# 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**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1. Yatin Bawa | 4. Rehatpreet Kaur |
| 2. Muhammetyar Yarov | 5. Aayush Bhogal |
| 3. Nehmat Ladhar | 6. Omar Carraso |

## 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** |
| **Aayush Bhogal** | **Scrum Report, Reflection Questions, Traceability Matrix. Encountered delays due to additional required testing for new functions.** | **Extension** |
| **Yatin Bawa** | **Implemented Dispatch Function, faced challenges with integration testing that required more time for debugging.** | **Extension** |
| **Rehatpreet Kaur** | **Developed white box tests, but encountered issues with edge cases that necessitated further test development.** | **Extension** |
| **Nehmat Ladhar** | **Led function implementation, but needed additional time to address complex bugs discovered during integration.** | **Extension** |
| **Muhammetyar** | **Communication with professor regarding extensions, worked on PromptUser function which required further UX testing.** | **Extension** |
| **Omar Carraso** | **Reviewed custom functions, but required additional time to optimize code for performance efficiency.** | **Extension** |
|  |  |  |

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** |  |
| **Reason for delay or block** | **Integration complexities and the discovery of critical bugs during testing phases led to the need for a project extension.** |
| **Impact on Project** | **The extension will allow for a more comprehensive testing phase, ensuring a higher quality end product.** |
| **Solution or work-around** | **The team has increased collaboration with frequent code reviews and pair programming sessions to address the identified issues promptly.** |
|  |  |
| **Delayed or Blocked Task** |  |
| **Reason for delay or block** |  |
| **Impact on Project** |  |
| **Solution or work-around** |  |

**Summary of Meeting:**

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

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| Scrum Report and Documentation | **Aayush Bhogal provided an update on the Scrum Report and the Traceability Matrix, seeking input for completion.** | **Collaborative editing session planned to finalize documents.** |
| Function Implementation | **Nehmat Ladhar reported on function implementation progress and the need for additional unit tests.** | **Scheduled a review session to assess test coverage and function performance.** |
| Testing Procedures | **Yatin Bawa and Rehatpreet Kaur discussed the current state of testing, identifying areas that required deeper investigation.** | **Decided to introduce new test scenarios to ensure comprehensive coverage.** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Expand Test Coverage | To validate the robustness of new features and functions against a variety of scenarios. |
| Enhance Documentation | To accurately reflect the project's status and maintain alignment with our goals. |
| Optimize Codebase | To improve efficiency and maintainability of the code as we progress. |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Aayush Bhogal | **Enhancing the Traceability Matrix** | **1.5 hours** | **Yes** |
| Yatin Bawa | **Debugging of Dispatch Function** | **1 hour** | **Yes** |
| Rehatpreet Kaur | **Development of White Box Test Cases** | **2.5 hours** | **Yes** |
| Nehmat Ladhar | **Function Implementation Review** | **1.5 hours** | **Yes** |
| Muhammetyar Yarov | **Drafting of Communication for Stakeholders** | **1 hour** | **Yes** |
| Omar carraso | **Reviewing and Optimizing Custom Functions** | **1 hour** | **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 |
| Aayush Bhogal | Expand and explore more requirements |
| Yatin Bawa | Execute additional test cases and finalize debugging of Dispatch Function. |
| Rehatpreet Kaur | Expand White Box Testing to cover additional scenarios. |
| Nehmat Ladhar | Incorporate peer feedback into function implementation and continue documentation. |
| Omar carraso | Finish optimizing custom functions and confirm performance metrics. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Task Prioritization | **Aligns team efforts with project milestones and ensures timely progress.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Task Execution Review | **The team’s thorough examination of current work helped to identify areas for improvement.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Integration Planning | **We will implement more rigorous integration checkpoints in future phases to avoid bottlenecks.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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.

Ans - Keeping track of test results in both the original test document and Jira serves numerous important purposes in our project management and quality assurance procedures. To begin, the original test document is our source of truth; it contains a full and detailed record of all test cases and their intended outcomes, as well as the parameters utilized and the environment configuration. This document is essential for audit reasons and for new team members to thoroughly grasp the test coverage.

Incorporating results into Jira, on the other hand, offers real-time visibility into the testing phase of our development cycle. Jira's issue-tracking and project management capabilities allow us to link test results directly to corresponding tickets, ensuring that all team members, including those not directly involved in testing, stay informed about the project's current state. This dual-recording system enhances collaboration, facilitates faster decision-making, and aids in tracking the resolution of any issues discovered during testing.

1. Why did we wait until the fourth milestone to write the whitebox tests?  
     
   Ans: The primary reason for waiting to write the white box tests in the fourth milestone was that we did not have the actual code with us and white box testing requires a fully working code

and not just the prototype (unlike black box testing). Until now we were just planning, making the function prototypes, and performing black box testing based on the prototypes we made.

Our main focus was on selecting the best possible data structure and planning the functions to be made to fulfill the required task. But now, since we have finally implemented the function as planned and we know the internal structure of the function, we are ready to perform white box testing on it. Now, since we can perform the white box test on the completed code, we can test it considering the edge cases, making sure that our code works perfectly fine for all the different scenarios and is ready to be integrated. Performing white box testing, helped

us identify logical errors in our code, optimize our code, and achieve a successful build.

1. 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.  
     
   Ans- For the function calculateOptimalRoute, we produced more whitebox tests than blackbox tests. Whitebox testing was the primary focus because this function involves complex logic that calculates the most efficient path based on various parameters such as distance, time, and traffic conditions. Understanding the internal workings of this function was crucial to ensure that all possible execution paths were tested and that the logic handled edge cases appropriately.

Generally, whitebox testing is predominant for functions where the internal structure influences the outcome significantly, as it allows us to verify the correctness of intricate algorithms. Conversely, blackbox testing is often more suitable for functions with clear input-output relationships that can be tested without knowledge of the internal implementation. However, in the case of calculateOptimalRoute, the intricate logic dictated a need for a thorough whitebox approach to validate the internal processing and ensure the function's reliability in real-world scenarios.

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

Ans: With large software developments and the competition in the market, version control has become a necessity to safely track changes and enable reversions, integrity checking, and collaboration among other benefits. Using "hooks" system -

1) Git allows developers and administrators to extend functionality by specifying scripts that git will call based on different events and actions. Git hooks are event-based. When you run certain git commands, the software will check the hooks directory within the git repository to see if there is an associated script to run.

2) Some scripts run prior to an action taking place, which can be used to ensure code compliance to standards, for sanity checking, or to set up the environment. Using these abilities, it is possible to enforce policies, ensure consistency, control your environment, and even manage deployment tasks.

3) By enabling testing to be performed for new code after it is committed, it assists us to perform automated testing by spotting problems and errors.

4) By performing pre-commit tests for syntax mistakes, coding style violations, security vulnerabilities, the automation hook ensures coding conventions and standard practices.

5) Enforcing project specific requirements which will ensure that a specific amount of test coverage is conducted. This helps maintain a consistent code style and prevents the introduction of common errors.

6) Once the push is approved, it is conducted on the server. Performs automatic deployment and makes sure that team members are notified. This helps in ensuring that the latest modifications are swiftly and reliably distributed to production by automating processes and reducing human errors.

7) The automated hook builds an organized repository that carries out code review prior to merging changes.

In conclusion, hooks system is a helpful tool for developers as it allows them to attach scripts to different actions in code-saving process. These scripts can do various checks and tasks before or after saving changes.