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

[Report format described here](http://www.ece.rutgers.edu/~marsic/Teaching/SE1/report2.html)

**Points will be deducted** from the [maximum possible score](http://www.ece.rutgers.edu/~marsic/Teaching/SE1/report2.html#GRADING) if the following issues are found in the report.

**Project Management**—typical issues: (max score: 16 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
* it is clear that different sections were created by a different students who did not bother to make any explicit connections in their diagrams and text to other sections of the report
* 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
* the report is not following the prescribed format
* different diagramming tools used for different UML diagrams, but without explaining why this was necessary
* some diagrams are impossible to read; even worse, there is so much white space left on the margins around them
* your project is heavily relying on a third-party hardware or software platform, but you are not providing adequate information about that platform, which makes it is hard to read your report. You do not need to give tutorials on third-party systems, but at least say in your references where such tutorials can be found. Also, a brief comment (one sentence) on each concept or construct from third-party systems would help improve the readability of your report
* Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Section 1: **Interaction Diagrams**—typical issues: (max score: 30 points)

* the diagrams in your Report #2 seem to have been invented independently of all analysis that you did for Report #1 (see more details about [traceability](http://www.ece.rutgers.edu/~marsic/Teaching/SE1/report2.html#TRACING))
* it is not clear a particular UML sequence diagrams relate to one another, where they originate and where they end
* diagrams are given out of context, with none or inadequate description
* it is not explicity stated with which system method a particular sequence diagram originates—what is the initiating event?
* refer to a system sequence diagram from Report #1 and say which system method you're describing with your current interaction diagram
* concepts that you identified in your domain model (Report #1) are not used in your sequence diagrams; new concepts are introduced without explanation of how they relate to the domain model of Report #1
* some objects/classes are never explained, they just pop up out of nowhere; much guesswork is needed for reading your report
* the names of classes/objects/methods/variable are *unintuitive* and they are not defined or described
* the description of a UML diagram does not refer to specific elements of the diagram, so it is difficult to connect to the diagram that it supposedly describes
* the project uses the Web architectural style (client running in a browser and connecting to a server), but in sequence diagrams it is not clear where the client-side ends and where the server-side begins
* the project uses classes from a third-party, but the attribution is not clear
* Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Section 2: **Class Diagram and Interface Specification**—typical issues: (max score: 10 points)

* class diagram and method signatures are provided *without* text description of classes, methods and attributes—each class, method and attribute must be defined
* class diagram is split and shown on several pages to improve readability (which is fine), but the connection points are not indicated and it is unclear how diagrams connect to each other and how they form a single program
* multiple class diagrams are shown, but it is not explained how they relate to one another
* association links in the class diagram(s) are not shown (inheritance, aggregation, navigation, ...), inadequately labeled, or not described—provide some text to explain your diagrams
* traceability matrix missing or not described
* some classes cannot be traced back to a domain concept and explanation is missing as to why new classes are introduced
* some concepts from the domain model do not have corresponding classes in the class diagram, and there is no explanation why so
* Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Section 3: **System Architecture and System Design**—typical issues: (max score: 15(\*) points; \* = variable)

* there is no explanation of what each package is doing and why the classes are grouped in the given way; for example, *why* a three-tier layered architecture is appropriate for your system?
* you keep talking about a well-known concept, such as “Model” which is part of the [MVC framework](http://en.wikipedia.org/wiki/Model-view-controller), but you are not explaining how the general concept maps to your software and it is not possible to figure this out from your descriptions—explain how they fit in your specific context
* for example, what is the “Model” in your software? Is it one specific class or a set of classes? Which ones? How did you decide to declare the part of the “Model” starting from your domain model from Report #1?
* hardware boundaries are not clearly indicated, so it is not clear what is part of your own software versus third party systems
* Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Section 4: **Algorithms and Data Structures**)—typical issues: (max score: 4 points, if applicable)

* algorithms for mathematical models from Report #1 are missing
* algorithms used in your project are inadequately described—some things are simply not describable precisely in plain language—provide clear diagrams, classical flowcharts or better, use UML activity diagrams
* Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Section 5: **User Interface Design and Implementation**—typical issues: (max score: 11 points)

* none or little visible progress made compared to the user interface description from your Report #1
* no sketches of visual appearance provided for different screens
* some screens 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
* 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
* Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Section 6: **Design of Tests**—typical issues: (max score: 12 points)

* only acceptance tests for use cases are provided, but unit tests for classes are missing
* test coverage not analyzed or the analysis is inadequate
* Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Section 7: **Plan of Work**—typical issues: (max score: 2 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
* Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Section 8: **References**—typical issues: (up to 5 *negative* points)

* your references are totally inadequate; it is clear from your report that you have drawn on many sources, but you never mentioned any of them
* there is generally an issue of *attribution* in your report. Your reference list is extremely thin, but there seems to be so many great ideas that you are mentioning, and you do not offer any evidence that you invented all of them
* 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 by number or author’s name, or otherwise mentioned in the main document
* Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_