Metropolitan State University of Denver
Regular Course Syllabus

Fall 2016

CS - 3250 - Software Development Methods and Tools

Status	completed
Tracking:	LAS 1617-42
Department	Mathematical and Computer Sciences, Department of
Status:	Active-Visible
Prefix:	CS
Course Number:	3250
Course Type:	Computer Science
Course Title:	Software Development Methods and Tools
Transcript Course Title:	SW Dev Methods and Tools
Check All That Apply:	Required for Major
Resource Implication Narrative	As those currently teaching CS 4250 will typically teach this course, no new resources are required.
Justification for Proposal	This course is based on the faculty's observations of preparedness for upper-division classes. Our belief is that our students need more software engineering tools and techniques in order to be successful in upper-division courses as those courses require more complex software. Our current introductory courses do not contain enough software engineering to adequately prepare our students.
Credit Hours:	4
Schedule Type:	Lecture
Grade Mode:	Letter
Lecture:	60
Lab:	
Internship:	
Practicum:	
Other:	
Additional Student Work Hours per course:	120
Variable topics umbrella course:	No
If yes, number of credits/ repeats allowed	
Specified repeatable course:	No
If yes, number of credits/ repeats allowed	
Prerequisite(s):	CS 2050, ENG 1020, and CAS 1010 with a grade of "C" or better.
Corequisite(s):	
Prerequisite(s) and/or Corequisite(s):	
Banner Prerequisite(s):	CS 2050, ENG 1020, and CAS 1010 with a grade of "C" or better.
Banner Corequisite(s):	
Banner Prerequisite(s) and/or Corequisite(s):	
Level	Undergraduate
Class	

Program/Major	
Student attribute	
Catalog Course Description:	This course introduces the basics of large-scale software development. As software size increases, so does the need to use appropriate tools and development techniques. The phases of traditional software development and several current software development lifecycles are introduced. The use of object-oriented techniques for large projects is covered. Creating appropriate and sufficient tests for test-driven and behavior-driven development is discussed. Students learn how to analyze their programs to detect errors and increase performance. The various types of automation used in creating a product are introduced. Students learn about group dynamics and work on a significant project in groups.
Required Reading and Other Materials will be equivalent to:	• Beginning Software Engineering 1st Edition, Rod Stephens, ISBN-13: 978-1118969144, ISBN-10: 1118969146, 2015 • Software Engineering: A Practitioner's Approach 8th edition, Roger S. Pressman and Bruce Maxim, ISBN-13: 978-0078022128, ISBN-10: 0078022126, 2014 • Clean Code: A Handbook of Agile Software Craftsmanship 1st Edition, Robert C. Martin, ISBN-13: 978-0132350884, ISBN-10: 0132350882, 2008
Specific, Measurable Student Behavioral Learning Objectives:	1. Select an appropriate software development lifecycle for a project. 2. Compare the various agile software development lifecycles. 3. Evaluate different object-oriented techniques for problem solving. 4. Construct a testing environment that includes code coverage analysis. 5. Collect and analyze both static and dynamic properties for a program. 6. Organize a project group with appropriate roles for members. 7. Compare and contrast several software architectures.
Detailed Outline of Course Content (Major Topics and Subtopics) or Outline of Field Experience/ Internship	1. Software Development Lifecycle 1. Traditional 2. Current 2. Object-oriented software development 1. Inheritance 2. Interfaces 3. Iterators/generators 4. Polymorphism 5. Operator overloading 6. Lambdas 7. Duck typing 8. Regular expressions 3. Software configuration management 1. Revision control 2. Branching, merging 4. Fundamentals of design patterns 5. Test 1. Test-driven development 2. Behavioral-driven development 3. Coverage 4. Mocks, stubs, etc. 6. Program analysis 1. Static 2. Dynamic 7. Debugging 1. Tracing 2. Break points 3. Watches 4. Logging

	8. Automated build/test/deploy 9. Human factors in software development 1. Working in groups 2. Project management 3. Career management 4. Human-computer interaction 10. Widely-used software architectures 11. Social dimensions of software			
Evaluation of Student Performance	Required: 1. Active participation on a development team. 2. Oral and written team presentations. 3. A mid-term and final examination on the material covered in the course. Optional: quizzes, classwork, homework.			
Learning Objectives				
Distribution of Credit Hours	4 (4+0)			
Steps	**Edits**	**Decision**	**Date**	
Originator				
Steve Beaty	6	approve	09/30/2016 03:40PM	
Department Curriculum Committee Chair				
Clark Dollard	0	approve	10/03/2016 02:13PM	
Department Chair				
Lindsay Packer	4	approve	10/05/2016 02:01PM	
Dean's Office Tracking Assignment				
Kelsey Smith	1	approve	10/06/2016 09:17AM	
Substantive College Level				
Linda Lang-Peralta	0	approve	12/19/2016 04:50PM	
Mona Mocanasu	1	approve	12/15/2016 01:39PM	
Steve Beaty	2	approve	12/15/2016 01:21PM	
AVP Academic and Student Affairs				
Bernice Harris	1	approve	12/21/2016 08:37AM	