# **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**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1. Kaitlyn Cassiela Marino | 4. Grace Gabrielle |
| 2. Bilal Umar | 5. |
| 3. Nolan Grossi | 6. |

## 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 in the repository, ensuring it shows both passed (green) and failed (red) tests.
* 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** |
| **Kaitlyn** | **Half of white box test documents, hook file, MS scrum report, updated Jira project** | **None** |
| **Bilal** | **Finished code implementation, finished black box coding, hook file** | **None** |
| **Nolan** | **Started white box test implementation, updated traceability matrix, hook file** | **None** |
| **Grace** | **Half of white box test documents, updated Jira project** | **Unlinked test project and source code project, hook files (DONE)** |
|  |  |  |
|  |  |  |
|  |  |  |

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** | **Unlinked test project and source code project (DONE)** |
| **Reason for delay or block** | **Had issues with linking both projects due to complicated property settings** |
| **Impact on Project** | **Minor delay** |
| **Solution or work-around** | **We will discuss how to properly link both projects in order to test the program independently** |
|  |  |
| **Delayed or Blocked Task** | **Hook file not submitted (DONE)** |
| **Reason for delay or block** | **Due to not being able to link the testing project and the source code project, the hook file could not be completed** |
| **Impact on Project** | **Minor delay** |
| **Solution or work-around** | **We will discuss how to properly link the testing and source code project so that the hook file can be submitted as soon as possible** |

**Summary of Meeting:**

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

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| Finish implementing white box test cases | **We discussed which member will finish implementing the code for white box test cases from the previous milestone** | **We decided that Bilal will finish the white box test implementation** |
| Write integration test cases in excel file | **We discussed which members will write integration test cases into the excel file** | **We decided that half of the integration tests will be written by Kaitlyn, and the other half will be written by Nolan** |
| Implement the integration tests and run them | **We discussed which members will implement the integration tests into the testing project** | **We decided that Bilal will be implementing the integration tests, with help from Grace** |
| Write acceptance test cases for each requirement | **We discussed which members will write the acceptance test cases for each requirement on the excel file** | **We decided that Kaitlyn and Nolan will be writing the acceptance test cases for each requirement** |
| Update requirements traceability matrix | **We discussed which member will update the requirements traceability matrix based on the results of all tests** | **We decided that Nolan will update the traceability matrix** |
| Scrum report + Jira project update + GitHub repository | **We discussed which members will be writing the scrum report for milestone 4 and update the issues and progress in the Jira project** | **We decided that Kaitlyn will be writing the scrum report, and Grace will be updating the Jira project. Both with feedback from other members. All needed files will be submitted accordingly to the Git repository** |
|  |  |  |

**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 |
| Finishing white box tests implementation | Bilal will be finishing the implementations for the rest of the white box test cases |
| Integration test cases document | Kaitlyn and Nolan will divide the integration test cases in half and write them in an excel file |
| Implementation of integration test cases, executed | Bilal will be implementing the integration tests with help from Grace, executing afterwards to get results |
| Acceptance test cases document | Kaitlyn and Nolan will write the acceptance tests in the excel file |
| Updated requirements traceability matrix | Nolan will update the requirements traceability matrix based on the results of all conducted test cases |
| Scrum report + Jira project + GitHub repository | Kaitlyn will be doing the scrum report with reflections, while Grace will be updating the Jira project. Both tasks will include feedback from other members. All necessary files will be uploaded in the Git repository |
|  |  |

**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 | **Beginning of scrum report, planning for milestone 5** | **~30 mins** |  |
| Bilal | **Finishing white box test implementations** | **~30 mins** |  |
| Nolan | **Planning for milestone 5** | **~30 mins** |  |
| Grace | **Planning for milestone 5, beginning of Jira updates** | **~30 mins** |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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 | Scrum report, part of the testing report |
| Bilal | Part of the testing report |
| Nolan | Implementation and execution of acceptance tests |
| Grace | Updated requirements traceability matrix, Jira updates |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Finished white box test cases implementation/execution | **We finished implementing and testing all white box test cases for the implemented functions, with test results. We were able to move on to integration testing.** |
| Writing of integration tests to excel file | **We finished creating integration test cases for the implemented functions and put them in the excel file.** |
| Implementing/executing integration tests | **We finished implementing and testing all integration test cases for implemented functions, with test results. We were able to move on to acceptance testing.** |
| Writing of acceptance tests to excel file | **We finished writing acceptance tests for each requirement and put them in the excel file. We wer able to move on to implementing the acceptance tests and executing them.** |
| Scrum report / Jira updates / GitHub repository updates | **We were able to complete and finish all necessary requirements for milestone 5 scrum report, uploaded updates and test results on Jira, and updated files in the Github repository** |
|  |  |
|  |  |

**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 |
| Finished implementing/executing white box test cases | **We were able to create test cases that clearly matched what the code needed. We also implemented the test cases clearly in the testing project solution.** |
| Finished writing and implementing/executing integration test cases | **We were able to create test cases that clearly matched and called all the appropriate functions. We also implemented the test cases clearly in the testing project solution.** |
| Finished writing acceptance test cases | **We were able to write acceptance test cases for each requirement as shown in the traceability matrix.** |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Acceptance test coding, executing, and test results | **We were unable to implement the acceptance tests due to lack of availability, so we couldn’t execute, debug and gather test results. We will solve these issues in milestone 6 and make sure all tests are covered for the entire project.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Reflections**:

Answer the following questions using your own words. Make sure that each answer comprises a minimum of 100 words.

1. What challenges did you encounter when testing the interactions between different components during integration testing? Reflect on how identifying and resolving these issues improved the overall functionality of the system.  
     
   During integration testing, the challenges we encountered were the dependency issues. When one component of our function relies on another function and that function fails or misbehaves, we have to debug and fix both functions and make sure they run properly to finally conduct the tests. The other challenge was also trying to fit the parameters of the other function with the data in the current function. There were sometimes a slight mismatch and we had to do certain variable casts and odd object accessing to fit the parameters of another function, which was slightly difficult to understand at first, but it worked nonetheless.
2. How did focusing on end-user requirements during acceptance testing influence your approach to creating test cases? Reflect on how this perspective helps ensure the software meets its intended purpose.

For the making of the acceptance tests, focusing on end-user requirements allowed us to see the bigger picture – by identifying what features the end-user could possibly want from our program, we made sure that everything was concise and easy to understand. We made sure that they do not need to see nor understand the inner workings of the code to know what it does and how to format their input so that the program accepts it accordingly. We prioritized the core functionalities of the program, which is focused on making sure that the program correctly validates the shipment and truck specifications and that the shipment is delivered to the right address with or without any diversion. Additionally, we considered the most common data input that the end-user could give and made sure that our test cases aligned with those kinds of scenarios. This leads to a more realistic and user-based approach that makes sure that the program can be used by both technical and non-technical users.

1. 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.

One integration test that we created involved two functions: checkSpace and validShip. The checkSpace function calculates if a truck can accept a given shipment or not, while the validShip function calculates if a given shipment is within the bounds of weight and size as per the project instructions.

The integration test case tests that an invalid shipment weight is first checked using validShip before being checked for available in a truck using checkSpace. Since the validShip will detect an invalid value immediately, the checkSpace should not be called.

Parameters:

validShip:

SHIPMENT:

Weight = -10 kg

Size = 2.0

checkSpace:

TRUCK:

Weight: 1000 kg

Volume: 50

Expected Results:  
validShip = false, checkSpace = NOT EXECUTED

Results: TEST PASSED