

CMPSC 280
Principles of Software Development
Fall 2015

Laboratory Assignment Five:
Releasing and Maintaining a Programming Systems Product

Introduction

In the past laboratory assignment, your team of three individuals created a preliminary version of a programming systems product and prepared a short presentation describing your system. At the start of today's laboratory session, you will hear the presentations from each of the teams and then decide which of these systems are best suited to moving forward for a public release as part of the current assignment. The ultimate goal for this laboratory assignment is for you and your team to create a publicly available GitHub repository that contains a complete, full-featured, working, and properly documented programming systems product. As part of the completion of this assignment, your team will use features of GitHub to, for instance, resolve issues raised by people who have begun to use your software tool. During the start of next week's laboratory session, you will give a detailed presentation describing all aspects of your system and then participate in a release party.

Organizing Your Software Development Team

Please organize yourselves into teams of exactly six students. All of the students in this class should work together to identify which of the six projects are most likely to yield a high-quality system that will be of practical use to individuals who have many Git repositories in their filesystem. Then, you should select three of the six projects and organize yourselves into teams of six students each. The membership of the new teams must consist of entire teams from the previous assignment. That is, all of the members of one team must agree to join together with all of the members of another team. As you are forming your teams, please think about the strengths and weaknesses of the potential members of your development team. Next, each of your teams should pick one person to serve in the role of "chief programmer" (CP) — this individual will lead the effort to release your system to GitHub and ensure the conceptual integrity of your programming systems product.

Since the purpose of this assignment is to release a final product to GitHub, it is important to ensure that you assemble a team with a diverse set of skills. While it is important to have some strong programmers on your team since you may add and/or enhance features of your software tool, it is also crucial for you to have good writers and presenters who can take the lead in, for example, polishing the requirements and design documents, revising and extending the tutorial, and creating a lengthy presentation. Finally, your team should ensure that all of your members know how to effectively use Git and Slack and then make a plan for how you will control your source code and documentation and communication using channel messages and integrations.

At this start of this assignment, each member of your team should create a GitHub account and add all of their relevant details to their account's profile (i.e., each student should include their full name, Web site, and profile picture in their GitHub account). Next, your team should pick a name for your project that best reflects the features that you intend to have in the final version of your tool. Now, your team's CP should create a GitHub repository (that has your chosen name)

and ensure that each team member is a “collaborator” who has write access. Additionally, the CP must ensure that the course instructor has write access to the GitHub repository. When your team is transitioning from the use of Bitbucket to GitHub for this assignment, you may consider adding the new GitHub repository as a “remote” for an existing Bitbucket repository. Please see the course instructor if you have questions about creating and configuring your GitHub repository.

Clarifying the Customer’s Requirements

Once you have formed your team and created and configured your new GitHub repository, then you and your team should decide what features will be part of the final programming systems product. Since your team will have members from the smaller teams of the past laboratory assignment, the CP should ensure that all potential features are openly discussed — that is, it may be possible for the final version of your tool to have features from the projects of different members of your new team. Next, your team should identify individual(s) who can continue to clarify the customer’s requirements for the tool. In particular, if your team develops ideas for new features, then it is crucial for you to present these to the customer before starting implementation.

Using the requirements documents produced by your past teams as a starting point, your new team should prepare a final requirements document that clearly explains all of the functional (behavior) and non-functional (quality) requirements for your system. Written in Markdown, your final requirements document should be available in your project’s GitHub repository. As you are finishing your requirements document, please ask yourself the following questions as you determine whether or not you have written high-quality and useful requirements. Students who want to learn more about the correct way to state the requirements of a software system should review Sections 4.1 through 4.4 of SETP, paying particular attention to the different types of requirements.

1. Are the requirements correct?
2. Are the requirements consistent?
3. Are the requirements unambiguous?
4. Are the requirements complete?
5. Are the requirements feasible?
6. Is every requirement relevant?
7. Are the requirements testable?
8. Are the requirements traceable?

Improving the System’s Design

Using the updated requirements document, your team should identify individual(s) who can translate the revised requirements document into a complete design that encompasses all of the system’s intended features. Your final design document, including any technical diagrams that illustrate the system’s design, should be made publicly available in your team’s GitHub repository.

As you are finalizing the system’s design, you should try to further develop answers to relevant questions such as: How many components will you use? What will be the relationship between the components? What functions will the components have? What will be the inputs and outputs

of the functions? Is the design easy to understand? Can you actually implement and test this design? The answers to these and other relevant questions should be written in a Markdown-based design document containing text and diagrams that explain the system. Since the main focus of this assignment is not design diagrams, students may use any reasonable tool to create their diagrams and, if necessary, include them as separate files in their team's repository. When clarification of the system's specification is necessary (e.g., due to writing ambiguities), the designers should interact with the individual(s) who elicited the system's requirements. Ultimately, your team should carefully ensure that the system's requirements and design are correctly in sync.

