# 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**: 2

**Members Present**:

| 1. Jubril Olawale Akolade 167529213 | 4.Iraklis Tsanachtsidis 122226228 |
| --- | --- |
| 2.Frank Prerez 141647222 | 5.Aum Rasikbhai Parsana 112872221 |
| 3.Tarun Thomas 113605224 | 6.Rutarj Mrushad Shah 170870216 |
| 7. Faaz Sherwani 113026223 |  |

## 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** |
| --- | --- | --- |
| **Iraklis Tsanachtsidis** | **SCRUM REPORT** |  |
| **Frank Prerez** | **Jira assignments** |  |
| **Tarun Thomas** | **Debugging (bugs fixed)** |  |
| **Faaz Sherwani** | **Implemented Functions (well-designed, written and documented)** |  |
| **Aum Rasikbhai Parsana** | **Implemented Functions** |  |
| **Rutarj Mrushad Shah** | **Implemented Functions** |  |
| **Jubril Olawale Akolade** | **SCRUM REPORT** |  |

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** |  |
| **Impact on Project** |  |
| **Solution or work-around** |  |
|  |  |
| **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 |
| --- | --- | --- |
| Test Automation and Milestone Progress | **In this meeting, the team discussed the completion of the milestone tasks, which include implementing functions, blackbox tests, executing blackbox tests, and creating new issues for any failed tests** | **completed the milestone tasks related to functions, blackbox tests, and the setup of test automation** |
| Function implementation | **Planning of function implementation** | **Successfully created the functions** |
| black box testing | **creation of black box testing to ensure thorough testing of the code** | **Successfully Implemented** |
| Handling Failed Test Cases and Creating New Issues | **process of handling failed test cases and creating new issues** | **Documented and handled everything correctly** |
| White box testing | **writing and store of the tests** | **Implemented and executed** |

**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 |
| --- | --- |
| Milestone Completion Strategy | Focus on completing the milestone tasks in a structured manner |
| Black box testing plan | Black Box Testing approach to ensure correct testing from an end-user side |
| Jira assignments | Assigning issues in Jira to each group members respectively |
| Discussion of the plan and deadlines | The discussion of the plan and deadlines was important in order to have a clear view of what needs to be done and when |

**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? |
| --- | --- | --- | --- |
| Iraklis Tsanachtsidis | **SCRUM REPORT** | **1 hr** | **Yes** |
| Frank Prerez | **Jira assignments** | **55 min** | **Yes** |
| Tarun Thomas | **Debugging (bugs fixed)** | **1hr 50 min** | **Yes** |
| Faaz Sherwani | **Well-designed, written and documented function** | **2 hr 25 min** | **Yes** |
| Aum Rasikbhai Parsana | **Implemented Functions** | **2 hr** | **Yes** |
| Rutarj Mrushad Shah | **Implemented Functions** | **2 hr** | **Yes** |
| Jubril Olawale Akolade | **SCRUM REPORT** | **1 hr** | **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 |
| --- | --- |
| Faaz Sherwani &  Tarun Thomas | Implementing functions and white box testing |
| Jubril Olawale Akolade  & Iraklis Tsanachtsidis | Review the code again and do the SCRUM report |
| Frank Prerez, Tarun Thomas  & Rutarj Mrushad Shah | Implementing functions, black box testing and Jira |

**Major Outcomes of Meeting:**

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

| Outcome | Impact on Project |
| --- | --- |
| Issue Tracking and team coordination towards fixing it | **Good and coordinated project** |
| Clear decision and goal making | **Effective decision-making and promotes alignment within the team.** |
| Black Box Testing Approach | **ensuring the system's functionality and user experience are validated.** |
| Milestone Progress and Deadlines | **Progress of milestone tasks and deadlines identified** |

**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 |
| --- | --- |
| Organization and planning | **Great team coordination** |
| Successful meeting held and discussion of the assigned work | **Effort from all the members** |
| Communication and Transparency | **Open communication and discussion about each members’ tasks** |
| Function implementation and prioritization | **Understanding and implementation of the functions in a timely manner** |

