# **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: 6**

**Members Present**:

1. Sahan Gallage
2. Ileperuma Gunarathna
3. Dhanuth Hennedige
4. Sean Li

## 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) | 10% |
| Visual Studio solution with 2 projects (complies and works) | 10% |
| Requirements traceability matrix (complete and added to GitHub) | 10% |
| 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** |
| **Dineth, Sahan** | **Function specifications and (blackbox) test descriptions** | **None** |
| **Dhanuth, Sean** | **Trace matrix and scrum report** | **None** |
| **All** | **Test project creation, black box tests code, and some of the functions, if required** | **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** | **N/A** |
| **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 |
| Scope of the milestone | What kinds of initial functions specifications and test package we need to hit the ground running, exactly how much deliverable content we need to create | **Completion of this milestone** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Basic functions and tests | Basic function specifications that covers the general functions of the program, and tests to verify that they work |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Dineth, Sahan | **Functions and test descriptions** | **8 Hours** | **Yes** |
| Dhanuth, Sean | **Trace matrix and scrum report** | **8 Hours** | **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 | Read milestone 4, and start creating more functions and prepare more tests on our own to contribute to the expedient completion of the project |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Discuss what’s need to be done for this milestone, as it’s has comparatively much more work compared to previous milestones | **Milestone understood, assigned tasks divided** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Milestone 3 | We know programming and have good chemistry |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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. How did your approach differ when creating test cases for blackbox testing versus whitebox testing? Reflect on the advantages and limitations of each method based on your experience with the assignment.  
     
   When creating blackbox tests we simply need to look at the functions from the user’s general perspective. However, for whitebox tests it’s more challenging. Mainly, what we assumed initially was that the shortestPath will simply generate the shortest path for each truck to a destination, taking the truck routes into account, but this was wrong. This misunderstanding was a catalyst for us to write a lot of the tests. Blackbox tests are more straightforward but do not show how things work in detail, while whitebox tests are more complicated but help us test each function critically, and writing whitebox tests show true understanding of the functions.
2. How did the traceability matrix help ensure that all functional specifications were adequately tested? Reflect on its role in maintaining comprehensive test coverage.  
     
   The matrix allows us to clearly display the correspondence between tests and the project requirement it fulfills, showing the rationale for each test. Once a sufficient number of checkmarks have been ticked under each requirement column, we can say for sure that we have comprehensive test coverage, while a lack of checkboxes will point to areas that need more testing. Ideally, we think there should be an evenly balanced number of checkmarks for each test row and requirement column as this shows that each test has been broken down into smaller units, and each requirement has just the right amount of testing.
3. 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?

float percentageVolumeFull(const struct Truck\* truck);

This function tells us how full a truck is based on how much space (volume) is used. Since each truck has a limit of 200 cubic meters, we need to know how much is already taken. It helps us decide if a new shipment can fit in the truck. When two trucks are close to the delivery location, we choose the one with more space left. This function helps us make that decision.

bool validateDestination(const struct Map\* map, const struct Point\* dest);

This function checks if the destination is a valid place to deliver. A valid destination must be inside a building (not a road or empty space) and inside the 25×25 map. It stops the program from trying to deliver to the wrong place. Before we send a truck or use the pathfinding function, we make sure the destination is correct. This keeps the system working properly and avoids errors.