Project Defense Rubric Spring 2016

	1	2	3	4	Score
Definition/Scope Was the project well-defined with testable/verifiable features?	Features lacked clear definition throughout the project. Execution suffered resulting in haphazard development. Ended up with "something" like what was planned. Most features not clearly testable.	Features somewhat defined but many are not clearly testable. Features were only defined by execution, not plan.	Features defined relatively clearly. Most features testable. Might lack definition of what is specifically excluded.	Very clearly defined by both what was included and what was not to be part of the project. Features clearly testable.	
Design Is there evidence of good design? (clean interfaces, cohesion, modularity, reusability, maintainability, extensibility)	No. Interfaces are not clean, code is not readily reusable or maintainable. OR Little or no evidence of design (diagrams, design docs, etc.)	Some evidence. Perhaps lacking documentation or weakness in one or two design practices.	Good evidence of modularity, reusability and clean interfaces. Some lack of documentation or weakness in good design practice.	Very clear evidence of quality design shown by documentation, software organization, interface definition, and component implementation.	
Test Is there evidence of a clear test plan and execution?	Very little. Resulting code is very brittle.	A little. Mostly ad-hoc.	Some systematic, but also some ad-hoc.	Plan and framework for regression testing designed-in and used.	
Results Were features implemented as planned?	No. Students gave up and just omitted features without significant effort at revision or plan features of similar complexity/value.	Significantly reduced function, but present. OR Alternative feature(s) substituted but not in a timely fashion.	Somewhat reduced function. Attempt was made focus on important elements. OR Alternative feature implemented, though late, or lowering final project quality.	All features implemented and tested as planned. OR Alternative features (where required) were of similar or necessarily reduced complexity, but implemented in a timely fashion with no significant impact on project quality.	
Complexity Was the project complex in terms of components, interfaces, required tools/frameworks, algorithms?	Low. Does not significantly build on student knowledge or experience. Student did not demonstrate a project of reasonable complexity for a Senior capstone project.	Medium. Parts have some complexity. OR Complexity not high relative to size of team.	Medium high. Complex relative to size of team. A number of components requiring interface, integration, and test planning.	High. Multiple components requiring significant interface/integration design. High workload per team member.	
Learning How much new material had to be learned?	None or very little.	Some.	Several new tools, concepts or skills.	A new area entirely (e.g., language, framework, tools, library API, etc.)	

Note: Complexity and **Learning** dimensions are used as "grade modifiers" for assessing a project based on the other dimensions above since they contribute to the effort required. The other factor considered along with complexity and learning is the size of the development team.