# **A checklist for grading the first software engineering report**

[Report format described here](http://www.ece.rutgers.edu/~marsic/Teaching/SE1/report1.html)

**Points will be deducted** from the [maximum possible score](http://www.ece.rutgers.edu/~marsic/Teaching/SE1/report1.html#GRADING) if the following issues are found in the report.

**Project Management**—typical issues: (max score: 10 points)

* project was apparently poorly managed, because the report is of substandard quality
* the perceived novelty of the project is low
* unfocused project with too many (unrelated) features that are haphazardly conceived and superficially designed
* poor and inconsistent writing, different styles, difficult to read and understand
* diagrams unclear, unintelligible, UML diagrams created using different tools
* lack of consistency and traceability between the requirements, use cases, user interface, and the domain model
* report is incomplete, not self-contained and no references are provided to additional relevant information
* missing page labeling (pagination), section headings, section numbering is messed up
* figures and tables are not labeled, missing captions, or not described and referenced in the text
* fonts and line width are different for different parts of the report
* the report is not following the prescribed format
* Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Section 1: **Customer Statement of Requirements**—typical issues: (max score: 9 points)

* the team obviously exhibits the lack of knowledge and understanding of their problem domain—desired system features are envisioned naively and superficially
* CSR poorly written, difficult to understand
* CSR written from programmer’s, not customer’s viewpoint; it describes the software-to-be rather than the problem that needs to be solvedt
* CSR vague, business policies that will be implemented are not mentioned
* glossary inadequate, missing terms, unclear definitions
* Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Section 2: **System Requirements**—typical issues: (max score: 6 points)

* requirements statements are not enumerated
* incomplete—more features or functions are mentioned in CSR than covered in the requirements (looks like you ignored the customer’s demands and came up with a highly narrowed down list of requirements!)
* overcomplete—some functional requirements appear from nowhere, are not derived from CSR or do not accurately capture the customer requirements
* improperly formulated: your requirements should define the *problem* to be solved by the software, *not* the software that solves it!
* the requirements are so abstract or generic that they do not define any concrete problem at all
* in other words, the requirements statements are not testable (unclear how to write test cases to test such statements)
* some of the requirements are very complex (unclear how they can be tested) and should be split into several simpler requirements
* non-functional requirements are not enumerated
* non-functional requirements are fuzzy and ambiguous—apparently not testable
* missing priorities for requirements
* the assigned priority weights are misplaced and will not make this project competitive with the rest of the class (e.g., too much attention paid to login and user registration, too little attention on the problem domain)
* a high or maximum priority is assigned to more than half of all requirements, which renders the prioritization effort useless
* Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Section 3: **Functional Requirements**—typical issues: (max score: 30 points)

