


CS - 390T - Software Quality Engineering (Fall 2018)

01. UG New Omnibus Course (17-18)

General Course Information

****Read before you begin****

1. Please turn on the help text before starting this proposal by clicking on the  icon in the top right corner of the heading.

2. All fields with an * are required. You will not be able to launch the proposal without completing required fields.

NOTE: Omnibus courses may be offered only three times before a regular course must be proposed for the Catalog.

DUE DATES:

For Fall 2017: Received in the Dean's office February 6, 2017

For Spring 2018: Received in the Dean's office September 22, 2017

For Summer 2018: Received in the Dean's office January 26, 2018

For Fall 2018: Received in the Dean's office February 16, 2018

You can consult the [Procedural Calendar](#) on the ASA Website for the most up to date information on curriculum due dates

For Additional Assistance and Step-by-Step Instructions on completing this form, consult this document: [Omnibus Form Assistance](#)

Department*	Department of Mathematical and Computer Sciences	
Status:*	<input checked="" type="checkbox"/> Active-Hidden	
Prefix:*	CS	Course Number:*
		390T
Course Type:*		

Computer Science

Course Title (include Semester and date for course to run):* Software Quality Engineering (Fall 2018)

Transcript Course Title:* Software Quality Engineering

Is this a study abroad course?* No Yes

Equivalent/Crosslisted? Equivalent Crosslisted

List all equivalent courses:

List all crosslisted courses:

To receive Title IV financial aid funds, all institutions of higher education must comply with the federal definition of a credit hour. The Higher Learning Commission requires institutions to maintain policies and procedures for verifying compliance with this definition.

Federal Credit Hour Definition: A credit hour is an amount of work represented in intended learning outcomes and verified by evidence of student achievement that is an institutionally-established equivalency that reasonably approximates not less than:

(1) one hour of classroom or direct faculty instruction and a minimum of two hours of out-of-class student work each week for approximately fifteen weeks for one semester or trimester hour of credit, or ten to twelve weeks for one quarter hour of credit, or the equivalent amount of work over a different amount of time; or (2) at least an equivalent amount of work as required in paragraph (1) of this definition for other activities as established by an institution, including laboratory work, internships, practica, studio work, and other academic work leading toward to the award of credit hours. 34CFR 600.2 (11/1/2010)

Credit Hours:* 4

Distribution of Credit Hours* 4+0

Schedule Type:*

Lecture

Grade Mode:*

Letter

Face-to-Face or Equivalent Hours per course:

Consult the Appendices of the [Curriculum Manual](#) to determine the hours for the course (appendices begin on page 33 of the document)

Lecture: 60

Lab: 0

Internship: 0

Practicum: 0

Other:

Additional Student Work Hours per course:* 120

Specified repeatable course:* No Yes

If yes, number of credits/ repeats allowed

Prerequisite(s): CS2050 (grade of C- or above) or permission of instructor

Corequisite(s):

Prerequisite(s) and/or Corequisite(s): CS3250

Banner Enforced Prerequisite(s): CS2050

Minimum passing grade for Banner enforced prerequisite course(s): C-

Banner Enforced Corequisite(s):

Minimum passing grade for Banner enforced corequisite course(s):

Banner Enforced Prerequisite(s) and/or Corequisite(s): CS3250

Minimum Passing Grade for Banner Enforced Pre/Corequisites: C-

Registration restrictions (optional):

Level	Class
Program	Student attribute
Major	
Other Registration Restriction(s):	
Course Description:*	This course is for students wishing to obtain an understanding of software quality and the means to improve the quality of software products. It

addresses the broad body of knowledge of software quality across the life-cycle of software products. Topics afford a fundamental understanding of quality philosophies, principles, methods, tools, standards, organizational and team dynamics, professional ethics, and legal implications.

Required Reading and Other Materials will be equivalent to:*

International Organization for Standardization, *ISO/IEC 25010:2011, Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models*, <https://www.iso.org/standard/35733.html> (accessed 7 February 2018).

Langr, J., *Pragmatic Unit Testing in Java 8 with JUnit*, Pragmatic Bookshelf (2015)

Martin, R. C., *Clean Code: A Handbook of Agile Software Craftsmanship*, Prentice Hall (2008)

Myers, R. C., *Essential Test-Driven Development*, Addison-Wesley Professional (August 2018)

Specific, Measurable Student Behavioral Learning Objectives:*

- Evaluate quality-based risks associated with a software development project
- Determine appropriate software quality metrics
- Create effective test plans
- Evaluate test plan effectiveness
- Conduct software testing and quality assessment
- Apply automated testing frameworks, methods, and tools
- Identify tools for software quality assurance
- Judge appropriateness of software quality assurance tools
- Create artifacts through the use of software quality assurance tools
- Analyze software quality artifacts
- Construct well-justified rationales for determining the adequacy of software quality for product release
- Identify ethical issues arising from software quality evidence and stakeholder concerns
- Assess test coverage
- Evaluate testing effort return on investment

Detailed Outline of Course Content (Major Topics and Subtopics) or Outline of Field Experience/ Internship*

- I. Software Quality
 - A. Perspectives
 - B. Expectations
 - C. Attributes
 - D. Measurement
 - E. Standards (ISO-25010:2011)
 - F. Professional Ethics and Social Responsibility
 - D. Legal Ramifications
- II. Software Quality Assurance (SQA)
 - A. Validation
 - B. Verification
 - C. Defect Discovery
 - D. Process Improvement
- III. Software Quality Engineering (SQE)
 - A. Defect Prevention
 - B. Defect Reduction
 - C. Defect Containment
 - D. Process Improvement
 - E. SQE Certification
- IV. Software Testing
 - A. Goals
 - B. Attitudes
 - C. Strategies
 - D. Activities
 - E. Automation

Evaluation of Student Performance*


1. Assignments
2. Project artifacts
3. Reflections on learning activities
4. Contributions to the learning environment

Course Proposal Objective(s)

Learning Objectives

No Learning Objectives


Steps for CS - 390T - Software Quality Engineering (Fall 2018)

Originator		<i>Status: Approved</i>
Participants	Activity	
 Jody Paul 2/8/2018 2:51 PM	Required for Approval: 100% required Date Completed: 2/8/2018 2:51 PM Changes: No Comments: No	




Department Curriculum Committee Chair		<i>Status: Approved</i>
Participants	Activity	
 Clark Dollard 2/14/2018 5:01 PM	Required for Approval: 100% required Date Completed: 2/14/2018 5:01 PM Changes: Yes Comments: Yes	




Department Chair		<i>Status: Approved</i>
Participants	Activity	
 Lindsay Packer 2/15/2018 8:48 AM	Required for Approval: 100% required Date Completed: 2/15/2018 8:48 AM Changes: Yes Comments: No	



Associate Dean		<i>Status: Approved</i>

Participants	Activity
 Linda Lang-Peralta 2/20/2018 5:29 PM	Required for Approval: 100% required Date Completed: 2/20/2018 5:29 PM Changes: No Comments: No



Registrar's Office	Status: <i>Approved</i>
Participants	Activity
 Connie Sanders 2/23/2018 12:05 PM	Required for Approval: 100% required Date Completed: 2/23/2018 12:05 PM Changes: Yes Comments: Yes