# SFT221 SCRUM Report and Reflections

This report should be completed in the class and submitted at the end of class. Late submissions cannot be accepted without prior approval of the instructor.

**GROUP**: \_\_\_\_\_\_\_\_3\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1. Rehatpreet Kaur | 4. Aayush Bhogal |
| 2. Yatin Bawa | 5. Muhammetyar |
| 3. Nehmat Ladhar | 6. Omar Carrasco |

## 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 distance 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) | 5% |
| Debugging (bugs fixed, documented, Jira updated) | 5% |
| Git usage (used properly with good structure) | 5% |
| Jira usage (creates issues, tracks progress) | 10% |
| Scrum report & reflections | 20% |
| Meets deadlines | 5% |

**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** |
| **Rehatpreet Kaur** | **Integration Test Case Documentation-** **Coding Blackbox and Whitebox Cases** | **Extension** |
| **Yatin Bawa** | **Finish Coding All Functions - Integration Test3** | **Extension** |
| **Nehmat Ladhar** | **Updated Traceability Matrix-** **Integration Test2** | **Extension** |
| **Aayush Bhogal** | **Reflection Questions- Scrum Report** | **Extension** |
| **Muhammetyar** | **Integration Test1-** **Acceptance Testing** | **Extension** |
| **Omar Carrasco** | **Integration Test4** | **Extension** |
|  |  |  |

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** | **Finish Coding All Functions** |
| **Reason for delay or block** | **Encountered complexities in unit testing which required additional time to resolve.** |
| **Impact on Project** | **Delay in finalizing the functions has postponed subsequent integration testing.** |
| **Solution or work-around** | **Prioritizing bug fixes in unit tests and allocating additional resources to expedite the coding process.** |
|  |  |
| **Delayed or Blocked Task** | **Integration Test** |
| **Reason for delay or block** | **Dependent on the completion of all functions, impacted by the delay in unit testing.** |
| **Impact on Project** | **Inability to commence integration testing as planned.** |
| **Solution or work-around** | **Coordination with the function coding team to identify partial integration testing opportunities.** |

**Summary of Meeting:**

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

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| Integration Testing Progress | **Reviewed the current status of integration tests, including the challenges faced due to delays in unit testing. Discussed the need for additional resources and potential strategies to accelerate testing without compromising quality.** | **Agreed to allocate more time for integration testing and to temporarily reassign team members to focus on this task.** |
| Acceptance Test Planning | **Focused on the development of acceptance tests, ensuring they align with customer requirements. Emphasized the importance of manual testing for these tests to validate the end-to-end functionality.** | **Decided to draft a detailed acceptance test plan and schedule a review session with the customer for feedback.** |
| Update on Traceability Matrix | **updated traceability matrix, highlighting the progress and identifying areas needing more attention.** | **Approved the updates and emphasized the need for continuous maintenance of the traceability matrix to reflect the project's current state accurately.** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Increase Focus on Integration Testing | Due to delays in unit testing, it is critical to intensify efforts on integration testing to ensure the project remains on schedule. |
| Regular Updates to Traceability Matrix | Regular updates to the traceability matrix are essential for keeping track of the project's progress and ensuring all requirements are being met |
| Reschedule Key Deadlines | Given the delays in certain tasks, particularly unit testing, rescheduling key milestones is necessary to provide more time and reduce pressure on the team. |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Rehatpreet Kaur | **Finalizing Integration Test Documentation** | **1 hour** | **partially** |
| Yatin Bawa | **Debugging Unit Tests** | **1 hour** | **Partially** |
| Nehmat Ladhar | **Updating Traceability Matrix** | **1 hour** | **Partially** |
| Aayush Bhogal | **Drafting Reflection Answers** | **1.5 hours** | **Partially** |
| Muhammetyar | **Preparing Data Sets for Acceptance Testing** | **2 hours** | **Partially** |
| Omar Carrasco | **Reviewing Integration Test4** | **1 hour** | **partially** |
|  |  |  |  |

**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 |
| Rehatpreet Kaur | Review and refine all test documentation |
| Yatin Bawa | Complete the debugging of unit tests |
| Nehmat Ladhar | Continuous update of the Traceability Matrix |
| Aayush Bhogal | Compile and finalize the SCRUM report |
| Muhammetyar | Complete and execute all acceptance tests |
| Omar Carrasco | Verify and finalize Integration Test4 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Accelerated Integration Testing | **Enhanced focus on integration testing will help to catch up with the delayed schedule and ensure project milestones are met.** |
| Commitment to Regular Updates on Traceability Matrix: | **Ensures that the project's progress is transparent and well-documented, facilitating better management and oversight.** |
| Rescheduling of Key Deadlines: | **Provides a more realistic timeline, reducing pressure on the team and allowing for more thorough and quality work.** |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Reflections**:

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

1. 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?

During the implementation of GIT hook for automated testing, we encountered instances where tests failed, preventing us from pushing code to the repository. Initially, this was frustrating as it slowed down our progress. However, we realized that these failures were beneficial. They highlighted flaws in our code at an early stage, allowing us to address them promptly. We handled these situations by reviewing the failed tests to understand the root cause of the failures. In most cases, it involved minor fixes or adjustments in the code. This process not only improved the quality of our code but also enhanced our debugging skills and understanding of the project’s intricacies.

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

Automating the testing process has been a game-changer for our project. The primary advantage is the significant time-saving it offers. Automation allows us to run tests quickly and frequently, which means we can detect and fix errors much faster than manual testing. Additionally, it ensures consistency in testing since automated tests run the same way every time, eliminating the possibility of human error. This consistency is crucial for maintaining the reliability of the tests over time. Moreover, automated testing allows us to focus more on designing and developing new features rather than spending time on repetitive testing tasks.

1. 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?  
     
   Writing integration and acceptance tests was indeed more challenging compared to black box and white box tests. The complexity lies in the fact that integration tests involve ensuring that different components of the application work together seamlessly, which often requires a thorough understanding of the entire system. Acceptance tests, on the other hand, require us to think from the end-user's perspective, which can be challenging to simulate accurately. In contrast, black box and white box tests are more focused on individual components, making them less complex. We found ourselves writing more white box and black box tests due to their straightforward nature, but the integration and acceptance tests were critical for ensuring the overall effectiveness and user satisfaction of the application.
2. 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.

Despite the code having passed black box and white box tests, writing integration and acceptance tests remains necessary. Black box and white box tests focus on individual units or components of the application, ensuring they function correctly in isolation. However, integration tests are crucial because they verify that these individual components work together as intended in a unified system. This is where complex issues often arise, as interactions between components can produce unforeseen errors. Acceptance tests are equally important as they evaluate the system from the end-user's perspective, ensuring the application meets their requirements and expectations. These tests are crucial for delivering a robust and user-centric product.