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

**Members Present**:

|  |  |
| --- | --- |
| 1. Song Nhat Nguyen | 4. Prince Ghumaan |
| 2. Nguyen Dang Khoa Huynh | 5. |
| 3. Mohamed Mohamed | 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:**

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** |
| **Song Nhat Nguyen** | **Integration Testing, Test Execution, update White Box Testing excel file, Traceability Matrix** | **N/A** |
| **Nguyen Dang Khoa Huynh** | **Acceptance Testing, Support Integration Testing, Test Execution, Acceptance Testing excel file, SCRUM Report, Assignment task on Jira and update files on GitHub Repo** | **N/A** |
| **Mohamed Mohamed** | **Reflection Question 1 and 2** | **N/A** |
| **Prince Ghumaan** | **Reflection Question 3** | **N/A** |
|  |  |  |
|  |  |  |
|  |  |  |

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** | **N/A** |
| **Reason for delay or block** | **N/A** |
| **Impact on Project** | **N/A** |
| **Solution or work-around** | **N/A** |
|  |  |
| **Delayed or Blocked Task** | **N/A** |
| **Reason for delay or block** | **N/A** |
| **Impact on Project** | **N/A** |
| **Solution or work-around** | **N/A** |

**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 | **Integration testing has been documented and analyzed.** | **Completed** |
| Acceptance Testing | **Acceptance testing has been documented and analyzed.** | **Completed** |
| Reflection | **Discussing Reflection question** | **Completed** |
| Git | **Git update files** |  |
| Jira | **Task Scheduled setup in Jira** | **Completed** |
|  |  |  |
|  |  |  |

**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 |
| Acceptance Testing | To validate that the entire system meets the requirements and is ready for deployment. Testing real-world scenarios such as package addition, truck selection, and shortest path calculations. And ensuring the system behaves as expected under normal and edge-case conditions. |
| Integration Testing | A determination has been reached regarding the "All components" for integration testing. Selecting Bottom-up Testing as a Methodology. |
| Testing Execution | Testing code has been executed and recorded |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Nguyen Dang Khoa Huynh | **SCRUM Report** | **45 min** | **Yes** |
| Nguyen Dang Khoa Huynh | **Jira and GitHub repo updated and assigned** | **30 min** | **Yes** |
| Nguyen Dang Khoa Huynh, Song Nhat Nguyen | **Test execution and discussion about debugging** | **2 hours** | **Yes** |
| ALL | **Discuss about Reflection Question** | **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 |
| Nguyen Dang Khoa Huynh | Managing Jira and GitHub |
| ALL | Reflection |
| ALL | Final Test Report |
| ALL | Test Execution |
| All | Debugging |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Comprehensive enhancement of program functionality | **Through extensive black box and white box testing, together with acceptance and integration testing, we have attained a significant degree of proficiency in our newly developed functions within the existing program's data structure.** |
| Verification of Acceptance Testing | **Based on White Box Testing, Executing codes for Acceptance Testing and listed out Debugs, if any** |
| Verification of Integration Testing | **Executing Testing codes (Alpha Testing) and listed out Debugs** |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Integration testing | **Made sure all test passed** |
| Acceptance Testing | **Made sure alpha testing done and documented** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| N/A | **N/A** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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?  
     
   We have implemented measures to improve our development workflow by incorporating informative prompt messages. These prompts assist developers in navigating the commit process within the Bash terminal, guaranteeing meticulous evaluation prior to finishing modifications.   
   We presently employ GitHub Desktop to track test results, detecting any problems indicated in the testing files. We swiftly resolve these issues, ensuring a strong and dependable code foundation.  
   Upon addressing issues and executing corrections, we advance with the commit. We recognize the importance of recording progress by use commits to mark substantial developments in files or the whole project, hence fostering transparency and collaboration.   
     
   Although automated hooks provide efficiency, we remain cognizant of any constraints regarding version control and progress monitoring. Achieving equilibrium between automation and developer adaptability continues to be a primary objective.   
   Fundamentally, our timely communications and deliberate utilization of GitHub Desktop highlight our dedication to code quality, meticulous version control, and extensive progress monitoring in our development workflow.

1. Explain why we are automating the testing process and what the advantages of this automation are.  
     
   Automating the testing process offers numerous benefits to software development:   
   Efficiency: Automated testing rapidly identify flaws and regressions, expediting the feedback loop.   
   **Consistency:** Accurate execution of test scripts mitigates human errors, guaranteeing dependable outcomes.   
   **Coverage:** Automated tests thoroughly encompass scenarios, including edge cases and setups.   
   **Frequency:** Incorporation into agile and CI/CD pipelines facilitates regular testing, enabling early detection of faults.   
   **Scalability:** Effectively managing expanding test cases aligns with developing codebases.   
   **Collaboration:** Objective assessments promote efficient communication among parties.   
   **Cost Savings:** Although there is an initial setup, the long-term decrease in human labor results in financial advantages.

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?  
     
     
   Integration testing presented a more significant barrier than white box testing due to the necessity for complex interconnections between units and the development of new testing programs. The Alpha and Beta rounds of acceptance testing introduced additional complexity, necessitating meticulous user selection and feedback management.   
   Identifying appropriate Beta users within our team was complex due to the need for real-world usage simulation and valuable feedback.   
   Confronting challenges, we heightened our emphasis on integration testing protocols. Discontent with the preliminary results of black and white box testing, we collectively dedicated more work to enhance our testing methodologies, especially in MS05.   
   This dedication sought to thoroughly analyze our software's performance, guarantee compatibility, and preemptively resolve possible difficulties, in accordance with our commitment to a resilient end result. Our testing endeavor showcased our commitment to refining tactics and providing superior software.

1. 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.

Black box and white box testing are essential for evaluating individual software components; however, integration and acceptance testing provide distinct benefits that enhance these phases.

Integration tests verify that several components, which have previously passed individual tests, function cohesively without complications. They reveal concealed anomalies, discrepancies, or interdependencies that arise during interactions between components. This ensures uninterrupted functionality in practical situations.

Acceptance tests, such as Alpha and Beta testing, engage actual users, uncovering insights that extend beyond mere technical accuracy. They detect usability problems, user interface errors, or performance constraints that previous testing may overlook. Incorporating actual users guarantees that the program fulfills their requirements, hence improving quality and market preparedness.

In conclusion, integration and acceptance tests encompass more than just black box and white box validations. Integration tests verify seamless component interaction, and acceptance tests validate user happiness and usability, thereby reducing errors and enhancing program reliability.