# SFT221 SCRUM Report and Reflections

This report should be completed in the class and submitted at the end of class. Late submissions cannot be accepted without prior approval of the instructor. All students are expected to attend the in-class SCRUM meetings and to participate. Failure to do so will result in greatly reduced grades.

**GROUP**: 4

**Members Present**:

|  |  |
| --- | --- |
| 1. Song Hwan Oh | 4.Sangjune Lee |
| 2. Shine Lee | 5. Ji Ho Nam |
| 3. Yoojin Lee | 6. |

## Milestone 4 Tasks

**Deliverables Due at end of Lab:**

* Completed SCRUM report and reflections

**Deliverables Due at 23:59 6 Days after Lab:**

* Implemented Functions
* Implemented blackbox tests (store in repo), executed (results in Jira and on corresponding test documents) and debugged,
* whitebox tests written and stored in repository.
* whitebox tests implemented (store in repo), executed (results in Jira and on corresponding test documents) and debugged.
* Updated function-test matrix stored in the repository.
* Completed hook for test automation

**Rubric**

|  |  |  |
| --- | --- | --- |
| Individual | Group Participation | 75% |
| Teamwork | 5% |
| SCRUM Report | 10% |
| Automation Hook | 10% |
| Group | Implemented Functions (well-designed, written and documented) | 20% |
| Whitebox tests (well-designed, written and documented) | 20% |
| Test Execution (performed, results recorded, issues created) | 20% |
| Debugging (Bugs fixed, documented, Jira updated) | 5% |
| Git Usage (used properly with good structure) | 5% |
| Jira Usage (creates issues, tracks progress) | 5% |
| Meets Deadlines | 5% |
| SCRUM Report and Reflections | 20% |

**SCRUM Report**

**Summary of Tasks Completed or Delayed in the last week:**

Here you can list all of the tasks completed in the last week along with any tasks which could not be completed with a reason why they could not be completed.

|  |  |  |
| --- | --- | --- |
| **Member** | **Tasks Completed** | **Tasks Delayed/Blocked** |
| **Song Hwang Oh** | **SCRUM, Function specs implementation** | **N/A** |
| **Shine Lee** | **SCRUM, Testing code(blackbox, whitebox)** | **N/A** |
| **YooJin Lee** | **SCRUM, Testing code(blackbox, whitebox)** | **N/A** |
| **Sangjune Lee** | **SCRUM, Testing code(blackbox, whitebox)** | **N/A** |
| **JI Ho Nam** | **SCRUM, Function specs implementation** | **N/A** |
|  |  |  |
|  |  |  |

For every task delayed or blocked, describe the reason for the delay or block, how it impacts the project and the proposed solution or workaround**.**

|  |  |  |
| --- | --- | --- |
| **Delayed or Blocked Task** | **N/A** |  |
| **Reason for delay or block** | **N/A** |  |
| **Impact on Project** | **N/A** |  |
| **Solution or work-around** | **N/A** |  |
|  |  |  |
| **Delayed or Blocked Task** | **N/A** |  |
| **Reason for delay or block** | **N/A** |  |
| **Impact on Project** | **N/A** |  |
| **Solution or work-around** | **N/A** |  |

**Summary of Meeting:**

A summary of the main points discusses in the meeting and the outcomes of the discussions.

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| Function specs implementation | **Implementing function specs that was developed in finder.h in last week** | **Implementation finished and discussed in meeting** |
| SCRUM | **SCRUM done** | **SCRUM Finished** |
| Testing Functions | **Testing Functions were discussed as it is black box testing and white box testing** | **Testing Functions written and executed.** |
| Jira | **Task Schedule setup in Jira (Debug ticket)** | **Completed** |
| Git | **Git update to each branch (Debug tickets on Git project)** | **Completed** |
|  |  |  |
|  |  |  |

**Summary of Decisions Made:**

This will include major architecture and design decisions, testing decisions, prioritization of tasks, dealing with problems encountered and other major outcomes from the meeting.

|  |  |
| --- | --- |
| Decision | Rationale |
| Prioritization of tasks | Equal amount of works assigned to each member of team. |
| White Box testing | Need new testing implementation, executed and recorded in matrix for MS04s |
| Black Box testing | DEF1 and DEF2 has been debugged and fixed and pushed to master. Debugging record exist in git project and jira. |
| Function implementations | Implementation will follow finder.h that was developed last week as function specs. |
|  |  |
|  |  |
|  |  |

**Tasks Attempted During Meeting:**

Each member is assumed to participate in the SCRUM meeting and contribute to the completion of the SCRUM report and reflections. Since the SCRUM meeting will not take more than 20-30 minutes, there is lots of time left to undertake some of the actual work tasks. In the table below, each member should list what they did to complete the SCRUM report, the reflections, and 1-4 other tasks they completed during the class period. If a task could not be completed, the student should indicate why this was not possible.

|  |  |  |  |
| --- | --- | --- | --- |
| Member | Task Attempted | Time Spent | Complete? |
| ALL | **Analysis discussion, Analysis of debugging black box testing that was done last week and discuss about white box implementation and execution.** | **1hr** | **Yes** |
| ALL | **Analysis discussion, Analysis Function implementation in program and discussed** | **1hr** | **Yes** |
| ALL | **Scrum report** | **30min** | **Yes** |
| ALL | **Jira and Github Project page updated and assigned** | **30min** | **Yes** |
| ALL | **Discussion on hook automation** | **30min** | **Yes** |
| ALL | **Discussion for next week tasks** | **15min** | **Yes** |
|  |  |  |  |

**SCRUM Tasks Selected for Next Week**:

The tasks each member has selected to pursue for this class or the next week.

|  |  |
| --- | --- |
| Group Member | Task Description |
| Sangjune Lee | Jira control |
| ALL | Meeting on July 31th Monday 9pm to 11pm |
| ALL | SCRUM , Reflection |
| ALL | Acceptance Tests |
| ALL | Integration Tests |
| ALL | Debuggings |
| ALL | Test Execution |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

This is where you should highlight the major accomplishments of the class.

