Objective

As a member of a software engineering team, you will likely be required to analyze complex software engineering problems; identify a solution to the problem; define a process to deliver the solution; provide analysis and design models of the solution; develop and document the solution; and manage software engineering activities. This project simulates such a request, while addressing all course objectives in that it requires you to:

1. Demonstrate Breadth and Depth in Defining, Analyzing, and Solving Complex Software Engineering Problems: After successfully completing this course, you should be able to demonstrate a basic capability to apply object-oriented design techniques and project methods to solve complex software engineering problems. Complex software engineering problems include one or more of the following characteristics: involving wide-ranging or conflicting technical issues, having no obvious solution, addressing problems not encompassed by current standards and codes, involving diverse groups of stakeholders, including many component parts or sub-problems, involving multiple disciplines, or having significant consequences in a range of contexts.
2. Communicate Technical Aspects of the Solution for Complex Software Engineering Problems to a Technical Audience: After successfully completing this course, you should be able to demonstrate a basic ability to communicate regarding the application of object-oriented design techniques and project methods to a technical audience.

Task	Activities and/or Deliverables	References
Identify The Problem, The Product and The Process	Identify problem Identify system requirements Identify process model	Chapters 1 – 3
Create Analysis Model	Detail requirements Create analysis model(s)	Chapters 2 – 5
Create Design Model	Architecture design Component design UX design Patterns applied	Chapters 2 – 17
Manage Software Engineering Activities	Plan project Estimate timelines & milestones Analyze risk Quality Management Test Methods Design Reviews Security Design, Implementation, and Monitoring Configuration Management System Support Implementation planning Process Improvement	Chapters 2 and 18 – 24
System Development	Develop and integrate select components Document system	SSE 550

Grading

The grading for the software engineering methods capstone project is based on professional expectations that might be required of you if you were a member of a software engineering team.

	Total Pts.	Excellent	Acceptable	Unsatisfactory
Identify Problem, Product, and Process Clearly identify a complex software problem as a problem statement. Clearly identify approach to solving problem as a product to be engineered. Define product requirements. Define the process to implement the product. Chapters 1 – 3	20	A complex software engineering problem is clearly identified. Approach to solving problem is clearly identified as a product to be engineered. At least 90% of product requirements are clearly identified. The process to develop the product is clearly identified. 20 points	Submission clearly identifies some of but not all four required items OR: Problem statement lacks clarity. Approach to solve problem lacks clarity or is not defined as a product to be engineered. 70% of product requirements are clearly stated. A software engineering process model to engineer the product is not clearly identified. 13 – 19 points	Submission fails to demonstrate the four required items or: Problem statement is not a complex problem. Approach to solve problem has significant gaps making the product ambiguous. Less than 70% of product requirements are clearly identified. A process model used to implement solution is not identified. 0 – 12 points
Create Analysis Model(s) System requirements are clearly detailed as use cases and functional and non-functional requirements. Analysis model(s) relevant to process model chosen to implement solution are complete and clear. Chapters 2 – 5	20	At least 90% of use cases and functional and non-functional requirements are clearly stated and are relevant to the problem and the engineering process model chosen. Required artifacts, including UML diagrams, are provided, as applicable for the process model chosen. 20 points	Submission fails to provide ALL required Analysis Model artifacts but provides some of the required artifacts per the process model chosen At least 75% of the use cases and functional and non-functional requirements are clearly articulated. 13 – 19 points	Submission fails to document ANY of the required Analysis Model artifacts and/or less than 75% of detailed requirements. 0 – 12 points

	Total Pts.	Excellent	Acceptable	Unsatisfactory
Create Design Model As required for process model selected: Architecture design; Component design; UX design; Patterns applied. Chapters 2 – 17	20	Submission provides Design Model artifacts required for the process model chosen, to include: Architecture design; Component design for at least 90% of the components selected for development); UX design; and patterns (with at least 90% coverage for components selected for development). 20 points	Submission fails to provide ALL required Design Model artifacts but provides at least 50% of the required artifacts for at least 75% of the components selected for development as required per the process model chosen. 13 – 19 points	Submission fails to document at least 50% of the Design Model artifacts required for the process model chosen for at least 75% of the components selected for development. 0 – 12 points

	Total Pts.	Excellent	Acceptable	Unsatisfactory
Manage Software Engineering Activities Plan project according to process model selected. Identify risks and risk mitigation plans. Identify timelines and milestones. Clearly articulate software quality management, including system tests. Conduct and document design reviews. Provide and document security design, implementation, and monitoring. Implement and document configuration management (CM). Plan for and document system support. Improve software engineering process. Chapters 2 and 18 – 24	15	Project is planned according to process model selected. Risks and risk mitigation plans are clearly identified. Implementation of risk mitigation is clearly articulated in documentation. Timelines and milestones are identified and relevant to the process chosen. Test plans and/or methods are documented and conducted. Design reviews are conducted and documented. Security design is complete and relevant to the solution. Security controls are implemented and monitored. Change control procedures are clear, and CM is implemented. System support plan, if applicable, is documented and complete. Documentation demonstrates attention and results from process improvement. 15 points	Submission demonstrates ANY of the following: No project plan or plan lacks sufficient details given process model selected. Risk analysis is incomplete. Risks are either not clearly identified, or risk mitigation plans are not clearly identified, if applicable. Project timelines and/or milestones are incomplete. At least 70% of requirements have test coverage. Security design lacks sufficient details. CM is implemented and used w/o clear procedures. System support plan lacks sufficient details. Some process improvement activities are documented. 10 – 14 points	Submission demonstrates lack of managing key software engineering activities. No or significantly lacking project plan as required for process chosen. No risks or risk mitigation plans documented, or those documented are significantly inadequate for the project, problem, or product. Lacks key timelines and milestones required for success. Insufficient tests or less than 70% of requirements are covered. No security plan or has major flaws. Lack of effective CM to control software baseline. No or significantly flawed support plan. No formal process improvement documented. 0 – 9 points

	Total Pts.	Excellent	Acceptable	Unsatisfactory
System Development / Implementation Develop and integrate software components required to demonstrate process model success. Document system for operators and maintainers.	15	All components needed to prove process model have been implemented and tested per testing plans with required operator and maintainer documentation. Solution expertly addresses all details of the problem. 15 points	Select components needed to prove process model have been implemented and tested per testing plans with needed operator and maintainer documentation. Solution addresses the problem, but with minor errors or a few logical flaws. 10 – 14 points	Submission does not include implemented components, fails to address the solution, or has significant logical flaws. 0 – 9 points
Professional student performance	10	Student actively participated in product as recognized by project management tools (e.g., submissions to configuration management for code development or documentation, project communications) 10 points	Student actively participated in project and contributed no less than 70% of the average of other student participation. 7 – 9 points	Student did not actively participate in project or contributed at least 30% less than the average of other student participation. 0 – 6 points