# **MILESTONE 6** -- 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**: \_\_\_\_\_\_\_\_\_\_\_\_\_E\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1. Kemal Batu Turgut | 4. Tu Yin Hnit Aung |
| 2. Kusum Acharya | 5. Roy Bryan D. Franck |
| 3. Krish Sanjaybhai Patel | 6. Shovana Shrestha |

## Milestone 6 Tasks

This is the final milestone where you will run the acceptance tests and fix any remaining bugs found. In addition, you will produce a testing report which lists all the tests conducted, the results and whether the bugs were fixed, and the final test passed. You will also review the test matrix to ensure every test has been performed and passed. You can change the color of the test in the matrix to show it was run and passed. At the end, all tests in the matrix should have been passed.

The final test report can be tabular like this:

|  |  |  |  |
| --- | --- | --- | --- |
| Function/acceptance/requirement | Test Run | Bugs Fixed | Passed |
| Distance | TF001 | Did not handle negative coordinates | 🗹 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Deliverables due 4 days after your lab day:**

* Final testing report listing tests conducted, bugs fixed, and the final tests passed.
* Execute acceptance tests (results in Jira), and debug.
* Updated requirements traceability matrix stored in the repository.
* Completed scrum report including reflection questions answered.

**Rubric:**

|  |  |  |
| --- | --- | --- |
| **Individual** | Group participation (includes GitHub commits and Jira usage) | 80% |
| Teamwork | 20% |
| **Group** | Complete solution code running and executing successfully | 20% |
| Test execution (performed, results recorded, issues created) | 10% |
| Updated requirements traceability matrix | 5% |
| Final test report | 30% |
| 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 | 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** |
| **Kemal Batu Turgut** | Test documents |  |
| **Krish Sanjaybhai Patel** | Scrum Report |  |
| **Roy Bryan D. Franck** | Coding for acceptance testing and final check that complete solution is working properly before submission. |  |
| **Tu Yin Hnit Aung** | Test documents |  |
| **Kusum Acharya** | Updated requirements traceability matrix |  |
| **Shovana Shrestha** | Scrum Report |  |
|  |  |  |

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 |
| Task Allocation | We equally divided the tasks so that each member can finish their work on time. | Clear communication between team members. |
| Deadlines | The team discussed the upcoming deadline to ensure timely delivery without compromising quality work, deciding collectively to complete all tasks by Thursday for mutual review and potential fixes, ensuring all documents are submission-ready by Friday. | Timely delivery with quality work. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Dividing the tasks | We divided each task among the members, allowing for effective resource use and ensuring that each team member has clear responsibilities, hence increasing productivity and accountability. |
| Changing the deadline | As a team, we decided that all the tasks should be completed by Thursday so our team members can review each other’s work and suggest if some documents need to be fixed. Also, it will ensure that all documents are ready for submission by Friday. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Kemal Batu Turgut | Test documents | **2 hours** | **YES** |
| Krish Sanjaybhai Patel | Scrum Report | **1 hour** | **YES** |
| Roy Bryan D. Franck | Coding for acceptance testing and final check that complete solution is working properly before submission. | **2 hours** | **YES** |
| Tu Yin Hnit Aung | Test documents | **2 hours** | **YES** |
| Kusum Acharya | Updated requirements traceability matrix | **1 hour** | **YES** |
| Shovana Shrestha | Scrum Report | **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 |
| Kemal Batu Turgut | Test documents |
| Krish Sanjaybhai Patel | Scrum Report |
| Roy Bryan D. Franck | Coding for acceptance testing and final check that complete solution is working properly before submission. |
| Tu Yin Hnit Aung | Test documents |
| Kusum Acharya | Updated requirements traceability matrix |
| Shovana Shrestha | Scrum Report |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Decision Making | The decision-making was more transparent, and the tasks that were most crucial to achieving this week's objective were given top priority. This enhanced flexibility and adaptability, with a primary focus on quality assurance. |
| Task Division | We successfully divided this week's tasks among all the members. Prior to concluding the meeting, all members were given the opportunity to voice their opinions and choose the task they wanted to attempt. |
| Deadlines | We decided to finish all the tasks by Thursday ensuring timely review and submission of the 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 |
| Organized schedule | The meeting was held on time with all the members present, allowing for timely discussions and decisions. |
| Division of tasks among members | The division of tasks among team members guaranteed that responsibilities were clear and manageable, leading to productive growth and teamwork. |
| Clear communication & Collaboration | Open and transparent communication between team members created effective collaboration and idea sharing. |
| Problem-solving | Team members identified issues and helped each other find a solution. |
|  |  |
|  |  |
|  |  |

