

**Master Syllabus**  
**Course: CIS 499**  
**Cluster Requirement: USC 5A and 5B**

**Course Overview:**

Part 2 of two-semester sequence that require team development of a software product for a specific customer. This capstone project utilizes and integrates knowledge and skills acquired through study of software engineering. Provides hands-on experience with large-scale problem from conception to implementation of the solution. Is conducted in a framework of well-defined low-ceremony software process. The lecture covers software engineering models, quality management, risk management, and provides introduction to software processes. Technical, logistical, and social issues associated with software development are addressed.

**University Studies Course Rationale:**

This capstone project utilizes and integrates knowledge and skills acquired through study of software engineering, technical communication, and mathematical formulaic reasoning. Depending on the nature of the project specifically, student are required to draw on knowledge from various sciences (including, but not limited to, physics, biology, chemistry, cognitive sciences, sociology and anthropology), auditory arts (sound/music), and visual arts. Application of these skills, with approval from the customer, becomes integral during this semester. Students meet with the customer at regular intervals, making sure requirements and customer expectations are met. Feedback from end-users beyond the scope of the customer (customer is not necessarily the end-user of the software) is also integrated. Experience with large-scale problem from conception to implementation of the solution is finalized. In summary, students combine a vast major of the skills and knowledge acquired over their undergraduate career, culminating in a software deliverable capstone project suited to the client and user's needs.

**Learning Outcomes:**

Course-Specific Learning Outcomes:

1. Use current techniques, skills, and tools necessary for computing practices
2. Apply design and development principles in the construction of software systems of varying complexity
3. Analyze the local and global impact of computing on individuals, organizations and society
4. Develop software which meets the needs of a real customer
5. Analyze a problem, and identify and define the computing requirements appropriate to its solution
6. Function effectively in teams to accomplish a common goal

University Studies Learning Outcomes 5A:

1. Use current techniques, skills, and tools necessary for computing practices  
Maps to USC 5A1; Explanation: Synthesize the knowledge and skills gained within major CIS courses, independently complete a research-based project and integrate the results of both in an open-ended computer science based project.
2. Conduct technical reviews and presentations of project's artifacts to small, medium and large size audiences

- Maps to USC 5A2; Explanation: Integrate knowledge and principles from the computer science field of study with those of the broader University Studies curriculum.
3. Analyze the local and global impact of computing on individuals, organizations and society  
Maps to USC 5A2; Explanation: Integrate knowledge and principles from the computer science field of study with those of the broader University Studies curriculum.
  4. Develop software meeting the needs of a real customer  
Maps to USC 5A3; Explanation: Demonstrate advanced information literacy skills by selecting, evaluating, integrating and documenting information gathered from multiple sources, including project customer/sponsor, into discipline-specific writing.
  5. Demonstrate an ability to communicate effectively with a range of audiences  
Maps to USC 5A4; Explanation: Communicate effectively, both orally and in writing, the results of the project or experience.

University Studies Learning Outcomes 5B:

6. Demonstrate understanding of responsibilities and the professional etiquette of software engineer (as an individual and a team) in interaction with the customer in software development process  
Maps to USC 5B1; Identify the needs and resources of the communities (customer and end-users) to which they belong.
7. Design, implement and evaluate a computer-based system, process, component, or program to meet desired needs of the customer and end-users.  
Maps to USC 5B1; Identify the needs and resources of the communities (customer and end-users) to which they belong.
8. Demonstrate understanding of contemporary issues of software engineering  
Maps to USC 5B2; Apply knowledge and skills gained through academic study to real problems and/or opportunities within their communities (customer and end-users).
9. Recognize the need for, and demonstrate an ability to engage in continuing professional development  
Maps to USC 5B3; Describe the connections between learning on campus and the issues and needs of broader academic, professional or civic communities.
10. Demonstrate an ability to communicate effectively with a range of audiences  
Maps to USC 5B4; Articulate the value of engagement to other members of their communities
11. Conduct technical reviews and presentations of project's artifacts to small, medium and large size audiences  
Maps to USC 5B4; Articulate the value of engagement to other members of their communities

