Review of the CEN4020 Software Engineering I Course

DATE: May 4th, 2018

Review by: Dr. David Gaitros, Dr. Sonia Haiduc, Dr. Adrian Nistor, and Dr. Jeremy Hatcher

Discussion:

1. **Are there any problems with course last year? If Yes, describe the problems.**

a. The current textbook has some problems, such as outdated material, missing details, unsupported claims, and can be difficult to follow at times by students.

b. The SRS writing requirement was usually done by students last minute, resulting in a low quality document.

c. Despite considerable time spent on explaining differences between functional and non-functional requirements, students still had problems grasping what non-functional requirements were and how to write them.

d. Some students were overwhelmed by the class project, as they had not taken some essential courses before, such as Data Structures, Databases, Mobile Programming, or Web Development. Also, since different students had taken different classes up until this point, it made it hard for them to work together in a team on the same project, leading to some students doing more work than others.

e. Due to the large amount of material that needs to be covered, the class felt rushed at times, with not enough time spent on some important subjects.

2. **Recommendations: Course committee recommendations to the current/future instructors.**

a. We recommend a change of textbook. The proposed textbook is:
Object Oriented and Classical Software Engineering, 8th Edition by Stephen R. Schach

b. We recommend making *COP 4530 Data Structures, Algorithms, and Generic Programming* a prerequisite for the class (right now it's a co-requisite). We believe this will also increase the probability of students taking other advanced courses, such as Databases, before taking CEN4020.

c. We recommend having mobile programming lectures, exercises, and assignments during recitations in the first few weeks of classes, in order to offer a common background for students in the class and help them get started earlier and more confidently on the class project.

d. We recommend reducing the list of topics covered, to ensure that there is enough time for the students to get a deep understanding of the following important topics: requirements engineering, object oriented design, design patterns, agile methodologies, testing, and open source software.

e. We recommend having class presentations about the project often, at least after each major iteration. Seeing the work done by the best groups in the first iterations seems to motivate the less productive groups to do better in the following iterations.

f. Hosting the project in a version control system and checking the contributions of each student to their group's project after each iteration may be useful, as it allows for personalized feedback and motivate some students for participating more in the project.

g. Using the Atlassian suite of free software tools for project management could help the students in the class during their group project. Atlassian provides solutions for issue tracking, collaboration, and configuration management. Many DoD programs have begun utilizing these tools for very large projects. The use of these tools would also be a big bonus on a student's résumé.