# 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**: \_\_\_\_\_\_\_\_\_\_\_\_\_5\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1.Thomas Lewis | 4.Kiki |
| 2.Asem M | 5. |
| 3.Hla Myint Myat | 6. |

## Milestone 4 Tasks

**Deliverables due 4 days after your lab day:**

* Finish implementing/coding the functions.
* Finish implementing/coding blackbox tests. Store in repo, executed, results in Jira (and on corresponding test documents, and debugged.
* A set of whitebox tests as test documents with test data for the functions you created. At least 4 sets of test data are required for each function. You must have test cases for at least 6 functions (including all your custom function). Stored in the repository.
* Whitebox tests implemented, stored in repo, executed, results in Jira and on corresponding test documents, and debugged (at least 1 SET is required for this milestone).
* Updated requirements traceability matrix stored in the repository.
* Completed hook file (for EACH team member) for test automation stored in the repository.
* Completed scrum report including reflection questions answered.

**Rubric:**

|  |  |  |
| --- | --- | --- |
| **Individual** | Group participation (includes GitHub commits and Jira usage) | 80% |
| Teamwork | 20% |
| **Group** | Implemented functions and main (well-designed, and documented) | 5% |
| Finish coding blackbox cases (well-designed, written, and documented) | 10% |
| Whitebox test case document (well written, complete, good test data) | 10% |
| Whitebox test code (well designed and documented) | 10% |
| Updated requirements traceability matrix | 5% |
| Test execution (performed, results recorded, issues created) | 5% |
| Debugging (bugs fixed, documented, Jira updated) | 5% |
| Hook files | 10% |
| Git usage (used properly with good structure) | 5% |
| Jira usage (creates issues, tracks progress) | 10% |
| Scrum report & reflections | 20% |
| Meets deadlines | 5% |

**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** |
| **Thomas Lewis** | **Testing Doc** | **N/A** |
| **Asem M** | **Testing functions** | **N/A** |
| **Hla Myint Myat** | **Scrum Report** | **N/A** |
| **Kiki** | **Function Spec** | **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** | **Completed Implementation** |
| Testing Functions | **Testing Functions were discussed as it is black box testing and white box testing** | **Testing Functions written and executed.** |
| Testing Doc | **Created Testing Doc** | **Completed for this ms** |
| SCRUM | **SCRUM done** | **Completed Scrum** |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Breaking down the task | Equal amount of works assigned to each member of team. |
| White Box testing | Need new testing implementation, executed and recorded |
| Black Box testing | DEF1 and DEF2 has been debugged and fixed |
| Function implementations | finder.h 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? |
| Asem M | **White Box Testing** | **1hr** | **Yes** |
| Thomas Lewis | **Hook file** | **1hr** | **Yes** |
| Hla Myint Myat | **Scrum & Reflection** | **1hr** | **Yes** |
| Kiki | **Black Box Testing** | **1hr** | **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 |
| Thomas Lewis | Git Control |
| Asem M | Jira Contorl |
| Hla Myint Myat | Scrum & Reflection |
| Kiki | Testing |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 | **Attempting black box testing last week even though it was for this week’s requirement. We ticked on matrix and Jira Kanban.** |
| White box testing | **White box testing codes were implemented and executed.** |
| Hook implementation | **Hook implementation was discussed** |
| Scrum | **Completed Scrum Report** |
|  |  |
|  |  |

**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 |
| Git | **Useful for version control and keeping track of changes** |
| Jira | **Useful tool for Project** |
| Scrum | **All members contributed** |
| Meeting | **Discussed the overall milestone and deadlines to submit** |
|  |  |
|  |  |
|  |  |

**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**:

Answer the following questions using your own words. Make sure that each answer comprises a minimum of 100 words.

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.  
     
     
   There are several advantages of recording test results in Jira as well as the original test document. The test document ensures traceability for future reference by serving as an extensive record of all tests performed through, their goals, and the results acquired. In the meantime, Jira's connection with the development workflow makes it simple to follow the installation of new features or bug fixes by connecting test results to particular issues or user stories. Because developers, project managers, and product owners may access the results for communication and decision-making, this encourages teamwork among team members. Everyone is constantly informed about potential issues and the software's current quality status through real-time updates and notifications in Jira. Furthermore, Jira's integrated reporting and metrics features enable the creation of insightful analyses into the general status of the programme and the efficacy of testing initiatives. Lastly, redundancy and compliance with documentation requirements for inspections and projects with challenging quality controls are ensured by having test findings kept in both locations.
2. Why did we wait until the fourth milestone to write the whitebox tests?  
     
   There are a few reasons why it was decided to postpone creating whitebox tests until the fourth milestone. When the software was first being tested, it might have been done so from a black-box perspective in order to create end-user experiences and find problems from the user's point of view. Furthermore, the internal codebase is frequently restructured and altered throughout the early phases of development, which makes it uncertain for creating accurate whitebox tests. It would be possible to have a reasonably stable codebase by waiting until the fourth milestone, which would reduce the frequency of test rewrites. The testing team can accomplish more test coverage and handle high-level issues resulting from user interactions and external systems by giving priority to blackbox testing in the early phases. Furthermore, by using an iterative testing methodology, testing complexity may be progressively increased as the project moves forward, guaranteeing a fair distribution of testing resources across the development lifecycle.
3. For this function did you produce more blackbox or whitebox tests? Explain why your answer (more blackbox or more whitebox) happens for most functions.  
     
     
   Black box software testing is performed from the user's perspective, without the tester knowing the internal workings or specifics of the program's code implementation. Making ensuring the program's core functionality functions as expected and satisfies the requirements is the main goal of black box testing. Without considering how the code produces such outcomes, testers rely on the program's inputs and expected outcomes.

We attempted to confirm that the function mapping.c external behavior—as observed by end users—aligns with the anticipated functionality specified in the milestone specification by taking advantage of black box testing.

Meanwhile, white box testing utilizes the use of internal code and logic knowledge when it comes to the function finder.c. We can thoroughly inspect and verify the internal systems of the function with its help, making sure that the code is running accurately and effectively. This level of testing is particularly helpful when working with newly implemented functions and custom data structures because it allows us to examine the code at a finer level and find potential bugs, corner cases, and performance optimizations.

1. Explain the purpose of the automation hook for GIT and explain how it can improve the quality of the software in the project.  
     
   GIT automation hooks are little helpers that, in response to specific conditions within the version control system, set up particular codes or tasks automatically. For instance, when someone wants to add their modifications to the main project (called pull requests) or adds new code or modifies the existing code (called code commits). There are numerous benefits to these hooks. They support continuous integration and deployment, which includes that code is automatically put through a series of tests to make sure everything is working properly whenever new code is added or modified. This facilitates the deployment of updates to the real-time website or app and provides developers with rapid feedback regarding the quality of their code. By executing and verifying the code to ensure that it complies with the guidelines and standards set by the team, the hooks also help in maintaining high code quality. Even bugs can be detected early by them, allowing developers to address them before they become significant issues. The development process is more consistent when automation hooks are used because everyone in the team sticks to the same list of guidelines. Because the hooks handle it, developers can stop worrying about manually running tests or remembering all the rules. Automation hooks, in general, ensure that the code is well-maintained, lowering the likelihood of errors or mistakes, and helping with management and oversight.