# **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 B**

**Members Present**:

|  |  |
| --- | --- |
| 1. Aung Moe Thwe | 4. Thiri Aung |
| 2. Jhonatan Lopez Olguin | 5. |
| 3. Kashish Verma | 6. |

## 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** |
| **Aung Moe Thwe** | Completed function implementation and testing | N/A |
| **Kashish Verma** | Developed Blackbox test cases and stored in the repository | N/A |
| **Thiri Aung** | Implemented and documented Blackbox tests | N/A |
| **Jhonathan** **Lopez Olguin** | Updated requirements traceability matrix | N/A |
| **All** | Developed functions and documented. Scrum Report | N/A |
| **Aung Moe Thwe** | Execution of unit test cases and a function | 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 |
| Function Specifications | |  | | --- | |  |  |  | | --- | | Discussed the final implementation details and documented them in the header file. | | All members finalized and stored the function specifications in the repo. |
|  |  |  |
| Black Box Testing | Refined test cases and discussed additional scenarios for validation. | Team members continued developing and refining test cases. |
| Whitebox Testing | Implemented whitebox tests and discussed results and debugging strategies. | At least one set of whitebox tests fully implemented and stored in the repo. |
| Traceability Matrix | Updated the matrix to ensure full coverage of all requirements and functions. | Updated matrix stored in the repository. |

**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 |
| Finalize Function Specifications | To ensure all team members have a clear and consistent understanding of each function’s purpose, inputs, and outputs. |
| Expand Black Box and Whitebox Testing | To create comprehensive sets of test cases that cover all possible scenarios ensuring robust validation of functions. |
| Implement and Document Whitebox Tests | To establish a dedicated testing environment within the GitHub repository allowing for systematic and automated testing of functions. |
| Maintain Traceability Matrix | To ensure that all requirements are adequately tested and any gaps are identified and addressed. |
|  |  |

**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? |
| Aung Moe Thwe | Function implementation and testing |  | No. Took some more time after meeting |
| Kashish Verma | Developing blackbox test cases | **15** | Yes. |
| Thiri Aung | Implementing and documenting whitebox tests | **25** | Yes. |
| Jhonatan Lopez Olguin | Updating requirements traceability matrix | **20** | Yes. |
| All | Reflection | **15** | Yes |
| All | Finding bugs and fixing it |  | No. This also required some more time |
|  |  |  |  |

**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 |
| Kashish | |  | | --- | |  |  |  | | --- | | Finalize and add more black box test cases to the repository. |   Do Reflection |
| Aung Moe Thwe | Implement additional test cases in the unit test project and store in the repo. Fix Bugs. Integration Testing |
| Jhonathan | |  | | --- | |  |  |  | | --- | | Continue with the project research and documentation |   . Update Traceability Matrix |
| Thiri | |  | | --- | |  |  |  | | --- | | Finalize whitebox tests and document results |   . Do more testing / acceptance |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Re-check-ins scheduled | Facilitates continuous progress and early issue resolution. |
| Brainstorming | Generating and discussing ideas, solutions, and strategies for testing cases. |
| Unit test project and white box test cases implementation | Allows for the testing of functions to ensure they meet specifications. |
| Traceability matrix updated | Helps map test cases to requirements ensuring full coverage. |
|  |  |

**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 |
| Collaborative Function Specification | All members actively participated in creating detailed function specifications. |
| Effective Test Case Development | Test cases were well thought out and documented. |
| Whitebox Testing Implementation | Successful implementation and debugging of whitebox tests. |
| Traceability Matrix Update | Ensured good coverage of project requirements. |
| Task Breakdown | Members were clear about their responsibilities and efficiently managed their tasks. |
|  |  |
|  |  |

**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 | Everything went smoothly in this meeting. |
|  |  |

**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 white box tests?  
     
   Whitebox tests are often done later in the development process because they need a good grasp of the code's inner workings, which is easier to achieve after the main functions and initial black box tests are in place. By the fourth milestone, our code is more stable, which makes it a better time to write detailed white box tests. These tests help us check each part of the code thoroughly and find any deep-seated issues that black box tests might miss. This step-by-step approach helps us avoid having to redo a lot of work and ensures our code is solid.
2. 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.  
     
   Agile methodology focuses on teamwork, making progress in small steps, and getting regular feedback. It keeps everyone involved through daily stand-up meetings, sprint planning, and reviews. These practices help everyone know what’s happening in the project and what their tasks are. For instance, while one team member is writing a function, others can create test cases, update documents, or work on other features. This way, no one is stuck waiting for someone else to finish their part. This method keeps the work moving smoothly and allows the team to adjust quickly to any changes, promoting a collaborative and efficient work environment. For example, if Aung is developing a new feature, Thiri can simultaneously work on testing that feature, and another team member can update the project documentation. This method keeps the work moving smoothly and allows the team to adjust quickly to any changes, promoting a collaborative and efficient work environment.
3. What is a shell script and how are we going to utilize a hook script in this project?

A shell script is a program you write for the command line of an operating system. It helps automate tasks by running a series of commands. In our project, we will use hook scripts, which are special scripts that Git can run at different stages of the repository's life cycle. For example, a pre-commit hook can automatically run tests or check the code formatting before you finalize a commit. This ensures that only good-quality, error-free code gets added to the repository. This automation helps keep the code quality high and reduces the amount of manual work needed.