CMPSC 280
Principles of Software Development
Fall 2015

Laboratory Assignment Three: Team-Based Verification and Validation of Software

Introduction

Now that you are completely comfortable with the use of the Git version control repository, the Bitbucket site for managing these Git repositories—and the use of the Markdown language for the purpose of documenting your software—we will continue team-based implementation of a simple software system. In particular, your team will "receive" all of the deliverables that another team implemented in the last laboratory assignment. Then, it will be your team's responsibility to compare and contrast the deliverables that you created with those that the other team produced.

Next, you must carefully validate and verify both the deliverables that you received and those that you produced. For this assignment, these tasks will involve you finding and discussing mistakes in the requirements, design, and tutorial documents from your team and your "partner" team while also writing JUnit test cases for both the Java code that you implemented and the code created by your team. Members of your team who have questions about the deliverables that you received should direct them all to the maintainer from the team who shared their system with you.

Evaluating the Specification and Design of the Systems

Paying careful attention to both the phases of the software lifecycle and the roles of members of a development team, as explained in Sections 1.5 through 1.8 of the textbook, your team will interact with the customer to revisit the requirements of the program and then, as needed, re-design, re-implement, re-document, and fully test the system. The maintainer of your system is also still responsible for interacting with the developers in your partner team who will want to validate and verify your product. Teams with questions about these tasks should contact the course instructor.

Before you start working on this assignment, please carefully review the content at the end of Section 2.2 in SETP to learn more about the tasks associated with the validation and verification of software. Once all members of your team understand these tasks, you should pick one member who will be responsible for evaluating both the written deliverables that you have received from your partner team and the system that you have created as part of the last assignment. At a high level, you should begin to ask and objectively answer questions such as "what are the similarities and differences between our two systems?" and "which system is better and why do I think that it is better?" and "are these two systems fundamentally similar to or different from each other?"

Specifically, this member of the team should assess the requirements documents in an effort to determine if it is clearly written in a style that is precise, unambiguous, and free from errors in spelling and grammar. When this individual is reviewing the object-oriented design, it is important to fully answer relevant questions such as: "How many classes does this design use?" and "What will be the relationship between the classes?" and "What methods do the classes have?" and "What are the inputs and outputs of the methods?" and "Is the design easy to understand?" Again, this team member should ask and answer all of these questions about the design of both their own system and the one that was shared with them at the start of the laboratory assignment.

Verifying the Systems Through Testing

In the last assignment, the developer(s) and tester(s) took the requirements and design documents and thought about how the system should be implemented. For this assignment, one of the people who was not chosen to evaluate the requirements and design documents should be tasked with validating the implementation of their own system. In particular, this task involves writing complete JUnit test cases for all of the methods in your program. The other member of your team is then responsible for writing tests in JUnit for the system that was shared with your team by the partner team. Whenever possible, these two team members should try to re-use tests across the two systems.

The task of these two testers is to ensure that the program faithfully adheres to the description(s) already produced by the specifiers and designers of both systems. When an aspect of the requirements and design documents is not clear, the developer(s) must talk with maintainer of the system to resolve any outstanding concerns. If it is necessary to do so, this laboratory assignment may necessitate that the requirements, design, or implementation of either system evolve so as to best support comprehensive verification. The creators of these deliverables must quickly commit any changes to their repository so as to best ensure that the requirements and design of the system are in sync with its implementation and the testing effort that you complete during this assignment.

At the start of this laboratory assignment, the two testers on your team may think that "they do not have any work to do". First, these individuals should study sources on the Internet and in the ACM Digital Library to learn how to test a Java program using JUnit test cases. Next, these two team members should review the content in Section 3.1 so that they can learn more about project planning and the best way to define project milestones and activities. Using the knowledge from this content in SETP, the team members should outline the activities and milestones that are relevant for the completion of this assignment in one week. Finally, they can examine the content in Section 3.2 to learn more about the roles in a software engineering project and the working styles that team members may exhibit during this assignment. Ultimately, these team members must plan the completion of the project to ensure the timely delivery of two well-tested programs.

Summary of the Required Deliverables

This assignment invites you to submit printed and signed versions of the following deliverables; please see the instructor if you have questions about any of these items. You must make sure that your team uses the same version control repository in Bitbucket as you did for the last assignment when storing the evaluations and tests for your project. Also, you should store the evaluations and tests for your "partner" team's system in their shared repository. Don't forget that all of the documents must be prepared with Markdown and, for submission, converted to PDF using `pandoc`.

1. A complete description of the roles that each team member fulfilled.
2. A full-featured evaluation of your system and the system that you received.
3. A detailed description of the project management plan that your team followed.
4. Well-documented Java source code for all of the tests for both of the systems.
5. A tutorial that explains how to run the test suites for both of the systems.
6. A collaboratively written "lessons learned" document reflecting on these two labs.