# 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**: \_\_\_\_\_\_\_\_5\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1. Audrey Duzon | 4. Ka Ying Chan |
| 2. Tae Yong Eom | 5.Julia Alekseev |
| 3.Azad Zeynalov | 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 at end of Lab:**

* Completed SCRUM report and reflections

**Deliverables Due at 23:59 12 Days after Lab:**

* integration tests written and stored in repository,
* integration tests written (store in repo), executed (results in Jira and in test documents) and debugged.
* acceptance tests written and stored in repository.
* Updated function-integration-requirements-test matrix stored to the repository.

**Rubric**

|  |  |  |
| --- | --- | --- |
| Individual | Group Participation | 75% |
| Teamwork | 10% |
| SCRUM Report and reflections | 15% |
| Group | integration tests (well-designed, written and documented) | 20% |
| acceptance tests (well-designed, written and documented) | 20% |
| Test Execution (performed, results recorded, issues created) | 15% |
| Debugging (Bugs fixed, documented, Jira updated) | 5% |
| Function-test matrix updated | 5% |
| Git Usage (used properly with good structure) | 5% |
| Jira Usage (creates issues, tracks progress) | 5% |
| Meets Deadlines | 5% |
| SCRUM Report and Reflections | 20% |

**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** |
| **Julia** | **Function Implementation, Debugging, Reflection** | **Completed** |
| **Taeyong** |  |  |
| **Audrey** |  |  |
| **Nicole** |  |  |
| **Azad** |  |  |
|  |  |  |
|  |  |  |

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** |  |
|  |  |
| **Delayed or Blocked Task** |  |
| **Reason for delay or block** |  |
| **Impact on Project** |  |
| **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 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

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

**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? |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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**:

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?**

While being unable to push into the repositories are quite cumbersome, they are quite useful in maintaining an efficient completion of the projects.

Any tests failures based on our additions ensure that we must take accountability and proof-read our changes. They encourage teamwork by asking for assistance with debugging & getting feedback from our own solutions to the current issues.

Two options were used:

* To remove our changes and save them locally for review with another team member.
* To debug and find solutions to why our additions are causing an issue. Only when resolved are we allowed to push.

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

Using GIT hook test automation ensures that any changes to the code must meet the standard set by the team. New additions to the **code must be diligently checked** for any bugs before being posted into the repository. While version control allows the team the ability to “roll back” via overwriting their local copy, it is much preferred **to halt any incorrect changes** from being uploaded.

In doing so, **every member can focus on their task** without worrying that the main branch is compromised.

Automating the test is essential not just during development phase but also in maintenance phase. It ensures that **our application is continuously improved, maintained and meets the quality standards** set by our company and our client. It also improves and maintains good user experience since no errors are introduced into the working product. It stays in the development environment.

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 was harder to write but due to being able to reference previous unit tests and test cases it is made manageable.

Integration tests covers more aspects of the application compared to Unit testing.

The difficulty lies in designing the tests itself. Planning and communication between the testers and the developers is essential in creating a well-made integration test.

Our team ensured to make a balanced amount of white box and black box tests. The tester team is further divided between white vs black box testing with each having the ability to view the test cases from one another to reduce redundancies.

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.**

It is essential in testing the functionality of each method created as a whole. Independently, each method may pass the test cases, but it may produce a different result.

Often, team members are tasked with a specific feature of the project. Integration testing ensures that each feature is functioning well with the current working application.

This additional test also ensures that the quality of the product is maintained and upgraded as needed.

Acceptance testing on the other hand involves the addition of user experience. This is essential in gathering feedback and finding flaws in the current product. It also allows clients to be able to see a working product while in development. It is important in maintaining a good working relationship with your current client and potential new clients. From this data, we can compare our product with other similar products which aids in further upgrades and debugging. Acceptance testing is further broken down into two aspects: alpha and beta testing.