**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. Although we wrote a report on the testing that shows which tests were run and passed or failed, we also updated the function test matrix. What are the advantages of updating the function test matrix in addition to writing the test report?  
     
   **ANS:** Updating the function test matrix alongside the test report offers a range of advantages that helps a comprehensive and systematic approach to software testing and quality assurance. The matrix serves as a clear and visual record of the functions and components that have undergone testing, promoting thorough coverage across the software. This traceability fosters accountability by linking specific test cases to corresponding code segments. The process of updating the matrix encourages thoughtful test planning, enabling efficient allocation of resources and prioritization of critical tests. Furthermore, it aids in risk management by highlighting areas with limited test coverage, thereby identifying potential vulnerabilities. The matrix's role in assessing testing comprehensiveness provides valuable insights into software quality, which facilitates in decision-making regarding its release readiness. In the context of evolving software, an up-to-date matrix streamlines regression testing by pinpointing necessary test reruns following code changes. Its presence as a shared reference point makes collaboration and communication among development, testing, and management teams. Regular updates also support continuous improvement efforts by identifying trends and areas for testing process enhancement.
2. Teamwork on a project like this is vital to success. How well did your teamwork? If it worked well, what contributed to its success? If it did not work well, what contributed to the problems?

**ANS:** We employed both GitHub and Jira for streamlined collaborative project management. GitHub provided robust version control, enabling our team to efficiently manage code changes, collaborate seamlessly, and track progress. Leveraging features such as pull requests, branches, and issue tracking, we effectively assigned tasks, resolved conflicts, and maintained a comprehensive history of project developments. Additionally, Jira was a versatile project management tool, allowing us to plan, track, and prioritize tasks with a clear overview of project milestones. The combined utilization of GitHub and Jira empowered our team to work cohesively, harmonize efforts, and maintain an organized and successful project.

1. In every milestone you were asked what worked and did not work along the way. Were you able to incorporate what you learned to improve your team’s performance on the next milestone? Did your team learn from its mistakes and improve? If so, why? If not, why?

**ANS:** We've embraced each milestone as an opportunity to learn and grow. By dissecting what worked and what didn't, we've refined our approach, leveraging these insights to enhance subsequent milestones. As a united team, we've harnessed our collective experience, applying lessons learned to foster continual improvement. Through transparent communication and shared accountability, we've not only addressed challenges but also elevated our performance. Our dedication to learning from past experiences has driven us to adapt, resulting in smoother executions and more successful outcomes.

1. Did you end up testing the code to the point where you were convinced it worked correctly? Were there any tests that had not passed at the end? If so, what was the impact of this on the project?

**ANS:** The acceptance testing phase revealed that not all tests had passed, indicating that the code did not meet the expected level of functionality and quality. This situation had a notable impact on the project, highlighting potential issues and shortcomings that needed to be addressed. The tests that did not pass exposed areas where the code failed to align with the project's requirements and specifications. This posed challenges in terms of meeting user expectations and delivering a reliable product.

The occurrence of failed acceptance tests prompted a reflection on various aspects of the project. It raised questions about the thoroughness of the initial testing strategy, the accuracy of the requirements gathering process, and the effectiveness of the development approach. The failed tests prompted a reevaluation of the codebase to identify root causes and areas requiring further development or bug fixes. Therefore, team members had to invest additional time and effort to diagnose and fix these issues, which in turn impacted project timelines, resource allocation, and overall project delivery.