**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 |
| --- | --- |
| Nothing in particular, everything went according to the group’s plan |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 test results in both the original test document and Jira is a good idea because it provides redundancy and ensures comprehensive tracking and visibility of the testing process and its outcomes. Here's why it's beneficial:

a. Test Documentation: The original test document contains detailed information about the test cases, their objectives, and the expected outcomes. By recording the results in this document, the testing team can easily refer back to it for historical context, analysis, and reporting purposes.

b. Issue Tracking and Collaboration: Jira is a popular issue and project tracking tool used in many software development teams. By logging the test results in Jira, any issues or bugs discovered during testing can be quickly documented and assigned to relevant team members for resolution. This facilitates effective collaboration among team members and ensures that all identified issues are appropriately addressed.

c. Centralized Reporting: Recording results in both places helps in centralizing the test data, making it easier for stakeholders and management to access and review the testing progress and outcomes. This centralized reporting also aids in making data-driven decisions and tracking the overall health of the project.

d. Audit and Compliance: Some projects may require adherence to specific testing standards or regulatory guidelines. Keeping records in multiple places can provide better support for audit purposes, ensuring that testing activities are carried out correctly and in line with the required standards.

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

a. Code Maturity: By the fourth milestone, a significant portion of the code may have been implemented and stabilized. Writing whitebox tests at this stage ensures that the codebase is mature enough to be thoroughly analyzed and tested based on its internal logic.

b. Code Availability: Writing whitebox tests requires access to the actual source code. In the early stages of development, the code might not be fully available, or it could be undergoing frequent changes. Waiting until later milestones ensures that the code is more stable and accessible for in-depth testing.

c. Testing Priorities: In the initial stages, the primary focus is often on validating the functionality and user experience through blackbox testing. Once the core features are tested and confirmed to be working as expected, attention can shift to testing the internal logic and edge cases using whitebox testing.

1. 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.

More Blackbox Tests: Blackbox tests tend to dominate when validating the external behavior of a function or component. These tests are designed to cover various input scenarios and expected outputs based on the function's specifications. Blackbox tests are more concerned with the observable behavior of the function and how it interacts with other components, making them valuable for functionality and integration testing.

Fewer Whitebox Tests: Whitebox tests are typically focused on testing the internal logic, paths, and code structure of a function. They are particularly useful for coverage of specific code branches, boundary conditions, and code optimization. However, writing comprehensive whitebox tests may require more effort, and they may not be necessary for every function, especially if blackbox tests provide sufficient coverage of the function's behavior.

The ratio of blackbox to whitebox tests can also be influenced by the project's testing strategy, time constraints, and the criticality of the function in the overall system.

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

An automation hook for GIT is a script or program that gets executed automatically whenever specific actions occur in a GIT repository, such as code commits or pushes. The purpose of the automation hook is to trigger specific automated processes, tests, or actions, providing several benefits that can significantly improve the quality of the software project.

Here's how automation hooks for GIT can improve software quality:

a. Continuous Integration (CI): Automation hooks are commonly used in CI systems to automatically trigger a series of build, test, and deployment processes whenever new code changes are pushed to the repository. CI helps detect integration issues early, identify bugs, and ensures that code changes are promptly validated against existing codebase and tests.

b. Automated Testing: Automation hooks can initiate automated test suites, including unit tests, integration tests, and even end-to-end tests. Automated testing helps ensure that the code changes do not introduce regressions and that the software behaves as expected.

c. Code Review and Quality Gates: Automation hooks can enforce certain quality gates, such as running static code analysis tools, before accepting code changes. This enforces coding standards, identifies potential code issues, and improves overall code quality.

d. Deployment Automation: Automation hooks can trigger deployment processes to staging or production environments after successful testing and code review. Automated deployments reduce the chances of human error during the deployment process, leading to more reliable and consistent releases.

e. Faster Feedback Loop: By automating various processes through hooks, developers receive faster feedback on their code changes. This enables quick identification and resolution of issues, reducing the time between code changes and bug fixes.

f. Consistency and Reliability: Automation hooks ensure that essential processes are executed consistently for every code change, reducing the risk of missing critical steps during development, testing, and deployment.