# **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**: \_\_\_\_\_\_D\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1. Manas Gandotra | 4. Gurmehal Kaur Uppal |
| 2. Ajaypratap Singh Maan | 5. |
| 3. Arshnoor Kaur | 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** |
| **Manas Gandotra** | **Overviewed the Scrum Report and participated in problem analysis.** | **No** |
| **GURMEHAK KAUR UPPAL** | **Analyze the code and create BlackBox test cases along with their implementation** | **No** |
| **ARSHNOOR KAUR** | **Overviewed reflections and traceability Matirx** | **No** |
| **AJAYPARTAP SINGH MAAN** | **Created function definitions and implemented in the project code** | **No** |
|  |  |  |
|  |  |  |
|  |  |  |

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 |
| Analysis of MS-3 | **Went through the requirements of MS3 creating plans for execution of deliverables on time** | **The analysis was executed successfully** |
| Addition of functions in header files along with their implementation | **A series of function which are supposed to be needed for proper execution of the assignment** | **Successfully added** |
| Completion of Scrum report | **The reflection questions were discussed and updated with apt responses** | **Scrum Report complete** |
| Blackbox testing | **A series of test cases to be developed and implemented later on to discuss the outcome** | **Testing complete** |
| Traceability Matrix | **Keeping a record of the aim of individual test cases and if it/they achieve it** | **Matrix updated** |
|  |  |  |
|  |  |  |

**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 |
| All the deliverables should be ready 6 hours prior to the deadline | The submission of deliverables prior to the deadline ensures no errors make to the final submission through the process of error checking |
| Any possible delays or inability for completion of assigned tasks must be reported in advance | If the delays are informed prior, it gives time for other team members to come up with a solution or work together on the problem as soon as possible |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| ARSHNOOR KAUR | **Overviewed the milestone 3 pdf thoroughly, reflection discussed** | **30 MIN** | **Yes** |
| GURMEHAK KAUR UPPAL | **Overviewed the milestone 3 pdf thoroughly, deliverables discussed** | **30 MIN** | **Yes** |
| MANAS GANDOTRA | **Overviewed the milestone 3 pdf thoroughly, Scrum Report discussed** | **30 MIN** | **Yes** |
| AJAYPARTAP SINGH MAAN | **Overviewed the milestone 3 pdf thoroughly, Jira management discussed** | **30 MIN** | **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 |
| Arshnoor Kaur | Reflection + Scrum report |
| Manas Gandotra | Whitebox test data + traceability matrix + hook file |
| Ajaypratap Singh Maan | Whitebox implementation |
| Gurmehak Kaur Uppal | Completion of Blackbox + function implementations |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Milestone 3 towards completion | **Deliverables for MS3 successfully submitted** |
| MILESTONE 3 TASKS ASSIGNED TO EACH MEMBER | **Milestone 3 will be completed before time** |
| THE PROJECT REQUIREMENTS UNDERSTOOD IN DETAIL | **Each member is aware of the details of the assignment for smooth execution of tasks** |
|  |  |
|  |  |
|  |  |
|  |  |

**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 participated in the discussion of reflection questions** |
| Jira task assignment | **Each member chose their roles and were willing to co-operate if a delay in any deliverable occurs** |
| Blackbox testing | **The test data for the functions was thoroughly discussed and agreed upon** |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Everything went well | **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 black box test cases and black box test codes? Explain how we use assertion in Visual Studio to execute tests.**

**ANSWER:**

**Black Box Test Cases**: The test cases that are primarily designed for the testing of software externally. In forming test cases, the requirements that a software has to meet are taken into consideration. In black box testing, one does not know what is going inside the function but is just aware about the input values into the function and the output that it generates. This is one of the major portions which comes into play when one has to test the function/software fully.

**Black Box Test Code**: The test code comes into play when one has to perform the black box testing. The code that is used to perform the black box testing is called the Black Box Test Code. It can be written in any language but until this course, we are implementing C++ to perform testing using assertion functions.

So, the black box test cases are crafted to check if the software behaves as expected based on the requirements. Whereas the test code is written using testing frameworks and tools such as native unit tests in Visual Studio. This is done to check if the output generated by a test code matches with the expected output or not.

The assertions in Visual Studio have been primarily used to test the code snippets to see if they prove conditions to be true or not. While using the assertion functions, we provide certain input to the code snippets and validate the outcome of that test against an expected result which returns a Boolean value to indicate the test case result. So, an extra project is added to the visual studio along with the main project that is a native unit test project. Then we write the test cases inside the methods of the namespace used. While doing black box testing, we can use various assertion methods, such as Assert::IsTrue() which takes the function to be checked and the argument along with an expected to be checked against. We can run these tests using Test Explorer in Visual Studio which indicates the time the test cases have taken to get checked, if the test cases have passed or not. It also gives an error message if the test case has not passed, and a special message is to be printed.

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

**ANSWER:** The traceability matrix helps to ensure that all the business requirements that are crucial for the code snippets to pass effectively are covered by test cases. So, test coverage is the major portion that this traceabilitymatrix is checking. The matrix helps in getting the information about the test cases horizontally and vertically. Tests are presented vertically, and the horizontal representation of requirements provides a deep analysis of what is going on in the test cases analytically**.** This helps us to judge which test cases are fulfilling which requirement. Each requirement must be fulfilled by at least one test case and if in case, there is a test case/function which is not fulfilling any requirement, it is not meant to be one of the test cases. So, overall traceability provides an overview of all the requirements of the code, their requirements, and the function on which each test case is being performed, along with their interconnection with each other.

For example, in the traceability matrix we submitted, the function assignShipmentToTruck can pass the requirement number 1, 3, 5, and 6. That means this function can judge that the shipment should be effectively assigned to the most suitable truck, the system must plan and display each truck's route, and the package should always be placed in the truck that requires the shortest detour to deliver it, and if two trucks are the same distance away, put the package in the truck that has more space. Similarly, other requirements are checked by each one of the functions and this makes up 24 test cases checking distinct business requirements and denoting the superficial scenario of testing the functions.

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?**

**ANSWER**: The two examples out of the function prototypes we have submitted are:

1) void displayRouteToDestination(const struct Route\* route, const struct Point destination);

2) int isTruckCapacitySufficient(const struct Truck\* truck, const struct PackageInfo\* shipment);

These function prototypes have helped the most in determining the code flow as it was meant to be. When we were reviewing the code, we came to know that the most important step in the transportation process is the route being followed by a truck. For this, the first function mentioned above came into existence. This function displays the route a truck uses to deliver the package to its desired destination while taking into consideration the route and the destination as parameters. So, this function will help the path followers to follow the correct route and monitor the stage at which the delivery has been made with respect to the destination mark. In short, this function will play a crucial role in improving the efficiency of deliveries and addressing any route or destination issues that arise during commuting.

As far as the second function is concerned, it is very important to know if the capacity of the truck allows less or more load when the capacity of the truck is taken into consideration. This accounts for the weight and volume which is standard for each and every truck. This can be understood from the point that if the trucks are overloaded, it can lead to severe repercussions which has to be prevented at any cost. So, this function takes care of this thing while taking the truck structure, that is the truck being checked, and the shipment which includes the shipment details, that are size and weight. So, this verifies not only the truck’s capacity before assigning a shipment but also maintains a perfect balance of standard weight and volume set for a truck for shipment. Hence, this function will prevent overloading of the trucks and assign the load to only those trucks that can handle it.