# **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: 2**

**Members Present**:

|  |  |
| --- | --- |
| 1. Luca Novello | 4.Eric Yakimoff |
| 2. Philip Ayomide Tijani | 5. Tyler Kay |
| 3. Karishma Singh Mahender | 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** |
| Eric | Additional functions, Test Code, Scrum report, Data structures | n/a |
| Luca | Additional functions, Scrum report, Data structures | n/a |
| Karishma | Analyze problem, Test-plan document | n/a |
| Phillip | requirements traceability matrix, Scrum report | n/a |
| Tyler | Github setup, Scrum report | Github setup |
|  |  |  |
|  |  |  |

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** | Github setup |
| **Reason for delay or block** | Personal schedule conflicts and VPN problems |
| **Impact on Project** | Unable to submit screenshot of Github |
| **Solution or work-around** | Will be reviewing in class to set up on school computers |
|  |  |

**Summary of Meeting:**

A summary of the main points discusses in the meeting and the outcomes of the discussions.

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| Dividing Work | Discussed the tasks that needed to be completed and divided work tasks | Updated Jira tasks. Scrum report tasks |
| Additional functions | Which functions would be useful | Additional function header added to repo |
| Test plan | Update test plan | Test plan updated |
| Update data.h | Add additional structures | Data.h updated |
| Test documents | How to go about creating blackbox tests | Test documents & requirements traceability matrix added |

**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 |
| Task distribution | Tasks were dispersed based on comfort with the skillsets needed as well as an emphasis on splitting the work equally. |
| Additional functions creation | Problem was reviewed to determine additional functions needed. |
| Testing approach | The black box test documents were initially discussed as a group and then further refined through individual contributions. |

**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? |
| Eric | Problem review, task distribution, additional functions | 10 min | Some |
| Luca | Problem review, task distribution, additional functions | 10 min | Some |
| Karishma | Problem review, task distribution, test-documents & traceability matrix | 10 min | Some |
| Philip | Problem review, task distribution, test-documents & traceability matrix | 10 min | Some |
| Tyler | Problem review, task distribution, test-documents & traceability matrix | 10 min | Some |

**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 |
| Eric | Review milestone deliverables, distribute tasks, scrum report, reflections |
| Luca | Review milestone deliverables, distribute tasks, scrum report, reflections |
| Karishma | Review milestone deliverables, distribute tasks, scrum report, reflections |
| Philip | Review milestone deliverables, distribute tasks, scrum report, reflections |
| Tyler | Review milestone deliverables, distribute tasks, scrum report, reflections |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Preliminary prioritization of tasks | Allows members to specialize and get comfortable with the work they prefer doing |
| Preliminary development of additional functions | Provided a base to work off when further structuring the additional functions |
| Preliminary development of Test Documents | Provided a base to work off when further structuring of the test documents and matrix. |

**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 meeting | Meeting was productive in establishing a starting point for the week’s tasks |
| Communication | Clear and honest discourse around the parts of the project we were all comfortable with ensured that work would be completed |
| Git | Allows us to share work easier and show our contributions |
| Jira | Allows us to communicate tasks and responsibilities easier |
|  |  |
|  |  |
|  |  |

**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. **What is the difference between blackbox tests cases and blackbox test code? Explain how we use assertion in Visual Studio to execute tests.**Blackbox Test Cases are designed scenarios and inputs used to validate the functionality of software without considering the internal workings. These cases outline what inputs should be given and what the expected outputs are. They focus on requirements and specifications instead of the code itself. Test cases document the conditions under which the software is to be tested, the steps to perform the test, and the expected results.

Blackbox Test Code, on the other hand, is the actual implementation of the test cases in a programming language using a testing framework. The code executes the test cases automatically, asserting whether the actual output match the expected output. The code includes the setup, execution, and validation of the test conditions.

In Visual Studio, assertions are used within test code to verify if a condition holds true. If the assertion fails, the test framework reports it, showing which tests failed and why. This helps ensure that the code meets the specifications provided.

1. **How can a traceability matrix help in the testing process?**A Traceability Matrix is a document that maps and traces user requirements with the test cases designed to verify those requirements. They are important in large projects. It ensures that all requirements defined for a system are tested in the test protocols.

* Ensuring that software has been written to implement each of the requirements
* Ensuring that tests have been developed to test all of the software written for each requirement.

The matrix lists the business requirements across the top and the tests down the left side. Overall, a traceability matrix enhances the organization, efficiency, and effectiveness of the testing process, ensuring comprehensive coverage and easier management of requirements and tests.

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?**

**Function Prototype 1:** int findTruckForShipment(*const struct Map map, const struct Truck trucks[], const int numTrucks, const struct Package package*);

**Reason and Importance:** This function is critical as it determines which truck should be assigned a given shipment based on factors like truck volume, proximity to the destination, and the shipment’s weight and size. It ensures that the logistics are optimized, minimizing the distance trucks need to deviate from their paths while considering their load capacities.

**Function Prototype 2:** bool isValidDestination(const struct Map map, const std::string& destination);

**Reason and Importance:** This function checks if a given destination is valid, meaning it is within the delivery area and not an inaccessible point like inside a building. This is important to avoid errors in shipment assignments, ensuring that packages are only sent to reachable and correct addresses. By validating destinations, this function helps maintain the integrity and reliability of the delivery process, preventing the system from attempting to deliver to invalid locations.