# **MILESTONE 5** -- SFT221 SCRUM Report and Reflection

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.Noah Abebe | 4. |
| 2. Theo Oey | 5. |
| 3. Arman | 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 4 days after your lab day:**

* Integration tests document 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.
* Acceptance tests written and stored in repository.
* Updated requirements traceability matrix stored to 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) | 10% |
| Acceptance tests (well-designed, written and documented) | 5% |
| Requirements traceability matrix updated | 5% |
| Test execution (performed, results recorded, issues created) | 10% |
| Debugging (bugs fixed, documented, Jira updated) | 10% |
| Git usage (used properly with good structure) | 5% |
| Jira usage (creates issues, tracks progress) | 10% |
| 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:

| Member | Tasks Completed | Tasks Delayed/Blocked |
| --- | --- | --- |
| Noah | Implemented findValidTruckPaths function, Completed printRoute function, Wrote unit tests for isBoxSizeExceeded | Delayed: Writing integration tests for findValidTruckPaths due to unexpected complexity |
| Theo | Implemented isTruckOverloaded function, Refactored isBoxSizeExceeded function for better readability, Participated in code review | Delayed: Writing acceptance tests for getBestRoute due to time constraints |

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 |
| --- | --- | --- | --- |
| Writing integration tests for findValidTruckPaths | Unexpected complexity in mocking dependencies and setting up test environment | Delays the completion of testing phase for this function, potentially affects overall project timeline | Plan additional time for writing integration tests, seek assistance from team members if needed |

Summary of Meeting:

| Topic | Discussion Summary | Outcome |
| --- | --- | --- |
| Code Review | Reviewed implementation of findValidTruckPaths and isTruckOverloaded functions, Provided feedback for improvements | Agreed upon necessary changes, Assigned tasks for code refactoring |
| Project Timeline | Discussed potential impact of delayed tasks on project timeline, Identified areas for optimization | Adjusted project plan to accommodate delays, Scheduled additional testing phase to catch up on missed deadlines |

Summary of Decisions Made:

| Decision | Rationale |
| --- | --- |
| Scheduled additional testing phase to catch up on missed deadlines | Ensures thorough testing of all functionalities before project delivery, Minimizes the risk of undetected bugs in the final product |

Tasks Attempted During Meeting:

| Member | Task Attempted | Time Spent | Complete? |
| --- | --- | --- | --- |
| Noah | Implement feedback from code review, Begin writing integration tests | 30 mins | Partially |
| Noah | Refactor isTruckOverloaded function, Participate in code review | 45 mins | Yes |

SCRUM Tasks Selected for Next Week:

| Group Member | Task Description |
| --- | --- |
| Theo | Complete writing integration tests for findValidTruckPaths, Assist with code refactoring |
| Theo | Write acceptance tests for getBestRoute, Review code changes made by team members |

Major Outcomes of Meeting:

| Outcome | Impact on Project |
| --- | --- |
| Identified areas for improvement in code implementation | Ensures code quality and maintainability, |
| Adjusted project plan to accommodate delays | Maintains project timeline and delivery schedule, |

Things That Went Well in This Meeting:

| Topic/Work Item | Reason for Success |
| --- | --- |
| Effective communication and collaboration among team members | Facilitates problem-solving and decision-making process, Ensures alignment on project goals and objectives |
| Proactive identification of potential risks and issues | Enables timely mitigation strategies and contingency planning, Minimizes project disruptions |

Things That Did NOT go Well in This Meeting:

| Topic/Work Item | Reason for Problem and How to do Better |
| --- | --- |
| Difficulty in estimating testing effort accurately and | Lack of experience with complex testing scenarios, we were only a few people. |

Reflections:

**At this point, you are using the GIT hook to automate testing. Have you found that any of the tests failed and prevented you from pushing your code to the repository? If so, how did you handle the situation?**

Yes, some tests failed due to unexpected issues (incorrect test data or dependencies). I addressed the failures by investigating the root cause, making necessary corrections to the code or tests. Once all tests pass locally, I push the code to the repository.

**Explain why we are automating the testing process and what the advantages of this automation are.**

Automating the testing process helps to improve efficiency, reliability, and consistency in testing activities. The advantages of automation include faster test execution, broader test coverage, early detection of defects, and easier regression testing. It also frees up manual testers to focus on more complex testing scenarios and exploratory testing.

**Did you find the integration and acceptance tests more difficult to write than the black box and white box tests? If so, why were they harder to write? Did you write more white box and black box tests or more integration and acceptance tests?**

Integration and acceptance tests can be more challenging to write due to their dependencies on multiple components and external systems. I wrote more white box and black box tests initially to validate individual functionalities. But then, I understood the importance of integration and acceptance tests for validating end-to-end scenarios and ensuring overall system reliability.

**Explain why it is necessary to write integration and acceptance tests given that all of the code has already passed black box and white box tests.**

Integration and acceptance tests are necessary to validate the interactions and behavior of components. While black box and white box tests focus on individual units or components in isolation, integration tests verify the integration points and interactions between these units.