

Rubric: Project Requirements and Use Cases

SOFTWARE ENGINEERING I (F20)

| | Excellent | Fair | Poor |
|---------------------|---|--|---|
| Hierarchy | (5) The requirements are logically ordered and nested when appropriate. | (3) Some requirements seem oddly placed. | (1) There is no nesting or it isn't clear why the requirements are ordered the way that they are. |
| Completeness | (5) The requirements are essentially complete. | (3) Some requirements are missing. | (1) Some key requirements are missing. |
| Consistency | (3) None of the requirements are conflicting. | (2) Two of the requirements conflict with each other. | (1) Multiple requirements are in conflict or are superfluous. |
| Testability | (3) It is objectively possible to determine whether the requirements have been met. | (2) Qualitative or vague performance measures are given. | (1) Many requirements are untestable. |
| Clarity | (3) Each requirement is clear and able to stand on its own. | (2) Some requirements require additional support. | (1) How multiple methods handle changes is incorrect or not described. |
| Details | (3) There are no unnecessary design or implementation details. | (2) There are a few unnecessary implementation details. | (1) There are many implementation details. |
| Use Cases | (5) All of the use cases are present. | (3) Some minor cases are missing. | (1) Major cases are missing. |
| Diagram | (5) The diagram is clearly drawn and labeled using proper notation. | (3) There are some mistakes with the notation. | (1) The diagram doesn't follow standardized notation at all. |
| Extensions | (3) <includes> and <extends> are used when appropriate. | (2) Some case could be made clearer with extensions. | (1) Extensions are not used. |
| Grammar | (5) The diagrams are relatively free from spelling and grammar errors. | (3) There are some spelling and grammar errors. | (1) There are significant spelling and grammar errors that noticeably detract from the diagrams. |