# **MILESTONE 4** -- SFT221 SCRUM Report and Reflection

All students are expected to attend the SCRUM meetings and to participate. Failure to do so will result in greatly reduced grades.

**GROUP**: \_\_\_\_\_\_\_\_\_\_\_11\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1.Richa Koirala | 4. Ammar Ganjat |
| 2. Akansha | 5. |
| 3. Manjot Singh | 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 (in an Excel file) 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 (in the C++ testing project), stored in repository, executed, results in Jira and on corresponding test documents, and debugged (at least 1 SET is required).
* 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) | 10% |
| Finish coding blackbox code (well-designed, written, and documented) | 5% |
| Whitebox test case document (well written, complete, good test data) | 10% |
| Whitebox test code (well designed and documented) | 20% |
| Updated requirements traceability matrix | 5% |
| Test execution (performed, results recorded, issues created) | 10% |
| 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 | 10% |
| **Deadline** | 20% deduction for each day you are late |  |

**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** |
| **Richa Koirala** | **SCRUM , finish black box test code (Unit test).** | **N/A** |
|  |  |  |
| **Akansha** | **SCRUM, Function implementation, update black box document.** | **N/A** |
|  |  |  |
| **Ammar** | **SCRUM,** **Function implementation, Requirements traceability matrix.** | **N/A** |
|  |  |  |
| **Manjot** | **SCRUM,** **White box test documentation, White box test code (Unit test)** | **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** |  |
| **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 |
| Function implementation | **About implementation of function specs that was developed in cargo.h in last week** | **Finished implementation** |
| Testing function | **Discussion about testing function (black box testing and white box testing)** | **Testing function written** |
| Jira | **Task schedule setup in Jira** | **Completed** |
|  |  |  |
| Git | **Each branch git update** | **Completed** |
|  |  |  |
| SCRUM | **SCRUM done** | **SCRUM finished** |

**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 |
| Prioritizing tasks | Equal amount of work assigned to each member of team |
|  |  |
| White box testing | About implementation of white box test cases |
|  |  |
| Black box testing | About implementation of white box test cases |
|  |  |
| Function implementations | Implementation of different 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? |
| ALL | **Brief discussion on black box and white box testing and implementation** | **1 hr** | **Yes** |
|  |  |  |  |
| ALL | **Brief discussion on function implementation in the program** | **1 hr** | **Yes** |
|  |  |  |  |
| ALL | **Scrum report** | **30min** | **Yes** |
| ALL | **Discussion on hook automation** | **30min** | **Yes** |
| ALL | **Jira and GitHub** | **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 |
|  |  |
|  |  |
| Akansha | Looking after GitHub |
|  |  |
| Richa | Manjot will conduct the meeting with tasks for milestone 4 |
|  |  |
| Manjot | Matching the availability and get a time schedule for the meeting in person and the location for the meeting. |
|  |  |
| Ammar | Looking after Jira |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Test plan | **Blackbox testing ad white box testing and its implementation** |
|  |  |
| Function implementation | **Function implementation was done according to function specs that was written** |
|  |  |
| Attendance | **Everyone attended the meeting** |
|  |  |
|  |  |

**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 | **Usefulness for version control** |
|  |  |
| SCRUM | **All contributed** |
|  |  |
| Meeting | **All 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 |
| 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.

After we run our Blackbox and Whitebox tests we are asked to record the results in both test documents as well as in Jira. It is a good idea to record the results in both places because:

1. Recording helps in ensuring that there is a comprehensive and centralized record of test performed which also includes detailing test cases, inputs, expected results and other issues.
2. To capture the evolution of the testing process over time, recording the result will play the role of historical references which will help later for different versions of the software.
3. Jira is a popular management tool so recording in Jira can allow team members to get access to test information when it is needed.
4. Also having test results stored as records helps to ensure redundancy.
5. Why did we wait until the fourth milestone to write the whitebox tests?  
     
   We waited till our fourth milestone to write the Whitebox tests due to several factors. In our opinion, validation of the software from a black box perspective as testing software without knowing its internal structure could have a problem. This approach can simulate the interaction of real users and software and identify any issues if possible. Also, another factor is the case which in initial stage the codebase often can go through changes and improvements. Also, waiting till out fourth milestone our code was more stable and it also went through changes which made it easier for us to write precise tests to see how code is working.
6. 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.

As we know, black box testing is like observing our machine from outside without being aware of what’s inside. Like for mapping.c we are testing this function without being aware of its internal details as our goal is to check if out functions behaves correctly based on inputs and outputs we are expecting. This approach helps us to ensure that the function is doing what it is supposed to do according to the requirements without needing to be understanding its coding internally.

Similarly, white box testing is like observing our machine being aware of what’s inside. As for cargo.c we are dividing its internal logic and implementation of code. We are examining the pathway and decision inside of the code checking if it works correctly and efficiently. This type of testing is crucial for newer function because it helps us to catch bugs, make the code run faster by checking its internal workings.

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 purpose is so that incorrect and bugged code is not pushed into the repository. The automation ensures that only code that is tested and has passed all the tests can be pushed. This improves the quality of the software as it lets us focus on making sure all the working components of the project can work together, instead of focusing on why individual functions and pieces of code don't work on their own.