# SFT221 SCRUM Report and Reflections

**Members Present**:

Humaira Shaikh

## 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** |
| **Humaira Shaikh** | **SCRUM, Function specs implementation** | **N/A** |
| **Humaira Shaikh** | **SCRUM, Testing code(blackbox, whitebox)** | **N/A** |
| **Humaira Shaikh** | **SCRUM, Testing code(blackbox, whitebox)** | **N/A** |
| **Humaira Shaikh** | **SCRUM, Testing code(blackbox, whitebox)** | **N/A** |
| **Humaira Shaikh** | **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** | **Implementation finished** |
| SCRUM | **SCRUM done** | **SCRUM Finished** |
| Testing Functions | **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** |
|  |  |  |
|  |  |  |

**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** |
| Black Box testing | **finished attempting black box testing**  **found some bugs in testing code, ticketed on matrix and issue was resolved** |
| White box testing | **White box testing codes were implemented and executed.** |
|  |  |
|  |  |
|  |  |
|  |  |

**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. 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. Waiting until the fourth milestone would allow for a relatively stable codebase, minimizing the need for frequent test rewrites. Prioritizing black box 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.
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 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. They can 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.