# 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.Asem M | 4.Kiki |
| 2.Thomas Lewis | 5. |
| 3.Hla Myint Myat | 6. |

## 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 colour 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 at end of Lab:**

* SCRUM Report and reflections

**Deliverables Due at 23:59 4 Days after Lab:**

* Execute acceptance tests(results in Jira), and debug.
* Updated function-test matrix stored to the repository.
* Final Testing report listing tests conducted, bugs fixed and the final test passed.

**Rubric**

|  |  |  |
| --- | --- | --- |
| Individual | Group Participation | 75% |
|  | Teamwork | 10% |
|  | SCRUM Report & reflections | 15% |
| Group | Updated test matrix | 20% |
|  | Final test report | 20% |
|  | Test Execution (performed, results recorded, issues created) | 10% |
|  | Debugging (Bugs fixed, documented, Jira updated) | 5% |
|  | Git Usage (used properly with good structure) | 5% |
|  | Jira Usage (creates issues, tracks progress) | 5% |
|  | Meets Deadlines | 5% |
|  | SCRUM Report & reflections | 30% |

**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** |
| **Thomas Lewis** | **Acceptance testing, test matrix, scrum , reflection, final report.** | **N/A** |
| **Asem M** | **test matrix** | **N/A** |
| **Hla Myint Myat** | **scrum report** | **N/A** |
| **Kiki** | **final report** | **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 |
| Function implementation | **function implementation for complete project** | **Completed** |
| Acceptance testing | **Reported during last milestone** | **Executed** |
| SCRUM report | **SCRUM Done** | **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 |
| Breaking down the task | Equal amount of works assigned to each member of team. |
| Acceptance Testing | Choosing Alpha and beta testing as process. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Asem M | **Jira assigned** | **30 mins** | **Yes** |
| Thomas Lewis | **GitHub updated** | **30 mins** | **Yes** |
| Hla Myint Myat | **Scrum** | **1 hr** | **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 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Acceptance testing | **Executing program individually (Alpha and beta) and listed out Debugs for matrix and ticketed to members** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Git | **Useful for version control** |
| SCRUM | All contributed |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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**:

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?  
     
     
   There are several benefits to updating the function test matrix in addition to the test report, which support an extensive and logical approach to software testing and quality assurance. The matrix fosters comprehensive coverage throughout the software by acting as an easily readable and visible record of the features or parts that have been tested. Because traceability connects particular test cases to matching code segments, it promotes accountability. Updating the matrix promotes intentional test planning, which helps effective use of resources and a listing of important tests in order of their value. Moreover, it helps risk control by drawing attention to sections that receive little test coverage, which helps to uncover potential weaknesses. The matrix's function in evaluating the depth of testing offers insightful information about the quality of the software and facilitates decision-making concerning allow accessibility. An updated matrix facilitates regression testing in the context of changing software by identifying required test reruns after code changes. The development, testing, and management teams can work together and communicate more easily when it is there as a common reference point. By highlighting patterns and potential areas for testing process improvement, regular updates also aid in continuous improvement initiatives. The matrix provides benefits for checking and satisfaction in regulated business sectors by demonstrating thorough testing procedures. Furthermore, it directs test automation by determining which functions are appropriate for automation, improving efficiency and reliability.
2. Teamwork on a project like this is vital to success. How well did your team work? If it worked well, what contributed to its success? If it did not work well, what contributed to the problems?  
     
     
   GitHub and Jira were both used by us for efficient, team-based project management. Our team was able to track progress, collaborate easily, and manage code changes with GitHub's strong version control. By utilizing tools like issue tracking, branches, and pull requests, we were able to efficiently assign work, settle disputes, and keep an accurate record of project developments. Jira was also a flexible project management tool that gave us a clear picture of project milestones and helped us plan, monitor, and prioritize tasks. By using Jira and GitHub together, our team was able to coordinate efforts, collaborate effectively, and keep the project well-organized.

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 improving your team’s performance on the next milestone? Did your team learn from its mistakes and improve? If so, why? If not, why?  
     
   Every milestone has been welcomed as a chance for us to improve and learn. We've improved our strategy by analyzing what worked and what didn't, and we've used these learnings to improve next milestones. Working together, we have taken advantage of our combined experience and applied the knowledge gained to promote ongoing progress. By means of open communication and mutual responsibility, we have not only tackled obstacles but also improved our output. We have adjusted because we are committed to learning from the past, which has led to more successful outcomes and more flawless executions.

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?

The results of the acceptance testing stage showed that some tests had failed, which suggested that the functionality and quality of the code were below expectations. This circumstance had a significant effect on the project, drawing attention to possible problems and weaknesses that required fixing. The tests that were unsuccessful revealed instances in which the code did not comply with the project's specifications and requirements. This created difficulties in delivering a dependable product and satisfying user expectations. Reflection on a number of project aspects was prompted by the failure of acceptance tests. It caused concerns about the effectiveness of the development strategy, the precision of the requirements gathering procedure, and the depth of the initial testing strategy. The codebase was reviewed in order to find the core causes of the failed tests and areas that needed bug fixes or additional development. Finding and resolution of these problems required more time and work from the project team, which had an effect on project schedules, resource allocation, and overall project delivery.