# **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**: Team4

**Members Present**:

|  |  |
| --- | --- |
| 1. Seulgi Lee | 4. Alejandro Mercado |
| 2. Khassan Suleimanov | 5. |
| 3. Philip Grahamm | 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** |
| Seulgi Lee | 1. An analysis of the problem 2. Tested storing a documentation or source code in the Git repository 3. Exchanged opinions on the answers to the reflection questions 4. Complete scrum report 5. Write the Reflection, and when all of team members finished, combine them into a group reflection 6. Write a test plan. 7. Made timeline and board about Milestone2 using Jira and assign team members. 8. Update Jira with the work performed and planned | None |
| Khassan Suleimanov | 1. An analysis of the problem 2. Tested storing a documentation or source code in the Git repository 3. Exchanged opinions on the answers to the reflection questions 4. Tested storing a documentation or source code in the Git repository. 5. Write a test strategy. 6. Stored a test strategy in the repository. 7. Update Jira with the work performed and planned. | Tested storing a documentation or source code in the Git repository (Mac wasn’t working so it delayed a day) |
| Philip Grahamm | 1. An analysis of the problem. 2. Exchanged opinions on the answers to the reflection questions. 3. Tested storing a documentation or source code in the Git repository. 4. Write a test plan. 5. Stored a test plan in the repository. 6. Update Jira with the work performed and planned. | None |
| Alejandro Mercado | 1. An analysis of the problem. 2. Tested storing a documentation or source code in the Git repository. 3. Exchanged opinions on the answers to the reflection questions. 4. A series of data structures created as header files. 5. A series of data structures created as header files. 6. Update Jira with the work performed and planned. | 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** | Tested storing a documentation or source code in the Git repository in MAC |
| **Reason for delay or block** | The task was delayed by a day due to issues with the Mac system. |
| **Impact on Project** | There was no significant impact on the project as storing files was completed successfully before the submission. |
| **Solution or work-around** | The issue was resolved by using a Windows desktop to complete the task. |

**Summary of Meeting:**

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

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| Function Specifications Added to New Header File | |  | | --- | | Discussed adding at least 4 function specifications to a new header file. This is to meet the project's requirements. | |  | | |  | | --- | | Understood the necessity of new function specifications and the roles of each function. Decided on individual responsibilities for efficient progress. | |  | |
| Blackbox Tests for Functions | |  | | --- | | Discussed creating blackbox test documents in an Excel file format for the newly written functions, with at least 4 sets of test data for each function. Prepared test cases for 6 functions. | |  | | |  | | --- | | Recognized the importance of test data and the need to prepare various test cases for each function. | |  | |
| C++ Testing Project Creation | |  | | --- | | Discussed adding a C++ testing project to the solution. This is a preparation step for writing blackbox test code for the functions. | |  | | |  | | --- | | Prepared to add the testing project for efficient test execution. | |  | |
| Blackbox Test Code Writing | |  | | --- | | Discussed starting to write blackbox test code for the functions. At least one test code is required for this milestone. | |  | | |  | | --- | | Delegated tasks for writing the first test code and reaffirmed the importance of writing test code. | |  | |
| Function Implementation | |  | | --- | | Discussed starting the implementation of functions. This is optional for this milestone. | |  | | |  | | --- | | Identified additional resources and references needed for function implementation and created an implementation plan. | |  | |
| Requirements Traceability Matrix | |  | | --- | | Discussed creating a requirements traceability matrix to map requirements to test cases. | |  | | |  | | --- | | Understood the necessity and method of creating the traceability matrix, and decided to clearly document the relationships between requirements and test cases. | |  | |
| Jira Project Update | |  | | --- | | Discussed updating the Jira project to show activities and progress. | |  | | |  | | --- | | Agreed to share progress transparently among team members via Jira and to perform regular updates. | |  | |
| Completed Scrum Report | |  | | --- | | Discussed writing a completed scrum report, including reflection questions. | |  | | Decided to use the scrum report to facilitate team reflection, gather feedback, and continuously improve project processes. |

**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 |
| Function Specifications Added to New Header File | Decided to add at least 4 function specifications to a new header file to meet project requirements. As the first task priority, we decided to create function specifications and blackbox test documents. |
| Blackbox Tests for Functions | Decided to create blackbox test documents with at least 4 sets of test data for each function to ensure comprehensive test coverage. As the first task priority, we decided to create function specifications and blackbox test documents. |
| C++ Testing Project Creation | Decided to add a C++ testing project to the solution to facilitate the writing of blackbox test code for the functions. As the second task priority, we decided to add the C++ testing project. |
| Blackbox Test Code Writing | Decided to write at least one blackbox test code and store it in the repository to reaffirm the importance of test code writing and to create the first test code. As the third task priority, we decided to write blackbox test code and create the requirements traceability matrix. |
| Function Implementation | Decided to start implementing functions. This is optional, and additional resources and references were identified to create an implementation plan. |
| Requirements Traceability Matrix | Decided to create a requirements traceability matrix to map requirements to test cases, ensuring clear documentation of the relationships between requirements and test cases. As the third task priority, we decided to write blackbox test code and create the requirements traceability matrix. |
| Jira Project Update | Decided to update the Jira project to share progress transparently among team members and to perform regular updates. |
| Completed Scrum Report | Decided to write a completed scrum report, including reflection questions, to facilitate team reflection, gather feedback, and continuously improve project processes. As the fourth task priority, we decided to write the reflection. |

