# 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. Chang Cui | 4. Xiaofei Xu |
| 2. Lok Yin Tai | 5. Xinyang Wu |
| 3. Siu Man Cheng | 6. Ye Tian |

## 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** |
| **Chang Cui** | Test documents  Test Traceability Matrix  Jira Board Management  GitHub Repository Management | N/A |
| **Lok Yin Tai** | Functions Spec  Jira Board Management  GitHub Repository Management | N/A |
| **Siu Man Cheng** | Functions Spec  Jira Board Management  GitHub Repository Management | N/A |
| **Xiaofei Xu** | Reflection  Jira Board Management  GitHub Repository Management | N/A |
| **Xinyang Wu** | Test Code  Jira Board Management  GitHub Repository Management | N/A |
| **Ye Tian** | Test documents  Test Traceability Matrix  Jira Board Management  GitHub Repository Management | 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 |
| Review on last week work | The team reviewed the work and difficulties from the previous week. The team provided comments to help each other. | Identified areas for improvement and appreciate each other. |
| Improvement base on professor comment and grading | Read the comments and discussed how to get better outcome in the coming works | Agreed to make more use in Jira to facilitate the communication. |
| Summary of lecture content | The team shared their understanding and knowledges of the lecture content and how it is related to MS4. | All team members have clear understanding on lecture material and requirements on MS4. |
| Discussion on MS4 task distribution | The team discussed the tasks required for Milestone 4 and divided the tasks evenly. | The tasks are assigned to team members by random drawing. |
| Knowledge sharing on milestone 4 tasks | The team discussed the requirements and expectations for each deliverable. | All agreed with the same direction to continue the works |
| Deadline discussion | The team discussed the dependencies between each task and determined the due dates for each member. | All team members agreed with the due dates and ensured the high-quality completeness of MS4 |

**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 |
| Tasks are divided into 6 parts based on workload | Fair distribution of work among team members. Ensured the tasks are finished with high efficiency and productivity. |
| Task distributed using random dice game | Fair and unbiased method in work division |
| Clarified the milestone deliverables and document to be submitted | Understanding of MS4 requirements and the whole project requirements are again emphasized, preventing misunderstandings. |
| Reemphasize the importance of Jira comments usage | Ensured communication and feedback exchange are made on a centralized platform, enhancing effective collaboration. |
|  |  |
|  |  |
|  |  |

**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? |
| Chang Cui | Reflection  Jira Board Management  GitHub Repository Management  Update test Traceability Matrix | 50 mins  10mins  10mins  10mins | Yes  Yes  Yes  Yes |
| Lok Yin Tai | Blackbox tests implementation, execution, and debug  Update test Traceability Matrix  Jira Board Management  GitHub Repository Management | 50 mins  10mins  10mins  10mins | Yes  Yes  Yes  Yes |
| Siu Man Cheng | Whitebox tests implementation, execution, and debug  Jira Board Management  GitHub Repository Management | 60 mins  10mins  10mins | Yes  Yes  Yes |
| Xiaofei Xu | Whitebox tests  Update test Traceability Matrix  Jira Board Management  GitHub Repository Management | 50 mins  10mins  10mins  10mins | Yes  Yes  Yes  Yes |
| Xinyang Wu | Functions implementation  Jira Board Management  GitHub Repository Management | 60 mins  10mins  10mins | Yes  Yes  Yes |
| Ye Tian | Functions implementation  Jira Board Management  GitHub Repository Management | 60 mins  10mins  10mins | Yes  Yes  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 |
| Chang Cui | * Reflection * Jira Board Management * GitHub Repository Management |
| Lok Yin Tai | * Blackbox tests implementation, execution, and debug * Update test Traceability Matrix * Jira Board Management * GitHub Repository Management |
| Siu Man Cheng | * Whitebox tests implementation, execution, and debug * Jira Board Management * GitHub Repository Management |
| Xiaofei Xu | * Whitebox tests * Update test Traceability Matrix * Jira Board Management * GitHub Repository Management |
| Xinyang Wu | * Functions implementation * Jira Board Management * GitHub Repository Management |
| Ye Tian | * Functions implementation * Jira Board Management * GitHub Repository Management |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Understand the correlation between tasks | Better planning and coordination, ensure everyone has enough time to work on their own tasks. |
| Tasks for MS4 divided into six parts based on workload and difficulty | Fair work division, optimize productivity, individual contributions, and efficiency. |
| Agreed on making improvement based on MS3 professor comments | Improve upcoming works, aimed for better outcome. |
| Ensured individual parts concerns are addressed | Ensured the team are clear with the requirements and instructions, without worries or concerns. |
| Strive for quality work | Ensured the team are clear with goals of the project, strive for 100/100 for each work. |

