# **Milestone 3 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 3 Tasks

In this milestone you will create issues to design the functions, design all of the functions you need to complete the project and store the specifications in the repository. As soon as the specifications start to be produced, you can start to design the blackbox tests (what they test, how to perform them and test data). Once tests are written, they can be implemented and added to the repository and any team members not otherwise busy can start to implement the functions. You will also build a function-test matrix that shows the blackbox tests for each function. This will be maintained through the testing cycle as new tests are added.

**Deliverables due 4 days after your lab day:**

* A set of AT LEAST 4 function specifications added to a new header file and stored in the repository.
* A set of blackbox 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.
* **Create and add a C++ testing project to your solution.**
* Start writing blackbox test code (for the functions above) and store in repository (at least 1 is required for this milestone).
* Start implementing the functions and store them in repository (optional).
* A requirements traceability matrix added to the repository and shows the mapping between the requirements and test cases.
* Updated Jira project to show activities and progress.
* Completed scrum report including reflection questions answered.

**Rubric:**

|  |  |  |
| --- | --- | --- |
| **Individual** | Group participation (includes GitHub commits and Jira usage) | 80% |
| Teamwork | 20% |
| **Group** | Function specifications (documented, complete, well-written, added to the project) | 10% |
| Blackbox test cases document (well-written, complete, good test data) | 10% |
| Blackbox test code (in the C++ project) well-designed and documented | 10% |
| Functions implementation (coded in the C project & well documented) | 15% |
| Requirements traceability matrix (complete and added to GitHub) | 15% |
| Git usage (used properly with good structure) | 10% |
| Jira usage (creates issues, tracks progress) | 15% |
| Scrum report & reflections | 15% |
| **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** |
| Member: 2,3 | Created function specifications, set up GitHub repository | N/A |
| Member : 2,3,4 | Developed initial black box test cases | N/A |
|  |  |  |
| Leader | Implemented unit test project setup in GitHub repository | N/A |
| Member: 4 | Started populating the traceability matrix | 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 creation of function specifications and documented them in the header file. | All members are tasked with finalizing and storing the function specifications in the repository. |
| Black Box Testing | Created initial test cases and discussed test data and expected results for each function. | Team members will continue developing and refining test cases, ensuring they are stored in the repository. |
| Unit Test Project | Set up unit test project in GitHub and started implementing test cases. | At least one test case will be fully implemented and stored in the repository. |
| Traceability Matrix | Began populating the traceability matrix to ensure coverage of all requirements. | Update the matrix and store in 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 |
| Document and Finalize Function Specifications | To ensure all team members have a clear and consistent understanding of each function’s purpose, inputs, and outputs, facilitating smooth implementation and testing. |
| Expand Black Box Testing | To create a comprehensive set of test cases that cover all possible scenarios, ensuring robust validation of function performance and correctness. |
| Implement Unit Test Project | To establish a dedicated testing environment within the GitHub repository, allowing for systematic and automated testing of functions. |
| Develop and Maintain Traceability Matrix: | To create a clear mapping between project requirements and test cases, ensuring 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? |
| ALL | Created function specifications and documented them in header file | 1.5 hours | Yes |
| ALL | Developed initial black box test cases | 2 hours | Yes |
| Aung Moe Thwe | |  | | --- | | Implemented unit test project setup in GitHub repository |  |  | | --- | |  | | 1 hour | Yes |
| ALL | Started populating the traceability matrix | 2 hours | Yes |
| ALL | Reflection | 30 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 |
| ALL | Meeting on 20th July, Saturday 3pm to 5pm |
| ALL | Finalize and add more black box test cases to the repository. |
| ALL | Implement additional test cases in the unit test project and store in the repository. |
| ALL | Continue with the project research and documentation |
| ALL | Do function implementations of methods created |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 set up | Allows for the testing of functions to ensure they meet specifications. |
| Traceability matrix started | 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 | Initial black box test cases were well thought out and documented. |
| Unit Test Project Setup | Successful setup of unit test project in GitHub repository |
| Traceability Matrix Initiation | Started populating the traceability matrix, ensuring good coverage of project requirements. |
| Task Breakdown | Members were clear about their responsibilities |
|  |  |
|  |  |

**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. What is the difference between blackbox tests cases and blackbox test code? Explain how we use assertion in Visual Studio to execute tests.

Black box test cases are detailed plans that outline input data, expected results, and steps for testing a function without examining its internal code, while black box test code is the actual implementation of these test cases. In Visual Studio, we use Assert: statements to check if the function's actual output matches the expected output; if the assertion fails, the test fails, helping identify issues in the code.

Assert:: AreEqual(expected, result);

1. How can a traceability matrix help in the testing process?

The traceability matrix helps us to improve the testing process as it provides full requirement coverage and efficient test case management. It also makes the validating and defect tracking easier and more efficient by enabling testers to connect each requirement to a corresponding test case. Moreover, this matrix ensures that new criteria are properly assessed with the help of a supporting change impact study. By serving as a communication tool for the development, testing, and business teams, the traceability matrix encourages teamwork and ensures higher product quality which reduces the possibility of having errors in the final program. 

1. Write down two of the function prototypes you submitted. Why did do you need each one of them and how will each one help you achieve the project needs?

struct Capacity calculateRemainingCapacity(struct Truck truck);

We need this function to keep track of the available capacity on the truck as in this project we can only load the shipments on the available trucks only. Therefore, by including this function in the program, we can figure out which truck has enough space to collect the new shipment and which truck is full.

int loadShipment(struct Truck\* truck, struct Shipment shipment);

This function is added to get the updated amount of shipment that is already in the truck so it can help calculate the available capacity. This function will update the amount of shipment in the truck from time to time after every shipment loading.