**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? |
| Seulgi Lee, Khassan Suleimanov,  Philip Grahamm,  Alejandro Mercado | Understood how to add function specifications to a new header file | 30min | Yes |
| Understood the initial test case documents | 10min | Yes |
| Reviewed C++ testing project samples | 10min | Yes |
| Exchanged opinions on the answers to the reflection questions | 10min | Yes |
| Assigned roles and completed Jira issues and assignments | 20min | 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 |
| Seulgi Lee | Complete scrum report. |
| Write the Reflection, and when all of team members finished, combine them into a group reflection. |
| Made timeline and board about Milestone3 using Jira and assign team members. |
| Write a requirements traceability matrix added to the repository and shows the mapping between the requirements and test cases. |
| Update Jira with the work performed and planned. |
| Khassan Suleimanov | Add new set of function specifications to a new header file stored in the repository. |
| Write reflection Q3 (about function prototypes) |
| Write function-description document and stored in a repository. |
| Update Jira with the work performed and planned. |
| Philip Grahamm | Write a set of blackbox tests as test documents with test data for the functions you created. |
| Write reflection Q1 (about blackbox test) |
| Update Jira with the work performed and planned. |
| Alejandro Mercado | Create and add a C++ testing project to your solution. |
| Add a set of AT LEAST 4 function specifications added to a new header file. |
| Write blackbox test code (for the functions above) and store in repository (at least 1 is required for this milestone). |
| Update Jira with the work performed and planned. |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Function Specifications Added to New Header File | Understood the necessity of new function specifications and the roles of each function. Decided on individual responsibilities for efficient progress. |
| Blackbox Tests for Functions | Recognized the importance of test data and the need to prepare various test cases for each function. Prepared test cases for 6 functions. |
| C++ Testing Project Creation | Prepared to add the testing project for efficient test execution. |
| Blackbox Test Code Writing | Delegated tasks for writing the first test code and reaffirmed the importance of writing test code. |
| Function Implementation | Identified additional resources and references needed for function implementation and created an implementation plan. |
| Requirements Traceability Matrix | Understood the necessity and method of creating the traceability matrix and decided to clearly document the relationships between requirements and test cases. |
| Jira Project Update | Agreed to share progress transparently among team members via Jira and to perform regular updates. |
| Completed Scrum Report | Decided to use the scrum report to facilitate team reflection, gather feedback, and continuously improve project processes. |

**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 |
| Held scrum meeting after class | Held a Scrum meeting immediately after class. It was time-efficient, allowed everyone to meet in person, and we could ask the professor questions directly if needed. |
| Discussion on Reflection | We discussed the overall reflection and decided that each person responsible for a particular task would write their part. This approach ensures that the reflection is more accurate and effective. |
| Problem Analysis | By analyzing the problem together, we shared what we know and what we don't know, significantly improving our overall understanding of the project. |
| Task Sharing via Jira | The team lead shared tasks on Jira, increasing clarity about the tasks and helping team members understand their roles better. |
| Prioritizing and Handling Tasks | By prioritizing and handling tasks in order, we ensured that critical tasks were completed on time, enhancing the overall efficiency of the team. |

**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 |
| None | **None** |

**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 serve as the framework, focusing on validating the functionality of an application purely based on its inputs and expected outputs, without knowing the internal workings of the system. On the other hand, blackbox test code translates these cases into executable scripts or programs. Assertions are pivotal in this context, acting as checkpoints that confirm the correctness of each test's outcome. By embedding assertions in the test code, developers can automate the validation process, ensuring that the application behaves as intended across various scenarios, thereby enhancing the reliability and robustness of the software.

1. **How can a traceability matrix help in the testing process?**  
   The traceability matrix is really useful during the testing process because it shows how requirements and test cases are linked. For example, if I have a requirement called R004, the matrix shows that it is tested by test cases T001, T002, T003, and T004. This way, I can make sure that all requirements are being tested, and I can easily find missing or duplicated tests.

The tracking matrix is also very useful when requirements change, as I can quickly see which test cases need to be updated. For example, if R004 changes, I can see right away that I need to modify test cases T001, T002, T003, and T004. This clear connection makes it easier for the team to manage testing and helps maintain the quality of the project.

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?**

First function, the ‘hasEnoughSpace’ function checks if a truck can hold an additional shipment by ensuring the new weight and volume won't exceed the truck's capacity. This function helps avoid overloading the truck, which is important for safety and following rules. It ensures only shipments that fit are loaded, preventing problems like truck damage or fines. This check is important for using trucks efficiently and safely in logistics and fleet management. Second function, the ‘validateDest’ function checks if a shipment's destination is valid. It ensures the destination is within the delivery area and can be reached for delivery. This function makes sure shipments are only sent to places that can be reached, which is important for efficient deliveries. By checking the destination, it helps avoid failed delivery attempts, saving time and resources. It ensures deliveries are planned and executed correctly, improving customer satisfaction and efficiency. This step is key for reliable and effective delivery services, which are important for any logistics or transportation project.