# **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**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**3**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1. Kaitlyn Marino | 4. Grace Gabrielle |
| 2. Nolan Grossi | 5. |
| 3. Bilal Umar | 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) | 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** |
| **Kaitlyn** | **Test Plan No. 1-9 (testing strategy included)** |  |
| **Grace** | **Test Plan No. 10-17, Jira Project Update** |  |
| **Bilal** | **Scrum Report** |  |
| **Nolan** | **Code Analysis, Data Structs, adding mapping.c and mapping.h** |  |
|  |  |  |
|  |  |  |
|  |  |  |

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 |
| Adding at least 4 function specifications to new header file | **Discussed who would add 4 function specifications to the new header file** | **We decided that Bilal would make the 4 function specifications** |
| 4 Sets of Black box Tests for each function | **Discussed who would work on the black box tests for each function** | **We discussed that Bilal and Kaitlyn would divide the making of black box tests for each function (Bilal for the first half of function tests, Kaitlyn for the second half of the function tests)** |
| Creating C++ testing project | **Working on creating a C++ testing project for the black box tests** | **We decided Grace would create the C++ testing project, with help from Kaitlyn** |
| Writing black box test code and adding to GitHub | **Discussed who would write black box tests code (at least one)** | **We decided that Grace would write the black box test codes for one function** |
| Function Implementation | **Discussed who would start doing the function implementation for this week onwards** | **We decided that Nolan would start writing the function implementation** |
| Requirements Traceability Matrix | **Discussed how to make the requirements traceability matrix and who would be doing it** | **We decided Nolan and Grace would make the requirements traceability matrix** |
| Updated Jira Project | **Worked on updating the Jira Project tasks for each member** | **We decided Grace would do the Jira updates** |
| Scrum Report | **Discussed who would do the overall scrum report for this milestone** | **We decided Kaitlyn would do the scrum report, reflections included** |

**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 |
| Adding function specifications | Bilal added the function specifications |
| Writing 4 sets of black box tests for each function | Bilal and Kaitlyn divided the making of the black box test cases and put them in in the Word doc template |
| Creating C++ testing project and writing at least 1 black box test | Grace created the testing project, and wrote black box tests for one function |
| Function implementation | Nolan started implementing the functions |
| Requirements traceability matrix | Nolan and Grace worked on the requirements traceability matrix |
| Jira Project update | Grace updated the Jira project for milestone 3 |
| Scrum Report | Scrum report was completed in full by Kaitlyn, with exception of getting some information from other group members |

**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? |
| Kaitlyn | **Start of scrum report, start black box test cases** | **~1hr** |  |
| Bilal | **Function specifications, making of excel file and start of black box test cases** | **~1hr** |  |
| Nolan | **Function implementation, start of requirements traceability matrix** | **~1hr** |  |
| Grace | **Updated Jira project, start of requirements traceability matrix** | **~1hr** |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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 |
| Kaitlyn | Half of the white box tests in the document, part of the white box code implementation, and milestone 4 scrum report |
| Bilal | Finishing black box tests and function implementations |
| Nolan | Updating the requirements traceability matrix, part of the white box code implementation |
| Grace | Half of the white box tests in the document, part of the white box code implementation, and Jira updates |
| All members | Hook files for test automation |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Adding function specifications | **Successfully added function specs according to the data structure from milestone 2** |
| Created black box test cases | **Successfully created black box test cases for 6 functions** |
| Created requirements traceability matrix | **Successfully created the requirements traceability matrix for the testing in future milestones** |
| C++ testing file and black box test code | **Created the testing file and black box test codes for one function** |
| Updated Jira | **Added issues, tasks, and tracked progress** |
| Scrum Report | **Successfully summarized all activities, decisions, outcomes of milestone 3** |
|  |  |

**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 |
| Adding function specifications | **Done early and according to what has been discussed.** |
| Making black box test cases | **Done early and according to function specifications, as well as what was planned.** |
| Making of requirements traceability matrix | **Done on time and according to test cases specifications.** |
| Jira Updated + Scrum Report | **All members contributed to updating the Jira and providing feedback for the report.** |
|  |  |
|  |  |
|  |  |

**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 | **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.

In milestone 3, we have not made any white box test cases yet, but we have done black box test cases. With black box test cases, we were able to create the multiple test cases even before the implementation of our different functions because to create black box test cases, we do not need to know the logic and inner workings of the functions that we have made. We simply need to declare the functions, then assume the possible outcomes that can be tested from those functions. Once we create and conduct white boxes in this project, it will be much more complex because we have to consider all the features and logic of each code. However, the advantages of white box tests are that they are tailored to our code and we would be able to cover more possible outcomes in comparison to black box tests, which are relatively easy to create, but are not specific enough to consider the project complete or error-free.

1. How did the traceability matrix help ensure that all functional specifications were adequately tested? Reflect on its role in maintaining comprehensive test coverage.

The requirements traceability matrix greatly helped in analyzing which requirements are linked to which test cases. In this way, all possible scenarios can be tested and testing gaps are found more easily. The traceability matrix also helps in tracking the progress of each test case outcomes. Requirements may also change in the middle of development, so tracking the coverage and status of test cases for each function will help greatly in preventing accidental repetition of code and analyzing if any new test cases must be added. Furthermore, it can also help the group realize which test cases should be a priority so that the debugging process will not be delayed by any difficult bugs.

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?

Two function prototypes:

bool checkSpace(Truck truck, Shipment shipment);  
int assignPackage(Map map, Truck trucks[], Shipment shipment);

These two function prototypes are necessary for our project because: checkSpace() calculates if a truck can accept a given shipment based on the weight and size of the shipment as well as the space available in the truck. On the other hand, the assignPackage() function calculates which truck a shipment should be distributed to, which route to follow, and if a diversion should be made depending on the input values. Both of these functions are necessary because it meets some of the main requirements of the project, where the shipment must be the correct size and weight, which will then be distributed to the appropriate truck, and which route to follow on the map depending on the type of truck and the point of destination.