# **Milestone 4 Scrum Report**

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

**GROUP**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_8\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |
| --- |
| Kate De Leon |
| Jarod Jian Kang Hery Chen |
| Ronak Jung Rayamajhi |
| Carson Ji |
| Kemono Onomek |

## Milestone 4 Tasks

* 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) | 10% |
| Updated requirements traceability matrix | 10% |
| 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) | 15% |
| 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** |
| Kate De Leon | * Blackbox test case and documentation * Whitebox test case and documentation | * None |
| Jarod Jian Kang Hery Chen | * Updated Traceability Matrix to map test cases to corresponding requirements of the project. * Completed Blackbox test code and documentation * Completed White box test code and documentation | * None |
| Ronak Jung Rayamajhi | * Implemented all the functions we created till now and updated the main. * Created hook files and implemented in the project * Updated Jira and GitHub repository control * Debugged test code for finalization * Scrum report | * None |
| Carson Ji | * Completed Blackbox test code and documentation * Completed White box test code and documentation | * None |
| Kemono Onomek | * Blackbox test case and documentation * Whitebox test case and documentation | * 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** |  |
| **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 |
| Whitebox testing | Discussed about the Whitebox testing and the functions we created. | Completed WhiteBox testing |
| Function specs implementation | Discussed about any issue that came/might come up during implementation of the functions | Implementation completed |
| Scrum report | About completing the report | Scrum completed |
| Jira | All tasks were updated | Done |
| GitHub | Updated the repository to current progress | Done |

**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 |
| Task Priority | All the members were assigned with equal amount of work. |
| White Box testing | Discussed and fixed any issues the member were having regarding White Box Testing |
| Black Box testing | Some issues from previous week were fixed. |
| Function implementation | All the new functions were implemented. |
|  |  |
|  |  |

**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? |
| Everyone | Analysis of debugging Black Box code from last week and implementing properly | 1 hour | Yes |
| Everyone | Discussion about White Box testing and implementation | 1 hours | Yes |
| Everyone | Scrum report discussion | 30 mins | Yes |
| Everyone | Discussion on hook automation | 20 mins | Yes |
| Everyone | Jira project and GitHub repository update | 20 mins | Yes |
| Everyone | Discussion for next week tasks | 15 mins | 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 |
| Ronak Jung Rayamajhi | Jira and GitHub repository management, Scrum report |
| Everyone | Meeting for next milestone |
| Everyone | Acceptance tests and Integration tests |
| Everyone | Debugging and Test Execution |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Function implementation | Function implementation was done as per the function specs that was done in the last week |
| Black Box testing | Some bugs were fixed, and all the Black Box test code and test cases are completed and documented. |
| White Box testing | White Box testing was implemented |
| Hook Implementation | Hook files were created and used before pushing the changes to the GitHub |

**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 Division Discussion | Active participation of all the members |
| Function Implementation Discussion | Active participation of all the members |
| Test Cases Documents Creation Discussion | Active participation of all the members |
| Updating traceability Matrix Discussion | Active participation of all the members |

**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 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Reflections**:

Answer the following questions using your own words. Make sure that each answer comprises a minimum of 100 words.

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

We waited until the fourth milestone to write whitebox tests because of the stages in software development. First, we plan, then build a base, implement the plan, and finally test the product. By milestone three, we have set up the base, created a test plan, implemented functions, and made the program executable. This allows us to deeply test the program's functionality. At this stage, we can explore different parts of the code, test them one by one, and fix any errors. To test the program, it needs to run and show basic functionality. We achieve this by milestone three, so we write whitebox tests in milestone four.

1. How does the Agile methodology ensure that all team members are consistently engaged throughout the software development process, avoiding downtime due to dependencies on others? Provide an example to illustrate your point.

The Agile methodology ensures that all team members are consistently engaged throughout the software development process by promoting continuous collaboration communication, and progress in the work. We, ourselves, can be taken as an example to prove this point.

We had regular meetings where team members share what they are currently working on and if they are facing any issues or not. This keeps everyone informed and helps identify and resolve dependencies quickly. Agile methodology divides the project into small parts with each member getting a specific goal which allows team members to focus on delivering a piece of functionality within a short keeping everyone actively involved. Agile teams are typically composed of members with various skills and in our case were developers, testers, and a leader. This reduces dependency on specific individuals because team members can help each other out and ensure continuous progress. Regularly reviewing the code ensures that there is always a clear list of tasks ready to be picked up by team members. This helps avoid downtime as there is always something to work on.

1. What is a shell script and how are we going to utilize a hook script in this project?

A shell script is a text file containing a sequence of commands for a UNIX-based shell to execute. It can automate tasks such as file manipulation, program execution, and printing text. A hook script is a type of shell script that is triggered by specific events in version control systems (like Git). In this project, we have used hook scripts to automate testing by setting up a pre-push hook. A pre-push hook script can ensure that all local commits pass tests before being pushed to a remote repository. This helps maintain code quality in shared repositories. Hook scripts catch issues early in the development process, reducing the likelihood of introducing bugs into the main codebase and automation reduces testing time compared to when it was done manually.