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Function implementation | **Function implementation was done according to the function specs that was written for last week** |
| Black Box testing | **Team has finished attempting black box testing last week even though it was for this week’s requirement. We have found some bugs in our testing codes, we ticketed on matrix and Jira kanban, Git project(kanban), and issue was resolved** |
| White box testing | **White box testing codes were implemented and executed.** |
| Hook implementation | **Hook implementation was discussed and screenshot was sent to professor.** |
|  |  |
|  |  |
|  |  |

**Things That Went Well in This Meeting:**

Here you can highlight things which worked well. This indicates that the way you worked on these items is working and should be continued.

|  |  |
| --- | --- |
| Topic/Work Item | Reason for Success |
| SCRUM | All contributed. |
| Git | **Useful for version control and keeping track of changes** |
| Meeting | All attended meeting. |
|  |  |
|  |  |
|  |  |
|  |  |

**Things That Did NOT go Well in This Meeting:**

This is where you can list things which did not go well in the class. You should analyze why this happened and suggest how you can improve it next time. This will lead to the goal of *continuous process improvement*.

|  |  |
| --- | --- |
| Topic/Work Item | Reason for Problem and How to do Better |
| N/A | **N/A** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Reflections**:

1. After you run your blackbox and whitebox tests you are asked to record the results in both the original test document as well as in Jira. Explain why it is a good idea to record the results in both places.

Recording the test results in both the original test document and Jira offers numerous benefits. The test document serves as a comprehensive historical record, documenting all tests conducted, their objectives, and obtained results, ensuring traceability for future reference. Meanwhile, Jira's integration with the development workflow links test results to specific issues or user stories, enabling easy tracking of bug fixes or feature implementations. This fosters collaboration among team members, including developers, project managers, and product owners, who can access the results for communication and decision-making. Real-time updates and notifications in Jira keep everyone informed about the software's current quality status and potential issues. Additionally, Jira's built-in reporting and metrics capabilities allow for the generation of valuable insights into the software's overall quality and the effectiveness of testing efforts. Finally, having test results stored in both places ensures redundancy and compliance with documentation requirements for audits and projects with strict quality standards.

1. Why did we wait until the fourth milestone to write the whitebox tests?  
     
   The decision to delay writing whitebox tests until the fourth milestone could be attributed to several factors. Initially, the focus might have been on validating the software from a black-box perspective to mimic end-user scenarios and identify issues from a user's standpoint. Moreover, the early stages of development often involve significant changes and refactorings to the internal codebase, making it unstable for writing precise whitebox tests. Waiting until the fourth milestone would allow for a relatively stable codebase, minimizing the need for frequent test rewrites. Prioritizing blackbox testing in the initial stages enables the testing team to achieve broader test coverage and address high-level issues arising from user interactions and external systems. Additionally, adopting an iterative testing approach gradually increases testing complexity as the project progresses, ensuring a balanced allocation of testing efforts throughout the development lifecycle.
2. For a given function did you produce more blackbox or whitebox tests? Explain why your answer (more blackbox or more whitebox) happens for most functions.  
     
   In software testing, black box testing is conducted from the user's point of view, where the internal workings and code implementation details of the program are not known to the tester. The primary objective of black box testing is to ensure that the basic functionality of the program works as intended and meets the specified requirements. Testers focus on the program's inputs and expected outputs without considering how the code achieves those results.

By employing black box testing for the given function, mapping.c, we aimed to verify that its external behavior, as seen by the end-users, aligns with the expected functionality outlined in the milestone specification. This approach allows us to assess whether the function fulfills its intended purpose without delving into the intricacies of its internal implementation.

On the other hand, white box testing, as applied to the function finder.c, leverages knowledge of the internal code and logic. It enables us to thoroughly examine and validate the function's internal pathways, ensuring that the code is executing correctly and efficiently. This level of testing is especially valuable when dealing with custom-made data structures and newly implemented functions, as it helps us identify potential bugs, corner cases, and performance optimizations by scrutinizing the code at a granular level.

1. Explain the purpose of the automation hook for GIT and explain how it can improve the quality of the software in the project.

Automation hooks in GIT are like little helpers that automatically run certain scripts or tasks when specific things happen in the version control system. For example, when someone adds new code or makes changes to the existing code (called code commits) or when they want to add their changes to the main project (called pull requests). These hooks offer lots of advantages. They help with continuous integration and continuous deployment, which means that whenever someone adds or changes code, it automatically goes through a bunch of tests to check if everything is okay. This gives quick feedback to developers about the quality of their code and makes it easier to deploy changes to the live website or app. The hooks also help in maintaining good code quality by running checks on the code to make sure it follows the rules and standards set by the team. They can even catch bugs early on, so developers can fix them before they become big problems. With automation hooks, everyone on the team follows the same rules, which makes the development process more consistent. This way, developers don't have to worry about running tests manually or remembering all the rules because the hooks take care of it. This saves time and lets developers focus on the fun and creative parts of coding. Overall, automation hooks make sure that the code is in good shape, reducing the chances of mistakes or errors, and make it easier to manage and keep an eye on things.