submission are disorganized or hard to understand	Generally organized and clear	Very clear, organized, and persuasive presentation of ideas and designs	Exceptionally clear, organized, and persuasive presentation of ideas and designs
Technical Soundness	Little technical understanding of, or insight into, material		Some technical understanding of material	Overall technical understanding of much material	Very good overall technical understanding of material, with some real depth	Excellent, deep technical understanding of material and its interrelationships
Thoroughness and Coverage	Hardly covers any of the major relevant issues		Covers some of the major relevant issues	Reasonable coverage of the major relevant areas	Thorough coverage of almost all of the major relevant issues	Exceptionally thorough coverage of all major relevant issues
Relevance	Mostly unfocused	Focus is off topic or on insubstantial or secondary issues	Only some content is meaningful and on topic	Most or all content is reasonably meaningful and on-topic	All content is reasonably meaningful and on-topic	All content is entirely relevant and meaningful
Utilization of Resources	No relevant use of notes, text(s), web, or tools; incorrect details or inapplicability		Some relevant use of notes, text(s), web, or tools, with mostly correct details and applicability	Fairly good use of notes, text(s), web, and tools, with correct details and applicability	Very good use of notes, text(s), web, and tools, with correct details and applicability	Excellent use of notes, text(s), web, and tools with entirely correct details and applicability

Policy for the Use of Generative AI

Students should learn how to use AI text generators and other AI-based assistive resources (collectively, AI tools) to enhance rather than damage their developing abilities as writers, coders, communicators, and thinkers.

When using Generative AI in coursework, students shall:

1. Give credit to AI tools whenever used, even if only to generate ideas rather than usable text or illustrations.
2. When using AI tools on assignments, add an appendix showing (a) the entire exchange, highlighting the most relevant sections; (b) a description of precisely which AI tools were used (e.g. ChatGPT private subscription version or DALL-E free version), (c) an explanation of how the AI tools were used (e.g. to generate ideas, turns of phrase, elements of text, long stretches of text, lines of argument, pieces of evidence, maps of conceptual territory, illustrations of key concepts, etc.); (d) an account of why AI tools were used (e.g. to save time, to surmount writer's block, to stimulate thinking, to handle mounting stress, to clarify prose, to translate text, to experiment for fun, etc.).
3. Not use AI tools during in-class examinations, or assignments, unless explicitly permitted and instructed.
4. Employ AI detection tools and originality checks prior to submission, ensuring that their submitted work is not mistakenly flagged.
5. Use AI tools wisely and intelligently, aiming to deepen understanding of subject matter and to support learning.

For more details, please see the Generative AI Assistance (GAIA) policy.

Course Policies and Academic Conduct

Academic Integrity: Plagiarism is the passing off of another's words or ideas as your own, and it is a serious academic offense. Plagiarism and cheating also defeat the purpose of getting an education. Plagiarism and cheating cases will be handled in accordance with the disciplinary procedures described in the College of Arts and Sciences Academic Conduct Code. You are expected to know and abide by the code, which can be read online: Academic Conduct Code. Penalties range from failing an assignment or course (first offense) to suspension or expulsion from BU. If in doubt, cite your source. If you have any questions about academic integrity, please ask your instructor.

Incidents of academic misconduct will be reported to the Academic Conduct Committee (ACC). The ACC may suspend/expel students found guilty of misconduct.

Important Message on Final Exams

Dear Boston University Computer Science Online Student,

As part of our ongoing efforts to maintain the high academic standard of all Boston University programs, including our online MSCIS degree program, the Computer Science Department at Boston University's Metropolitan College requires that each of the online courses includes a proctored final examination.

By requiring proctored finals, we are ensuring the excellence and fairness of our program. The final exam is administered online.

Specific information regarding final-exam scheduling will be provided approximately two weeks into the course. This early notification is being given so that you will have enough time to plan for where you will take the final exam.

I know that you recognize the value of your Boston University degree and that you will support the efforts of the University to maintain the highest standards in our online degree program.

Thank you very much for your support with this important issue.

Regards,

Professor Lou Chitkushev, Ph.D.
Associate Dean for Academic Affairs
Boston University Metropolitan College

Disability and Access Services

In accordance with university policy, every effort will be made to accommodate students with respect to speech, hearing, vision, or other disabilities. Any student who may need an accommodation for a documented disability should contact Disability and Access Services at 617-353-3658 or at access@bu.edu for review and approval of accommodation requests.

Once a student receives their accommodation letter, they must send it to their instructor and/or facilitator each semester. They must also send a copy to their Faculty & Student Support Administrator, who may need to update the course settings to ensure accommodation is in place. Accommodation cannot be implemented if the students do not send their letters.