# 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**: C

**Members Present**:

|  |  |  |  |
| --- | --- | --- | --- |
| 1.Manav Zadafiya | 4.Fenil Soni | 1. | 4. |
| 2.Sunny Vavadiya | 5. | 2. | 5. |
| 3.Ashraf Bharot | 6. | 3. | 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** |
| **ALL** | **SCRUM Report and Reflection, Acceptance testing** | **4 days** |
| **Sunny** | **Review all deliverables for final submission, Update test matrix** | **4 days** |
| **Ashraf, Fenil** | **Final Report** | **4 days** |
| **Manav** | **Git and Github, managing files and deliverables** | **4 days** |
|  |  |  |
|  |  |  |

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** | **Milestone 6** |
| **Reason for delay or block** | **As we are close to final week, everyone in group has final projects, assignments, and exam so we used 4 days extension provided by our professor** |
| **Impact on Project** | **we are late by 4 days** |
| **Solution or work-around** | **Manage and prioritize tasks** |
|  |  |
| **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 |
| SCRUM Report | **Everyone contributed** | **Completed and ready to submit.** |
| Acceptance testing | **Documented and executed Finished** | **Executed** |
| Final Report | **How we will complete final report and what should be documented** | **Preparing Final report** |
| Github | **Organizing Github repository for final submission** | **Repository organized and ready for final submission** |
| Jira | **Updating all task and events** | **JIRA Updated** |
|  |  |  |
|  |  |  |

**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 |
| Prioritization of tasks | Equal amount of work 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? |
| ALL | **Scrum report** | **40min** | **Yes** |
| ALL | **Jira and Github Project page updated and assigned** | **30min** | **Yes** |
| ALL | **Acceptance testing** | **1 hrs** | **Yes** |
| Fenil, Ashraf | **Final Report** | **30 min** | **No** |
| Manav | **Github and Jira Update** | **10 min** | **Yes** |
| Manav | **Testing Review and documentation** | **30 min** | **Yes** |
| Sunny | **Review Files and deliverables** | **45 min** | **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 |
| ALL | As this is final milestone, everyone team members is contributing to milestone to make it ready for final submission |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Confirmation on Acceptance testing | **Executing acceptance test and listed out Debugs for matrix** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| SCRUM | **All contributed** |
| GIT And Github | **Used for version control and repository management** |
| JIRA | **Track workflow and updating tasks** |
| meeting | **All attended** |
| Milestone 6 | **Everyone started working** |
|  |  |
|  |  |

**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. 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 various merits of updating the function test matrix in addition to the test report, which supports a comprehensive and methodical approach to software testing and quality assurance. The matrix encourages thorough coverage throughout the product by acting as an accessible and transparent record of the features or parts that have been tested. Because traceability connects test cases to matching code parts, it promotes accountability. Updating the matrix promotes strategic test planning, which facilitates effective resource allocation and the ranking of essential tests in order of importance. In addition, it allows risk management by drawing attention to parts that receive little test coverage, which helps identify potential flaws. The matrix's function in analyzing the scope of testing provides valuable insight of the quality of the program and helps making decisions within release readiness. An updated matrix enables regression testing in the context of modified software by recognizing required test repetitions after modifications to the code. The development, testing, and management teams can work together and communicate more easily when it is provided with a common reference point. By highlighting patterns and potential areas for testing process improvement, regular updates may help in continuous improvement efforts. The matrix provides benefits for monitoring and conformity in regulated businesses by demonstrating thorough testing procedures. Furthermore, it directs test automation by determining which functions are appropriate for automation, improving efficiency and reproducibility. In the end, the function test matrix becomes a vital tool for overseeing testing operations and ensuring an excellent quality of software when it is updated in combination with the test report.

1. 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?

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 using tools like issue tracking, branches, and pull requests, we were able to efficiently assign work, resolve issues, and keep a thorough record of project developments. Jira was a flexible project management application that gave us a clear picture of project milestones and helped us plan, monitor, and prioritize activities. By using Jira and GitHub together, our team was able to coordinate activities, 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 improve 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 from previous mistakes. We've improved our strategy by analyzing what worked and what didn't, and we've used these learnings to improve upcoming milestones. Working together, we utilized the benefit of our combined experience and implemented the knowledge gained to encourage ongoing progress. By means of open communication and mutual responsibility, we have not only tackled obstacles but also improved our output. We have adapted because we are committed to learning from the past, which has led to more effective outcomes and more seamless 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 a very few 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 criteria and requirements. This created difficulties in delivering a dependable product and satisfying user expectations.

Reflection on several project aspects was spurred by the failure of acceptance tests. It sparked concerns regarding the efficacy of the development strategy, the precision of the requirements gathering procedure, and the breadth of the initial testing plan. The codebase was reevaluated to find the core causes of the failed tests and places that needed bug patches or additional development. The diagnosis and resolution of these problems required more time and work from the project team, which influenced project schedules, resource allocation, and ultimate project delivery.