Metropolitan State University of Denver
Regular Course Syllabus

CS 4260 Software Engineering Practices Spring 2017

Status	completed
Approval Process Name	22. UG Senior Experience Course Modification (17-18)
Department	Mathematical and Computer Sciences, Department of
Status:	Active-Visible
Prefix:	CS
Course Number:	4260
Course Type:	Computer Science
Course Title:	Software Engineering Practices
Transcript Course Title:	Software Engineering Practices
Equivalent/ Crosslisted?	
List all equivalent courses:	
List all crosslisted courses:	
Check All That Apply:	Required for Major
Credit Hours:	4
Schedule Type:	Lecture
Grade Mode:	Letter
Lecture:	60
Lab:	
Internship:	
Practicum:	
Other:	
Additional Student Work Hours per course:	120
Specified repeatable course:	No
If yes, number of credits/ repeats allowed	
Prerequisite(s):	Senior Standing, CAS 1010, CS 3240, CS 3250, CS 3210, CS 4250 with grades of "C" or better, additional 8 upper division CS credits
Corequisite(s):	
Prerequisite(s) and/or Corequisite(s):	
Banner Prerequisite(s):	CAS 1010, CS 4250
Minimum Passing Grade for Banner Enforced Pre-requisite(s):	C
Banner Corequisite(s):	
Minimum Passing Grade for Banner Enforced Co-requisite(s):	
Banner Prerequisite(s) and/or Corequisite(s):	
Minimum Passing Grade for Banner Enforced Pre/Co-Requisite(s):	
Level	

Class	Senior
Program	
Student attribute	
Major	
Other Registration Restrictions	
Catalog Course Description:	This course continues the software engineering emphasis of CS 4250, Software Engineering Principles. Students work in teams and apply software engineering principles and best practices to the development of real-world projects for which the quality of the software products is significant. Some projects have been solicited from external constituents (industry, non-profit organizations, etc.) with progress evaluated in conjunction with constituent representatives.
Required Reading and Other Materials will be equivalent to:	Building Maintainable Software, by J. Visser, et al., O'Reilly [2016] New Programmer's Survival Manual, by J. D. Carter, Pragmatic Bookshelf [2011]
Specific, Measurable Student Behavioral Learning Objectives:	Upon completion of this course the student should be able to 1. Engage in, combine, interpret, and assess activities associated with the software development process as it applies to a real-world problem. 2. Interact effectively with stakeholders in the development of software artifacts, through the use of skills in construction, demonstration, design, estimation, evaluation, explanation, summarization, and synthesis. 3. Perform effectively as a member of a software development team, including appraisals, recommendations, critiques, analysis, scheduling, planning, estimating, and evaluation. 4. Identify and analyze the problems that arise during software development and suggest mitigations and remedies that effectively address or resolve the problems in a satisfactory manner; drawing upon cognitive skills for adaptation, discussion, improvement, proposition, measurement, recommendation, judgement, and prioritization.
Detailed Outline of Course Content (Major Topics and Subtopics) or Outline of Field Experience/ Internship	1. Feasibility Analysis 2. Negotiating with Stakeholders 3. Incremental Development and Delivery 4. Validation and Verification of Software Artifacts 5. Requirements Engineering 6. Architecture and Design 7. Unit and Integration Testing 8. Implementation
Evaluation of Student Performance	1. Participation on the development team as evaluated by team members, the stakeholder representative(s), and the instructor. (Such participation means active engagement as a member of a team in the performance of software engineering appropriate to a specific project. This includes "team" and "development" aspects. For example, the "development" might include requirements elicitation if that is meaningful in the project context . Given the broad range of potential projects and specific objectives of each, it is the responsibility of the instructor to determine what is appropriate to the context.) 2. Oral and written team presentations.

Learning Objectives	
Distribution of Credit Hours	(4 + 0)
Senior Experience Designation	
The Senior Experience must allow students to:	
1. Synthesize learning through critical analysis and logical thinking	
Explain:	Students must perform critical analysis and logical thinking to define the client's problem and determine appropriate software solutions. This is especially challenging because most clients do not have a clear understanding of the software system that they need. Additional challenges are often present because the client may not fully understand the problem for which they are seeking a solution. The students in CS 4260 (Software Engineering Practices) must then synthesize the knowledge they have gained in many of their previous CS courses, technical writing course (COM 2610), and computer ethics course (PHI 3370) to develop a software system that meets the user's needs.
2. Apply theoretical constructs to practical applications.	
Explain:	The implementation of a software solution to the client's problem requires students to apply many of the theoretical constructs to fully functional software. Students are expected to use appropriate data structures and algorithms in the software while putting into practice good software engineering principles.
3. Critique philosophical tenets and current practices.	
Explain:	Students must critique the client's needs and technology and computer science capabilities and determine the appropriate software engineering practices that are likely to result in a successful project. Software engineering is dynamic with many different current practices that can be successful depending on the situation. Deciding on the best practices requires critical analysis, logical thinking, and critiquing of the entire range of software engineering principles.
4. Integrate and refine oral and/or written communication skills.	
Explain:	Determining the real client problem and an appropriate software solution requires students to do more written and oral communication skills of the highest quality than they have encountered in any of their previous course work. One of the greatest sources of difficulty in real-world software development is poor communications and misunderstandings. Students are required to integrate and refine oral and/or written communication skills to successfully complete the project.
5. Verify their expertise.	
Explain:	Development of a software system that meets the client's needs, is delivered on time, and within budget provides students with the awareness that they can successfully apply nearly all of their prior coursework. As part of the summative evaluation, students are asked to reflect upon their total experience including their strengths and areas for improvement. The result is that students become more confident and develop a greater sense of their capabilities for solving the problems they will face in their future careers.

Steps	**Decision**	**Date**	
Originator			
Jody Paul	approve	03/16/2017 09:47PM	
Department Curriculum Committee Chair			
Clark Dollard	approve	03/17/2017 03:45PM	
Department Chair			
Lindsay Packer	approve	06/05/2017 02:25PM	
Dean's Office Tracking Assignment			
Kelsey Smith	approve	06/06/2017 12:18PM	
Senior Experience Approval			
Jody Paul	approve	06/20/2017 05:15PM	
Linda Lang-Peralta	None		
Mona Mocanasu	None		
Zsuzsa Balogh	None		
Jean-Francois Duclos	None		
Erica Buckland	restart	08/03/2017 10:56AM	
Senior Experience Approval			
Jean-Francois Duclos	approve	12/05/2017 03:24PM	
Jody Paul	approve	11/28/2017 09:33PM	
Linda Lang-Peralta	approve	12/01/2017 03:43PM	
Mona Mocanasu	approve	12/01/2017 01:54PM	
Faculty Senate President			
Matthew Makley	approve	12/18/2017 12:07PM	
Curriculum Manager			
Erica Buckland	approve	01/04/2018 10:25AM	
AVP Academic and Student Affairs			
Chad Harris	approve	01/16/2018 09:24AM	
Registrar's Office			
Jeremy Coleman	approve	01/18/2018 05:28PM	