Student Projects Grading Sheet

CPSC-4175, Fall, 2017

November 28, 2017

1 Grading Rubric

Software Requirements Specification Consider the following questions. Are all requirements uniquely identified? Are all requirements cross referenced to previous SRS? Are requirements organized to maximize readability ? Are requirements unambiguous? Are requirements complete (as to the scope of the project)? Are requirements consistent? Are requirements verifiable? Are requirements traceable? Are requirement modifiable?

Design Artifacts Consider the following questions. Does the design reflect the architecture of the product? Does the design reflect the logical components of the product? Does the design reflect the data flow (inputs, outputs)? Does the design reflect the (dynamic) behavior of the product? Are the design artifacts clear and understandable? Are the design artifacts traceable to the requirements?

Implementation You do not have to run the code you are reviewing, but only conduct a visual inspection. Consider the following questions. Does the source follow a consistent coding standard? Does the source use understandable names (variables properties, functions, methods, objects, etc.)? Is the program organized so that it's easy for the reviewer to follow? Is the code appropriately commented? Are the coding structures traceable to the design artifacts? Does the code contain any obvious errors?

Testing You do not have to conduct tests of the code you are reviewing. Consider the following questions. Is the testing code automated? Do the test provide reasonable coverage of the functions of the software? For black box testing, does the testing code target the requirements? For the white box testing, does the code reasonably account for all of the possible paths through the code?

Documentation Documentation generally consists of two kinds: a reference manual for technical users, and a user guide for ordinary users. Consider the following questions. *For the reference manual:* A reference manual documents the classes, properties, and methods of the software, similar to the documentation produced by `javadoc`. With respect to all code artifacts, it contains a summary of each. With respect to methods and functions, it identifies the return type, the usage of the component, and the number and type of all parameters. *For the user guide:*A user guide explains using non-technical language the purpose of the software, how to invoke the software, and how to use the software.

Comments to students The project requires you to provide comments to your student peers in the following manner. For the two highest team grades and the two lowest team grades, write a comment explaining your grade. Email your comments to the team members and to me. Other than that, you comments are *non-public*. The only people who should see your comments are the team members and myself. Be fair, be honest, and be polite. Before you send your comments, please read them over from the view point of the recipients, and revise you comments if necessary.

Due date **Wednesday, December 6, 2017**. Failure to complete the team evaluations will result in a grade of I (incomplete). Assessments are due either by the time for the scheduled final examination if delivered by hand, or if delivered electronically by midnight of the day of the final examination.

2 Grade sheet

Name: _____ Date: _____

Instructions There are five assessment categories: (1) requirements analysis, (2) design, (3) implementation, (4) testing, and (5) documentation. Grades must be integers (20, 19, 18, etc.) The maximum grade for each category is 20. The minimum grade for each category is 0. You may use an integer grade only once in each column, that is, once you award a team a 20, you may not award any other team a 20. You must award every team a grade in every category except your own team. *Do not grade your own project.* Sum the grades for each team and enter the sum of the grades in the **Total** column.

Team	**Requirements Analysis**	**Design**	**Implementation**	**Testing**	**Documentation**	**Total**
Team Alpha `https://github.com/ohorace/Team-Alpha`						
Team Bravo `wright_kristen3@columbusstate.edu`						
Team Charlie `https://github.com/jtaylor2637/CPSC4175`						
Team Delta `https://github.com/stoksc/Settlers-of-definitely-not-Katan`						
Team Echo `https://github.com/kerrgavin/cpsc4175`						
Team Foxtrot `https://github.com/erdmann-casey/Brisket`						
Team Golf `https://github.com/wadkinsDaniel/CPSC4175_Project`						
Team Hotel `https://github.com/TehJay/CPSC4175`						
Team India `https://github.com/Cpruitt96/Software-Engineering-Project`						
Team Juliette `https://github.com/JsoftEng/CPSC4175/tree/master/TeamJuliette`						