CEN 4020: Software Engineering I
Florida State University

Spring 2020

Term Project Rules and Requirements

Overview:

During the rest of the semester, students will work in teams to specify requirements, design, implement, test, and document a software system. There will be **three increments** in which the team will make progress on the implementation of the system and on the required documentation. The *source code of the project, as well as the documentation* need to be hosted in a cloud-based version control system such as GitHub.

The three project increments are designated as the course assignments being used to assess the ABET/SMALC requirements. See the grading criteria below for information about this.

Each increment has a set of **deliverables**:

1) A **progress report,** which will describe in detail what was achieved in the current increment, the overall status of your project, any challenges or changes to the project that happened during the increment and the contribution of each team member to the project and all the deliverables required. Place a new Project Report document in your GitHub repository by the deadline of each increment.

   Complete the *Progress Report Template.docx* (found on Canvas) for each increment.

2) A **software requirements and design document** (see *RD Template.docx* found on Canvas), which will be updated and improved through the semester. Place this document in your GitHub repository and keep it up to date. We will consider the latest version by the deadline for the current increment.
   a. By the end of increment 1, you are expected to have a first version of the *functional requirements* (something that the system shall do*)* and *non-functional requirements* (any requirement referring to a property of the system, such as security, safety, software quality, performance, reliability, etc.) specified, the use cases involved in the current state of your project and a preliminary version of the class diagram or sequence diagram.
   b. By the end of increment 2, you are expected to have an updated and revised version of your requirements, use cases, class diagram or sequence diagram, as well as the textual descriptions of all the use cases in your use case diagram.
   c. By the end of increment 3, you are expected to have the most up-to-date version of the requirements, use cases, class diagram or sequence diagram for your project, as well as the most up-do-fate textual descriptions of all the use cases in your use case diagram.
3) An **implementation and testing document** (see *IT Template.docx* found on Canvas) describing the programming language used, the APIs, platforms, technologies, etc. used in your system, as well as the testing strategy used by your team to test the system. This will also be updated and improved between increments as needed. Place this document in your GitHub repository and keep it up to date. We will consider the latest version by the deadline for the current increment.
   a. For the first increment, only sections 1 and 2 of the document are required.
   b. For increment two and three, all sections are required.
4) For increment one and two, a **video** (**5 – 7 min**) and for increment three, a **presentation** (**5 -7 min**). Both the video and presentation need to describe the current state of the project, including:
   a. a general overview of the project

   b. a short description of the state of the project and what was accomplished during this increment
   c. a demo of the current project
   d. describe any change in scope of the project from the initial plan proposed in your project proposal and explain why the change occurred.
   e. *only for the first and second increment*, talk also about your plan for the next increment (what features will be completed in the next increment).

   Paste the link to the videos in the progress report for increment one and two (host them somewhere where we can access them). Place the presentation in the GitHub repository directly or paste the link in the progress report for increment 3.

5) **Teammate evaluation forms –** each student will submit their own on Canvas using the assignment created for this on Canvas for each increment (see *Teammate Evaluation.docx* found on Canvas)
6) **Source code** up-to-date in the GitHub repository (no separate document needs to be submitted for this on Canvas).

All team members need to contribute to ALL of the Items 1-4 and Item 6.

Each deliverable will be evaluated separately (each deliverable will have a grade associated with it). By the end of the course, each team will present a complete implementation of the proposed software system together with the complete version of the required documentation.

Deadlines:

- **Friday, March 6th**: ***End of first increment.*** **All deliverables due @ 11:59 pm.**
- **Friday, April 3rd**: ***End of second increment.*** **All deliverables due @ 11:59 pm**
- **Friday, April 24th**: ***End of third increment.*** **All deliverables due @ 11:59 pm**

Grading:

Note: all points will be added up and divided by the maximum possible points to obtain your final project grade. ***The grades for all deliverables besides Teammate evaluation forms will be computed based on their overall quality and the performance delivered by the group, as well as the individual contribution of a student to the deliverable.*** It is therefore **very important to document the contributions of each team member in detail in the progress reports** at the end of each increment.

A. **Increment 1 – contribution to and quality of progress report:** 20 points
B. **Increment 1 – contribution to and quality of RD document:** 60 points
C. **Increment 1 – contribution to and quality of IT document:** 40 points
D. **Increment 1 – contribution to and quality of the video:** 20 points
E. **Increment 1 – turn in teammate evaluation forms:** 5 points
F. **Increment 1 – contribution to and quality of the project** based on source code (features implemented, progress made): 100 points
G. **Increment 1 – quality of your own source code** (descriptive identifiers, good comments, good code structure and organization, clean and easy to understand code)**:** 10 points

H. **Increment 2 – contribution to and quality of progress report:** 20 points
I. **Increment 2 – contribution to and quality of RD document:** 60 points
J. **Increment 2 – contribution to and quality of IT document:** 40 points

K. **Increment 2 – contribution to and quality of the video:** 20 points
L. **Increment 2 – turn in teammate evaluation forms:** 5 points
M. **Increment 2 – contribution to and quality of the project** based on source code (features implemented, progress made): 100 points
N. **Increment 2 – quality of your own source code** (descriptive identifiers, good comments, good code structure and organization, clean and easy to understand code)**:** 10 points

O. **Increment 3 – contribution to and quality of progress report:** 20 points
P. **Increment 3 – contribution to and quality of RD document:** 60 points
Q. **Increment 3 – contribution to and quality of IT document:** 40 points
R. **Increment 3 – contribution to and quality of the presentation:** 20 points
S. **Increment 3 – turn in teammate evaluation forms:** 5 points
T. **Increment 3 – contribution to and quality of the project** based on source code (features implemented, progress made): 100 points
U. **Increment 3 – quality of your own source code** (descriptive identifiers, good comments, good code structure and organization, clean and easy to understand code)**:** 10 points

This assignment is designated as one of the course assignments being used to assess basic programming skills for ABET/SMALC requirements (see the syllabus for more details). In addition to the normal grading scales, each student's submission will be judged on several aspects on a scale of "Highly Effective", "Effective", or "Ineffective", as specified by ABET/SMALC outcome assessment procedures. A student's submission that earns at least 85% of the available points will count as “Highly Effective”, earning between 70-85% of the available points will count as "Effective", and getting less than 70% of the points will count as “Ineffective”.

In order to pass the class you must obtain at least 70% of the points (“Effective”) on EACH of the items below:

1. ***Teamwork***. This will measure a student’s ability to function effectively in a team to accomplish a common goal. The teams will work together on the requirements, design, and implementation of a software system. The contribution of each student to the team project will be graded individually. **The ABET grade for this aspect will be computed as the average of items A, D, F, H, K, M, O, R, T above (your grades for the contribution to the state of the project, video/presentation, and progress report).**

2. ***Software Requirements***. This will measure a student’s ability to identify and document the requirements and an initial design of a system. Each student’s contribution to the RD document will be individually assessed. **The ABET grade for this aspect will be computed as the average of items B, I, and P above (your grades for the RD documents).**

3. ***Software Implementation***. This will measure a student’s ability to apply design and development principles in the construction of a software systems of significant complexity. In particular, this will measure the ability to implement and evaluate a computer-based system to meet desired needs and the ability to apply development principles in the construction of a complex software system. **The ABET grade for this aspect will be computed as the average of items C, F, J, M, Q, and T above (your grades for the IT documents and state of the project).**