

# Metropolitan State University of Denver

## Regular Course Syllabus

### CS 390T Software Quality Engineering Spring 2018

Status	completed
Hierarchy Entities	Department of Mathematical and Computer Sciences
Approval Process Name	01. UG New Omnibus Course (17-18)
Current Step	Registrar's Office
Originator	Jody Paul
Department	Department of Mathematical and Computer Sciences
Status:	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
Credit Hours:	4
Distribution of Credit Hours	4+0
Schedule Type:	Lecture
Grade Mode:	Letter
Lecture:	60
Lab:	0
Internship:	0
Practicum:	0
Other:	
Additional Student Work Hours per course:	120
Specified repeatable course:	No
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):	

# CS 390T

Banner Enforced Prerequisite(s) and/or Corequisite(s):	CS3250
Minimum Passing Grade for Banner Enforced Pre/Corequisites	C-
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:	<p>International Organization for Standardization, <i>ISO/IEC 25010:2011, Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models</i>, <a href="https://www.iso.org/standard/35733.html">https://www.iso.org/standard/35733.html</a> (accessed 7 February 2018).</p> <p>Langr, J., <i>Pragmatic Unit Testing in Java 8 with JUnit</i>, Pragmatic Bookshelf (2015)</p> <p>Martin, R. C., <i>Clean Code: A Handbook of Agile Software Craftsmanship</i>, Prentice Hall (2008)</p> <p>Myers, R. C., <i>Essential Test-Driven Development</i>, Addison-Wesley Professional (August 2018)</p>
Specific, Measurable Student Behavioral Learning Objectives:	<ol style="list-style-type: none"> <li>1. Evaluate quality-based risks associated with a software development project</li> <li>2. Determine appropriate software quality metrics</li> <li>3. Create effective test plans</li> <li>4. Evaluate test plan effectiveness</li> <li>5. Conduct software testing and quality assessment</li> <li>6. Apply automated testing frameworks, methods, and tools</li> <li>7. Identify tools for software quality assurance</li> <li>8. Judge appropriateness of software quality assurance tools</li> <li>9. Create artifacts through the use of software quality assurance tools</li> <li>10. Analyze software quality artifacts</li> <li>11. Construct well-justified rationales for determining the adequacy of software quality for product release</li> <li>12. Identify ethical issues arising from software quality evidence and stakeholder concerns</li> <li>13. Assess test coverage</li> <li>14. Evaluate testing effort return on investment</li> </ol>
Detailed Outline of Course Content (Major Topics and Subtopics)	<p>I. Software Quality</p> <ol style="list-style-type: none"> <li>A. Perspectives</li> <li>B. Expectations</li> <li>C. Attributes</li> </ol>

# CS 390T

<p>or Outline of Field Experience/ Internship</p>	<p>D. Measurement  E. Standards (ISO-25010:2011)  F. Professional Ethics and Social Responsibility  D. Legal Ramifications</p> <p>II. Software Quality Assurance (SQA)  A. Validation  B. Verification  C. Defect Discovery  D. Process Improvement</p> <p>III. Software Quality Engineering (SQE)  A. Defect Prevention  B. Defect Reduction  C. Defect Containment  D. Process Improvement  E. SQE Certification</p> <p>IV. Software Testing  A. Goals  B. Attitudes  C. Strategies  D. Activities  E. Automation</p>		
<p>Evaluation of Student Performance</p>	<p>1. Assignments  2. Project artifacts  3. Reflections on learning activities  4. Contributions to the learning environment</p>		
<p>Learning Objectives</p>			
<p>Steps</p>	<p><b>Decision</b></p>	<p><b>Date</b></p>	
<p>Originator</p>			
<p>Jody Paul</p>	<p>approve</p>	<p>02/08/2018 02:51PM</p>	
<p>Department Curriculum Committee Chair</p>			
<p>Clark Dollard</p>	<p>approve</p>	<p>02/14/2018 05:01PM</p>	
<p>Department Chair</p>			
<p>Lindsay Packer</p>	<p>approve</p>	<p>02/15/2018 08:48AM</p>	
<p>Associate Dean</p>			
<p>Linda Lang-Peralta</p>	<p>approve</p>	<p>02/20/2018 05:29PM</p>	
<p>Registrar's Office</p>			
<p>Connie Sanders</p>	<p>approve</p>	<p>02/23/2018 12:05PM</p>	