# **Milestone 5 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**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1.Mihyeon Park | 4.Manraj Singh |
| 2.Ana Masoumi | 5.Veronika Edith Turpo Meneses |
| 3.Hamzeh Khaled Nayef Muhiar | 6.Yashleen Brar |

## Milestone 5 Tasks

In this milestone, you should write, implement, and execute integration tests. Integration tests test how multiple functions work together to complete a task. Depending on what is being tested, you might be able to write unit tests to do the testing and automatically compare the results. In other cases, you might need to manually check the output to check it. This will all be stated in the tests where it discusses how they should be run.

As you update the function-test matrix, you will need to add a very brief description for each integration test so the matrix will clearly show what the tests are testing. Acceptance tests will be tested against actual user requirements and will list all the tests for each requirement.

Acceptance tests are the final tests and are largely aimed at showing the customer that the correct output is produced for different inputs. This will largely require manual testing.

**Deliverables due 11 days after your lab day:**

* Integration tests document (for the new functions you added) stored in repository with at least 4 sets of distinct test cases (each case must have at least 4 distinct test data).
* Integration tests coded (store in repo), executed (results in Jira and in test documents) and debugged.
* Finish implementing/coding whitebox tests. Store in repo, executed, results in Jira (and on corresponding test documents, and debugged.
* One acceptance test case for each requirement added to the test cases excel sheet.
* All acceptance tests implemented and added to the testing C++ project.
* Updated requirements traceability matrix stored in the repository.
* Completed scrum report including reflection questions answered.

**Rubric:**

|  |  |  |
| --- | --- | --- |
| **Individual** | Group participation (includes GitHub commits and Jira usage) | 80% |
| Teamwork | 20% |
| **Group** | Integration test case document (well written, complete, good test data) | 10% |
| Integration test code (well designed and documented) | 10% |
| Finish coding all functions and main (well-designed, written, and documented) | 10% |
| Finish coding blackbox and whitebox cases (well-designed, written, and documented) | 5% |
| Acceptance tests (well-designed, documented, and implemented) | 15% |
| Requirements traceability matrix updated | 5% |
| Test execution (performed, results recorded, issues created) | 5% |
| Debugging (bugs fixed, documented, Jira updated) | 5% |
| Git usage (used properly with good structure) | 5% |
| 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** |
| Mihyeon Park | Contributed to scrum report, coding for main, made test cases for integration, made test cases for acceptance test cases, updated test cases and traceability matrix | N/A |
| Ana Masoumi | Contributed to reflection question 3 , filling the scrum report ,made integration test cases and helped with some bug issues in the program. | N/A |
| Hamzeh Khaled Nayef Muhiar | Failed to do tasks assigned. Filled Scrum Report | Integration Tests, Acceptance Tests, Test Cases and traceability matrix. |
| Manraj Singh | Contributed to scrum report, made test cases for integration test cases and acceptance test cases, updated test cases and traceability matrix | N/A |
| Veronika Edith Turpo Meneses | Filled Scrum Report. Failed to do tasks assigned | Integration Tests, Acceptance Tests, Test Cases and traceability matrix. |
| Yashleen Brar | Contributed to scrum report, made test cases for integration. |  |

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** | **Integration Test Cases** |
| **Reason for delay or block** | **No communication, delayed work** |
| **Impact on Project** | **Less functionality of test cases** |
| **Solution or work-around** |  |
|  |  |
| **Delayed or Blocked Task** | **Acceptance Test Cases** |
| **Reason for delay or block** | **No communication, delayed work** |
| **Impact on Project** | **Less functionality of test cases** |
| **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 |
| Fixing Bugs | There where some bugs in functions that needed to be fixed. | Bugs fixed |
| Planning on how to do integration testing | Discussing how to do the integration testing and creating test cases | Integration tests done |
| Adding integration test into the traceability matrix. | Added the sets of integration test into the traceability matrix. | Integration tests added to traceability matrix. |
| Adding the test cases to the test cases template | All test units were added to the test cases template. | Unit tests added to test cases template document. |
| Planning the acceptance tests | Discussed and planned the acceptance tests for each requirement | Acceptance tests planned |
| Reviewing and updating the scrum report | Reviewed the scrum report and added missing details and reflections | Scrum report updated |
| Code review and debugging | | | Conducted a code review session and identified bugs to be fixed | Bugs identified and fixe |

**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 |
| Prioritization of tasks | Equal amount of works assigned to each member of team. |
| Acceptance Testing | Made Acceptance Test Cases for each requirements |
| Integration Testing | On what functions to integrate for |
| Testing execution | Testing code has been executed and updated |
| Bug Fixing | Identified and fixed bugs in the code. |
| Documentation Update | Updated documentation, including test cases and traceability matrix |
| Meeting Participation | Ensured all members attended and contributed to the meeting. |