**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 |
| Understanding of individual parts and the dependencies of each part | Ensured every member fully understood the correlation between tasks and agreed with the assigned tasks and individual due dates. |
| Fair workload and tasks rotation for MS4 | Every member agreed with the fairness on task division, and the work rotated between members. |
| Ensured all with the same target and objective in the project | Ensured every member agreed with the objective in finishing the project and aimed at a high-quality submission. |
| Active problem solving and knowledge sharing | Encouraged teamwork and problem-solving, maintain good team collaboration |

**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 | The meeting went smooth, all items were well discussed. Team members participated actively during the meeting. |

**Reflections**:

* **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.**  
    
  The test document is a vital repository for our test cases, encompassing crucial details such as input data, test code, execution procedures, and expected results. Documenting our test results in this comprehensive test document ensures a clear historical record of our testing process, facilitating future reference and analysis. Additionally, the Requirements Traceability Matrix underscores the significance of traceability in our project, addressing two critical challenges: ensuring that our software impeccably implements each requirement and verifying that all necessary tests have been developed for the software.

On the other hand, Jira serves as our centralized platform for tracking issues, bugs, and monitoring project progress. Recording our test results in Jira offers our team, including developers and testers, valuable insights into the outcomes of our testing efforts, enhancing collaboration and communication within our team. Jira's seamless task assignment and prioritization features streamline issue resolution, ensuring timely handling of any identified issues.

Moreover, using Jira for storing test results elevates traceability and accountability in our project. We can effortlessly establish connections between our test cases, requirements, and defects through the Jira platform, simplifying the debugging process and allowing us to identify the root cause of any issues efficiently. The reporting capabilities of Jira also empower project stakeholders to generate meaningful metrics and progress reports, offering comprehensive insights into the overall status and health of our testing and development efforts.

* **Why did we wait until the fourth milestone to write the whitebox tests?**

Firstly, writing whitebox tests typically requires a deep understanding of the internal workings of the code, including its structure, logic, and implementation details. By waiting until the fourth milestone, we had already completed the implementation of functions and blackbox tests. This allowed us to gain a solid understanding of the codebase and ensure that it was functioning correctly from an external perspective before diving into the intricacies of whitebox testing.

Secondly, by focusing on blackbox tests first, we could prioritize the testing of the code's functionalities and ensure that the intended behavior was met. This approach enabled us to validate the expected outputs based on the specified requirements, and only after confirming the external behavior could we proceed with fine-tuning the code through whitebox testing.

Additionally, whitebox testing requires inspecting the code's internal structure, which may involve additional time and effort to identify potential weak points or corner cases. By deferring whitebox testing, we avoided the risk of spending excessive time on lower-level details while ensuring that the primary functionality was sound and met the initial requirements.

* **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.**

FUNCTION: findTruckForShipment

In our team, we produced more whitebox tests for this function. The reason behind this choice is that whitebox testing allows us to have a deeper understanding of the internal logic and structure of the function. By having access to the function's source code, we were able to design test cases that specifically target different paths and branches within the function. This approach helped us assess the function's robustness and identify potential edge cases that might not be evident from the function's prototype alone.

Whitebox testing enabled us to exercise various conditional statements, loops, and boundary conditions within the function, ensuring that different code segments were adequately tested. We could verify if the function behaves as expected under different scenarios and validate its handling of invalid inputs and exceptional cases.

On the other hand, blackbox testing is equally important as it focuses on the function's external behavior and input-output relationships without considering the internal implementation details. The blackbox tests we conducted covered typical use cases and boundary values based on the function's specifications. They allowed us to evaluate the function's overall correctness and ensure that it produces the expected results for different inputs.

In conclusion, our emphasis on whitebox testing was driven by the desire to gain a comprehensive understanding of the function's behavior, enabling us to design test cases that thoroughly scrutinized its internal workings. By combining both whitebox and blackbox testing approaches, we aimed to achieve a high level of test coverage, ensuring the function's reliability and accuracy across various scenarios.

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

The purpose of the automation hook for GIT is to streamline the testing process and ensure the code meets the required quality standards before it is pushed to the remote repository. The hook runs tests automatically when a push is attempted. If any tests fail, the push is aborted, prompting group members to fix the code before committing changes. This automation hook acts as a pre-push check that automatically runs the unit tests on the code, preventing the introduction of bugs and defects into the repository.

This automation hook significantly improves software quality in several ways. Firstly, it detects bugs early, allowing prompt resolution. Secondly, it enforces consistent testing across team members, ensuring uniformity. Thirdly, it aligns with continuous integration principles, encouraging frequent code integration and automated testing.

Moreover, the hook helps prevent code conflicts and encourages best practices by motivating comprehensive unit testing. It reduces manual errors and provides faster feedback, leading to quicker iterations and delivery of high-quality code.