# **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**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1. Mihyeon Park | 4. Manraj Singh |
| 2. Ana Masoumi | 5. Veronika Edith Turpo Meneses |
| 3. Hamzeh Khaled Nayef Muhiar | 6. Yashleen Brar |

## 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** |
| Ana Masoumi | Helped with the sruct.h function declaration , black box testing and scrum report Q3. | N/A |
| Mihyeon Park | Worked on struct.h function specifications, created four Blackbox test cases for one function, filled traceability matrix, filled scrum report. | N/A |
| Yashleen Brar | Helped with the struct.h function declaration , black box testing and reflection Q1 and 2 in scrum report. | N/A |
| Manraj Singh | Helped with the struct.h function declaration , black box testing, and filled the matrix and helped in scrum report. | N/A |
| Hamzeh Muhiar | Helped create C++ test projects, filled in scrum report ms3, filled in traceability matrix R006. | N/A |
| Veronika Turpo | Helped fix multiple compilation and incompatibility issues and filled in matrix 07. | 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 prototypes | Creating six function prototypes | Created and updated in struct.h |
| Function specifications | Adding function specifications for function prototypes | Added function specifications, updated in struct.h |
| Blackbox Test Cases Document | Creating four sets of Blackbox test cases for six functions | Each created four sets of Blackbox Test Cases |
| Jira | Using and being more active on Jira | Created tokens and commented more on updates |
| Scrum Report | Filling scrum report and working on reflections | Filled and reflections filled |
|  |  |  |
|  |  |  |

**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 |
| Testing Decision | Each member working on testing for each function |
| Prioritization of tasks | Equal amount of works assigned to each member of team. |
| PENDING TASKS | Sone tasks were left from last milestone, completing them is a duty of everyone. |
|  |  |
|  |  |
|  |  |
|  |  |

**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 | **Jira contribution** | **30min** | **Completed** |
| Everyone | **Created Jira tasks** | **30mins** | **Completed** |
| Everyone | **Filled scrum report** | **1 hour** | **Filled** |
| Everyone | **Discussed on function prototypes and specifications** | **30mins** | **Completed** |

**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 |
| Everyone | Jira Usage |
| Everyone | Scrum report, reflection |
| Everyone | Finish implementing/coding the functions |
| Everyone | Create sets of Whitebox test cases |
| Everyone | Implement Whitebox test cases and store in repo |
| Everyone | Update traceability matrix |
| Everyone | Complete hook file for test automation |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Function Prototypes | **Discussed on creating function prototypes which helped with the Blackbox test cases** |
| More Jira Usage | **Discussed on updating more on Jira so we can see progress** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Scrum report | **Everyone Contributed** |
| Jira | **More Jira Usage** |
| Git | **More Git contributions** |
| Documentations | **Completed** |
| Scrum report | **Everyone Contributed** |
|  |  |
|  |  |

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

**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.  
     
   The functionality of software is tested using Blackbox test cases, which are high-level descriptions of scenarios that are designed to function without considering the internal code. Upon the basis of requirements, they concentrate on the inputs and anticipated outputs. For instance, a test case may require the user to enter a valid username and password to confirm a successful login.  
     
   In contrast, Blackbox test code is the actual implementation of these test cases using a testing framework and programming language. The tests are automated by this code, which provides inputs, executes the software, and compares the actual results with the expected outcomes using assertions. For instance, in C# with NUnit, assertions such as Assert.AreEqual(expected, actual) are employed to verify outcomes.

To confirm that the software functions as anticipated, assertions are implemented within test methods in Visual Studio. Assertions that are similar to "Assert."AreEqual(expected, actual) or Assert.IsTrue(condition) is a frequently used expression. The test fails if the actual output does not match the expected output, as these assertions verify. This is accomplished in a test project, where test methods are defined with attributes such as [TestMethod] for MSTest or [Test] for NUnit. The Test Explorer in Visual Studio is employed to execute tests, which displays the results and specifics of each test, indicating whether it was successful or unsuccessful.

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

As it maps and traces user requirements with test cases, a traceability matrix is a potent tool in the testing process. This guarantees that all requirements are addressed by test cases, thereby improving the overall coverage of the test. It aids in the early detection of any gaps in the test coverage or missing functionalities involved in the testing process. It ensures that any modifications to requirements are adequately tested by providing a clear overview of the relationships between requirements and tests, which facilitates impact analysis of changes. Furthermore, a traceability matrix facilitates improved project management by documenting the testing process, ensuring compliance with standards, and establishing a foundation for test planning and reporting.

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?  
     
    The two prototypes that are useful for this project are:

1.`checkBuilding(const struct Route route, const struct Map\* map)`

2.`validShipment(struct Shipment shipment)`.

The `checkBuilding` function is necessary for verifying if the destination building is accessible via a given route on the map. This makes that the truck can reach the intended delivery point,by this function we can make sure that the destination is a building and not an empty road.

The function contributes to accurate route planning and reliable delivery operations.

The `validShipment` function is necessary for checking that the shipment doesn’t exceed the limitation of the trucks for size and weight.

By validating the shipment before loading, this function helps to prevent overloading the truck and makes sure that only valid shipments are accepted for delivery. This validation process is crucial for maintaining the truck's safety and operational efficiency. Both functions are integral to achieving the project's objective of efficient and reliable shipment delivery by ensuring that deliveries are feasible and compliant with logistical constraints.