**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? |
| Everyone | Contributed to Jira | **1 hour** | **yes** |
| Mihyeon Park, Anahita, Manraj Singh, Yashleen Brar | Contributed to coding functions | **3 hour** | **yes** |
| Everyone | Attended in meetings | **30 min** | **yes** |
| Everyone | Fixed bugs(Debugging) | **4 hour** | **yes** |
| Mihyeon Park, Anahita, Manraj Singh, Yashleen Brar | Acceptance test Coding | **3 hour** | **Yes** |
| Mihyeon Park, Anahita, Manraj Singh, Yashleen Brar | Blackbox test coding | **1 hour** | **Yes** |
| Mihyeon Park, Anahita, Manraj Singh, Yashleen Brar | Test execution | **N/A** | **Yes** |
| Mihyeon Park, Anahita, Manraj Singh, Yashleen Brar | Integration Test case coding | **3 hour** | **Yes** |
| Mihyeon Park, Anahita, Manraj Singh, Yashleen Brar | Integration Test case Document | **1 hour** | **Yes** |
| Mihyeon Park, Anahita, Manraj Singh, Yashleen Brar | Coding Functions in main | **3 hour** | **Yes** |
| Everyone | Git usage | **2 hour** | **Yes** |
| Mihyeon Park, Anahita, Manraj Singh, Yashleen Brar | Scrum report and Reflection | **2 hour** | **Yes** |
| Mihyeon Park, Anahita, Manraj Singh, Yashleen Brar | Traceability matrix updated | **1 hour** | **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 | Final test report |
| ALL | Test Execution() |
| ALL | Debugging |
| ALL | Scrum Report and Reflection |
| ALL | Jira usage and GitHub Commits |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Bugs fixed | **Some bugs were fixed which helped with testing** |
| New coded added for functionality enhancement . | **New codes added for main functionality** |
| Integration Testing | **Helped with making sure two functions work together** |
| Acceptance Testing | **Made sure all the requirements passed the acceptance testing** |
| Documentation Updated | **Ensured all test cases and traceability matrix were up to date** |
| Reflections and Discussions | **Improved understanding and alignment on project goals** |
|  |  |

**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 |
| Task distribution | Each person got to know what task they should do |
| SCRUM | All contributed. |
| Git | Useful for version control and keeping track of changes |
| Meeting | All attended meeting. |
| Collaborative Debugging | Effective identification and fixing of bugs through team effort |
| Planning and Coordination | Clear planning and coordination led to efficient task execution |
|  |  |

**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 |
| Main | **Had problems coding execution for main with some bugs** |
| Integration Testing | **Faced challenges with integrating different modules. Plan for additional integration testing sessions and include more pair programming to identify issues early.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 manual and automated testing? Why are we automating the testing process and what benefits does automation offer?  
   Manual evaluation is an approach where subjects save test documents without using any automation instruments especially for exploratory, usability and ad hoc assessments. On the other hand automated examination applies scripts and gears for conducting evaluations; this makes it suitable for performance measurement, regression tracking as well as repetition checking. We automate the verification mechanism so as to enhance its effectiveness, accuracy then reach. Automating has a lot of benefits among them includes; running tests at faster rates, being able always to execute them repeatedly producing similar outcomes. Moreover, it speeds up overall development process by allowing testers to concentrate on more complicated issues thereby minimizing human faults while providing speedy feedback mechanisms.
2. Why it is necessary to write integration tests given that the code has already passed blackbox and whitebox tests?   
   Integration tests are crucial even when code has succeeded in passing both black box and white box tests due to their ability to check interactions and dependencies across various components or modules. White box tests examine how an application’s internal structures or operations work while black box tests assess whether it works according to its specification without requiring knowledge on what goes on inside it. However, this does not mean that these tests confirm that individual units will work correctly once put together. Integration testing ensures that the integrated components will behave as expected with regards to detecting problems like data flow issues, mismatching interfaces among others as well as integration errors. Such thorough validation is necessary for system functionality and reliability guarantee.
3. List and describe one of the integration tests you created. Provide a thorough explanation of how the integration operates, detailing the flow of parameters from one function to another. Use one of your integration tests to support your answer.

Check Destination And check building function, this function is designed to cheek for 2 things , validshipment function which validates if the shipment is valid by checking the weight and size limitations , then if the shipment is valid it checks if the destination of the shipment is a building and its deliverable or not. This integration test with 4 distinct test case makes sure that these 2 functions work together correctly and provide the correct output.

It also receives routes for the best route function as to which route it should take to deliver the shipment and which path yellow blue or green.