* non-human stakeholders identified, or stakeholders are not defined
* actor definitions unclear; the difference between actors unclear—their roles appear to be same or similar—clarify the difference
* actors poorly named, such as “Yahoo!”, which happens to be your current provider—you should instead assign a conceptual name, derived from the service that this actor provides to make it substitutable with a similar system
* missing actors that inexplicably appear in later parts of the report
* UC's named not as verb phrases
* UC numbering not consistent throughout the report
* casual UC's do not list the associated actors; actor types (relative to UC's) are not defined
* use cases appear from nowhere—not derived from the system requirements
* confusion between <<include>> (or <<uses>>) and <<extend>>
* too many use cases are elaborated, but only superficially and none of them passes a reality check—none of them would be able to perform an actual task in a real system
* format for fully-dressed/detailed UC's not followed
* scenario steps are very coarse, fuzzy, simplistic, or apparently missing—it is impossible to assess whether the listed steps would allow the user to successfully accomplish the task, or whether the relevant requirements are satisfied
* no alternate scenarios considered (only the main success scenario shown); or alternative scenarios are very naive and simplistic;
* alternate scenarios should be broken into individual steps (instead of a one-sentence mention) and properly explained
* unclear where alternate scenarios branch from the main scenario, or whether and how they merge back
* formatting of fully-dressed UC's is inconsistent; e.g., switching between bullets, numbers, and prose
* detailed use cases are apparently duplicated and there is very little difference between the use cases—either the use cases are poorly identified or they are redundant;
* even if there is a need for several similar use cases, their differences should be highlighted and their existence justified—can these use cases be replaced with a single use case with subroutine use cases («include» or «extend»)?
* use cases describe what happens *inside* the system, instead of focusing on what happens on the *boundary* of the system (how the system behaves relative to the actors)
* terms specific to the problem domain are used in use cases, but they are not defined in the glossary
* evident contradictions of use cases with the requirements
* entry/exit conditions (or preconditions, postconditions) not stated or contradictory (e.g., being logged in is a precondition, but then the login is mentioned in the event flow)
* the focus of use cases is misplaced—instead of focusing on the most important functions of the system, they are focused on supporting activities, such as login, user registration, account management, viewing different tables, etc.
* UC diagram has UML notation problems
* use case diagram shows more or less UC's than mentioned in the text; UC's not shown by names in the diagram
* UC diagram not described in narrative
* UC diagram inconsistent with the use case descriptions
* UC diagram missing labels for UC relationships
* traceability matrix missing or inaccurate or no description provided; some use cases appeared out of nowhere and are not responding to the system requirements
* not clearly stated which system sequence diagram corresponds to which use case and scenario (main success scenario or alternative scenarios or both)
* system sequence diagrams inconsistent with the corresponding scenarios in use case description, unclear procedure
* Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Section 4: **User Interface**—typical issues: (max score: 15 points)

* no sketches of visual appearance provided for different screens
* some mockup interfaces unexplained or unclear description
* the navigation path through the screens is unclear or missing
* not described how each component of the user interface addresses the specific requirements (referred by their label);
* difficult to see how the user interface will support the execution of your use cases
* user effort estimation does not appear to represent a typical usage scenario
* worst-case scenarios not considered
* all of the estimations are vague and shown just as “variable keystrokes”—you should instead analyze the user effort with some sample typical input sequence
* terms specific to the problem domain are used in describing the user interface, but they are not defined in the glossary
* Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Section 5: **Domain Model**—typical issues: (max score: 25 points)

* the report shows a lack of understanding of the domain model
* there are fewer (or much fewer!) domain concepts than use cases or system requirements—this implies that concepts too coarse grained and are assigned too many responsibilities per concept
* unclear or not explained how the concepts/attributes/associations were derived
* poor choice of concepts and concept names, such as “login”, “buy”, “sell”, etc. (concept name should be a noun-phrase, such that it represents the *role* for a given concept)
* some concepts have unclear/undefined responsibility or disproportionally large responsibility or too many responsibilities
* missing specification of the protocol for communicating with other systems (used by your system but developed by 3rd party)
* traceability matrix missing or inaccurate or no description provided; some concepts appeared out of nowhere and are not responding to the use cases and elements of the user interface
* system operation contracts are mostly trivial (no effort was made to identify significant potential issues that need to be enforced by a contract)
* contracts are missing for some issues identified in detailed use cases that evidently must be enforced by contracts
* mathematical model missing, but apparently should be present
* terms specific to the problem domain are used in the domain model, but they are not defined in the glossary
* Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Section 6: **Plan of Work**—typical issues: (max score: 5 points)

* no timeline diagram, or the diagram is cluttered and difficult to read
* breakdown of responsibilities missing, or uneven (some students assumed disproportionate responsibilities), or fuzzy
* missing text description for plan of work
* Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Section 7: **References**—typical issues: (up to 5 *negative* points)

* references missing or incomplete
* only URL without a title shown
* some parts of text or figures or design ideas appear to be adopted from elsewhere, but citations are missing—for anything that you did not invent and is not part of general knowledge, a reference should be cited
* references not cited or otherwise mentioned in the main document
* Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_