**Examples of Texts and/or Assigned Readings:**

1. Pressman, R., 2014, Software Engineering - A Practitioner's Approach, McGraw-Hill
2. T. Clear, M. Goldweber, P. Leiding, K. Scott, F. Young, 2001, Resources for Instructors of Capstone Courses in Computing, ACM SIGCSE Bulletin, Vol. 33, Issue 4, pp. 93-113
3. P. Kruchten, 2000, The Rational Unified Process- an Introduction, Addison-Wesley
4. P.Kroll, P. Kruchen, 2003, The Rational Unified Process Made Easy, Addison-Wesley
5. R.Wysocki, 2003, Effective Project Management, Wiley Jacobson, G. Booch, J. Rumbauch, 1999, The Unified Software Development Process, Addison-Wesley

**Example Learning Activities and Assignments:**

Grade distribution: Project 40% Exams 40% Final Presentation 20%

Vision Document: (measurable for USC 5A2) Integrating knowledge and principles from the computer science field, including analysis of the domain where product will be used, how will it be used, how it will be incorporated into the organization/entity of the customer, how it is going to benefit the users, what problem is it going to solve, what are the alternatives, difference from other products or current practices, and general impact effects (at the broadest level). Specifically students will engage their knowledge from:

- Foundations of engagement (ENL 101/ENL 102 writing skills, ENL 266 technical communications, MTH 153/154) to effectively create the document (formulaically accurate) to the specification of non-technical customer expectations.
- The Natural World to set forth a road map that will result in the product being create in an ethically sound process. (CIS 381)
- The Cultural and Social World to interpret the needs of the customer and user with respect the environment in which the project will be deployed.

Project Reviews (2): Each team must submit and present their project status reports. During these presentation, teams will undergo critical scrutiny to improve upon their projects and civic engagement with the customer and end-users.

Final Presentation: Upon the completion of the semester and project, the teams will formally present their project in front of the academic and public communities.

Final Capstone Project: Utilizes and integrates knowledge and skills acquired through study of software engineering. Provides hands-on experience with large-scale problem from conception to implementation of the solution. Is conducted in a framework of well-defined low-ceremony software process. The lecture covers software engineering models, quality management, risk management, and provides introduction to software processes. Technical, logistical, and social issues associated with software development are addressed.

Semester 2 project submission includes a culmination of the following documents:

1. All documentation associated with design, implementation, and use of the software product.
2. Submission of the software product to the customer and computer science department.
3. Deployment of the software product.

#### Outcome Map:

Univ St Learning Outcome	Teaching and Learning Activities	Student Work Products
5A1	Lecture	Capstone Project; Exam
5A2	Case Studies	Vision Document
5A3	Lecture	Exam
5A4	Peer and Faculty Reviews	Project Review; Project Presentation
5B1	Case Studies; Lecture	Vision Document Revisions
5B2	Faculty Supervised Customer Meetings	Mock Deployment
5B3	Case Studies	Vision Document Revisions
5B4	Peer and Faculty Reviews	Project Review; Project Presentation

#### Sample Course Outline:

Learning Module	Subject Topic
1	Software Process Models
2	Software Process Management
3	Software Quality Management
4	Software Process Improvement

**Examples of Texts and/or Assigned Readings:**

6. Pressman, R., 2014, Software Engineering - A Practitioner's Approach, McGraw-Hill
7. T. Clear, M. Goldweber, P. Leiding, K. Scott, F. Young, 2001, Resources for Instructors of Capstone Courses in Computing, ACM SIGCSE Bulletin, Vol. 33, Issue 4, pp. 93-113
8. P. Kruchten, 2000, The Rational Unified Process- an Introduction, Addison-Wesley
9. P.Kroll, P. Kruchen, 2003, The Rational Unified Process Made Easy, Addison-Wesley
10. R.Wysocki, 2003, Effective Project Management, Wiley Jacobson, G. Booch, J. Rumbauch, 1999, The Unified Software Development Process, Addison-Wesley

**Example Learning Activities and Assignments:**

Grade distribution: Project 60% Exams 40%

Semester 1 project submission includes a culmination of the following documents:

4. Risk assessments
5. Rational Unified Process plan with milestones
6. System requirements
7. Customer Expectations and Requirements
8. Design and Analysis Models
9. Software Engineering Models

**Sample Course Outline:**

Learning Module	Subject Topic
1	Risk Management
2	Software Engineering Practices
3	System Engineering
4	Requirements Engineering
5	Analysis Modelling
6	Design Engineering
7	Architectural Engineering
8	Component Engineering
9	Interface Design
10	Software Testing