# 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**: 1

**Members Present**:

|  |  |
| --- | --- |
| 1. Taehwa Hong | 4.Hyunjoo Han |
| 2. Jenna Moon | 5. Natalya Pak |
| 3. Farouk Alhassan | 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** |
| **Taehwa Hong** | **SCRUM, Function implementation** | **NONE** |
| **Jenna Moon** | **SCRUM, Tracebility Matrix** | **NONE** |
| **Farouk Alhassan** | **Test-description** | **NONE** |
| **Hyunjoo Han** | **SCRUM, Function implementation** | **NONE** |
| **Natalya Pak** | **Test-cases** | **NONE** |
|  |  |  |
|  |  |  |

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** | **NONE** |
| **Reason for delay or block** | **NONE** |
| **Impact on Project** | **NONE** |
| **Solution or work-around** | **NONE** |
|  |  |
| **Delayed or Blocked Task** | **NONE** |
| **Reason for delay or block** | **NONE** |
| **Impact on Project** | **NONE** |
| **Solution or work-around** | **NONE** |

**Summary of Meeting:**

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

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| Function implementation | **Implementing functions in shipment.h in last week** | **Complete implementation and discuss meeting** |
| Scrum Report | **Scrum Report done** | **Scrum Report Done** |
| Testing Functions | **Discussed black box test and white test** | **Test fucntions written and run** |
| 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. |
| Black Box Testing | MS4 requires new test implementation and execution and recording in matrix |
| Function Implementation | Implementation will follow shipment.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? |
| Taehwa Hong  Jenna Moon  Farouk Alhassan  Hyunjoo Han | **Analysis discussion, analysis of the debugging black box tests conducted last week, and discussion of white box implementation and execution.** | **1hr** | **Yes** |
| Taehwa Hong  Jenna Moon  Farouk Alhassan  Hyunjoo Han | **Analysis discussion, implementation and discussion of analysis functions within the program** | **1hr** | **Yes** |
| Taehwa Hong  Jenna Moon  Farouk Alhassan  Hyunjoo Han | **Scrum Report** | **30min** | **Yes** |
| Taehwa Hong  Jenna Moon  Farouk Alhassan  Hyunjoo Han | **Jira and Github update** | **30min** | **Yes** |
| Taehwa Hong  Jenna Moon  Farouk Alhassan  Hyunjoo Han | **Hook automation** | **30min** | **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 |
|  |  |
| ALL | Meeting on November 23T  Thursday 2pm to 3pm |
| 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 | **The function implementation was done according to the function specification written last week** |
| Black Box testing | **Found a few bugs in my test code, ticketed them in Matrix, Jira Kanban, and Git project (Kanban) and the issue was resolved** |
| White box testing | **White box test code was implemented and run** |
| Hook implementation | **Hook implementation and sent the video to our 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 | Most of members contributed. |
| Git | Useful for version control and change tracking. |
| Meeting | Most of members 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 |
| Meeting | Natalya Pak was absent |
| Prioritization of tasks | Natalya Pak was absent from meeting |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 benefits to recording test results both in the original test document and in Jira. Cross-verification and correction are possible by ensuring document consistency. Jira's traceability between test results and specific development tasks improves project management. Streamline collaboration by enabling stakeholders to easily access and review test results, promoting effective communication. Jira's reporting and dashboarding capabilities help you generate meaningful metrics to support data-driven decisions. Integration with development workflows aligns test information with the broader context of the project. Tracking history in both locations makes it easier to identify patterns and recurring issues over time. Overall, this dual-recording approach improves organizational efficiency and accessibility of critical test data.
2. Why did we wait until the fourth milestone to write the whitebox tests?  
     
   The decision to delay white box testing until the fourth milestone is influenced by various factors. Initially, the focus is on validating the software through a black box perspective, mimicking end-user scenarios to comprehend issues from the user's viewpoint. Additionally, early development stages often entail significant changes or refactoring in the internal code base, rendering it unstable for accurate white box testing. Postponing until the fourth milestone ensures a relatively stable code base, minimizing the need for frequent test rewrites. Prioritizing black-box testing early on enables comprehensive test coverage and resolution of high-level issues arising from user interactions and external systems. Moreover, adopting an iterative testing approach allows testing complexity to incrementally increase as the project progresses, ensuring a well-balanced distribution of testing efforts across the development life cycle.
3. Pick one of the functions you created and list its name. For this 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 standpoint, with the tester having no knowledge of the program's inner workings or code implementation details. The primary goal is to verify that the program's fundamental functionality operates as intended and adheres to specified requirements. Testers concentrate on input and expected output without considering the underlying code mechanisms.

Applying black box testing to the feature mapping.c, the objective is to confirm that the external behavior perceivable by end users aligns with the anticipated functionality outlined in the milestone specification. This methodology allows for an evaluation of whether the function accomplishes its intended purpose without delving into the intricacies of its internal implementation.

On the other hand, white box testing, exemplified by the shipment.c function, involves utilizing knowledge of the internal code and logic. This permits a comprehensive examination and validation of the internal pathways to ensure correct and efficient code execution. This level of testing is valuable for scrutinizing the code at a granular level, identifying potential bugs, addressing special cases, and optimizing performance.

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

The automation hook for Git is a customizable script mechanism that triggers automated processes before or after specific events in the Git workflow, enhancing the development and quality assurance processes in a software project. By integrating with Continuous Integration (CI) systems, enforcing code quality checks, and triggering automated testing suites, the automation hook ensures early detection of issues, adherence to coding standards, and thorough validation of changes. It also aids in maintaining consistent commit message standards, preventing unauthorized actions, updating documentation automatically, and automating deployment processes. This contributes to a more efficient and collaborative development pipeline, leading to improved code quality, faster feedback loops, and a streamlined development workflow.