Implementing and Testing the System

The developer(s) and tester(s) on your team should take the revised requirements and design documents and start to think about how the system will be (re-)implemented. Your task is to ensure that the program faithfully adheres to the descriptions already produced by other members of your team. When an aspect of the requirements and design documents is not clear, the developer(s) must talk with the people who created these documents to resolve any outstanding concerns. The creators of these documents must quickly commit any changes in them to their GitHub repository so as to best ensure that the requirements and design of the system are in sync with its implementation. Please bear in mind that it is now acceptable to share source code that was created by any member of the two teams that were combined to form the larger team for this assignment.

As this programming systems product will be initially released to GitHub by Monday, September 28, 2015, the implementors and testers should take care to create a system that is well-documented through comments and other relevant documentation. Whenever it is possible to do so, these members of your team must also add (semi-)automated methods for verifying that the implementation adheres to the requirements that previously were elicited from the customer. For instance, if your team decides to implement your tool in the Java programming language, then you should test it with automated tests cases that you wrote in JUnit, the industry standard for automated testing in Java. Overall, as you are implementing and testing your system you should hold yourselves to a high standard in the knowledge that other software engineers will review and use your code on Monday of next week. Please see the course instructor if you have questions about these matters.

Handling Issues Raised by External Users

One of the key features provided by the GitHub service is the ability to raise and respond to "issues". For instance, the issue tracking feature of GitHub would allow one member of your team to document a deficiency of your current system and assign the task to another member of the team who will then resolve the issue and later mark it as being completed. Alternatively, the course instructor could find a mistake in the documentation and then use the issue tracker to ask your team to fix the error. To get a better idea of how issue tracking works in GitHub, it is a good idea for each member of your team to browse the issue tracking site for a popular GitHub project.

Starting on Monday, September 28, 2015, the course instructor will begin inviting students and faculty in the Department of Computer Science at Allegheny College and computer scientists and software engineers from around the world to use your programming systems product. These individuals will be encouraged to use GitHub to raise issues about your tool, under the expectation that your team will professionally respond to and, ultimately, resolve their concerns.

Ensuring the Effective Operation of Your Team

When you start to work on this laboratory assignment, it may seem as though the designer(s), developer(s), and tester(s) “do not have any work to do” because the requirements of the system have not been established. Yet, if you carefully think about the work that you must complete for this assignment, it will become clear that this is not the case! For instance, one member of your team should be tasked with creating all of the needed means for communication with tools such as Slack and Bitbucket. Additionally, team members who are waiting to complete their chosen tasks should consider investigating the use of tools, like Trello, to organize their team’s efforts. Finally, it is important for team members to spend time creating the templates for their deliverables and then carefully “sharpening their tools”. For instance, the developer(s) on the team can ensure that they have a smoothly functioning development environment that will support the implementation of well-documented and correct code. Please see the instructor if you have questions about this matter.

Presenting Your Programming Systems Product

At the start of next week’s laboratory session, each team will give a detailed twenty to thirty minute presentation completely explaining all aspects of their final programming systems product. You will be responsible for highlighting the key features of your tool and the ways in which you did your best to specify, design, implement, release, and document it. Whenever possible, you should give a high-quality, interactive, and engaging presentation that is fun, interesting, and technically correct. Unless a team can demonstrate that it is not possible for them to implement their presentation as a program, all teams should use frameworks like `reveal.js`, `big`, or `beamer` to create their presentations. Creating your presentations using one of these tools is ideal because it will allow your team to store all of its content in your GitHub version control repository and thus ultimately be released with the rest of your programming systems product.

Since the presentations and the associated question and answer sessions are expected to take most of next week’s laboratory session, the course instructor will not release a new laboratory assignment next week. Instead, we will host a “release party” to celebrate all of your new tools. Please invite your friends to next week’s session so that they can enjoy your talks and the party!

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. Please make sure that your team creates a version control repository in GitHub to store all of these deliverables.

1. A requirements document that fully describes the features of your system.
2. Expressed in writing and as a simple technical diagram, the design of your system.
3. Well-documented source code that fulfills all of the system’s requirements.
4. A tutorial that explains how to use all of the features of your finished system.
5. An regularly-used issue tracker in GitHub that is populated with resolved issues.
6. An informative and interesting thirty-minute presentation that highlights your software.
7. Individually completed and submitted reviews of all of the team members.