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

**Members Present**:

|  |  |
| --- | --- |
| 1. Jenna Moon | 4. Farouk Alhassan |
| 2. Hyunjoo Han | 5. |
| 3. Taehwa Hong | 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** |
| **Hyunjoo Han** | **integration testing/unit testing automation/finish coding/acceptance testing/scrum report and reflection/Jira** | **None** |
| **Jenna Moon** | **integration testing/Requirement traceability matrix/scum report and reflection/Jira** | **None** |
| **Taehwa Hong** | **Integration testing** | **None** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

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** | **Function implementation fix match took some time** |
| **Reason for delay or block** | **N/A** |
| **Impact on Project** | **N/A** |
| **Solution or work-around** | **Putting time and figuring it out** |
|  |  |
| **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 |
| Final Report | **Discuss the method of Final Report** | **Planned and assigned** |
| Acceptance test | **Discuss the method of Acceptance Test** | **Planned and assigned** |
| Test execution | **Discuss the method of Test execution** | **Planned and assigned** |
| Scrum Report | **Discuss the method** | **Planned and assigned** |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Final testing report. | Ensure a comprehensive overview of the testing process, outcomes, and any identified issues. |
| Execute acceptance tests and debug. | Ensure that the software meets user requirements, identifying and resolving any issues that may impact the functionality or user experience. |
| Requirement Traceability Matrix | Ensure Requirement Traceability is up to date and reflected on the Final Testing Report |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Hyunjoo Han | Execute acceptance tests and debug, Final testing report, Jira, scrum report reflection, and GitHub Management. | **3 hours** | **Yes** |
| Jenna Moon | Requirement Traceability Matrix and scrum report reflection, Jira and GitHub Management. | **3 hours** | **Yes** |
| Taehwa Hong | scrum report reflection | **1 hour** | **Yes** |
| Farouk Alhassan | scrum report reflection | **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 |
|  | N/A- Milestones are done |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 program** |
| Jira and GitHub Management | **Organize the project** |
| Final Report Discussed | **Final Report Planned and outlined. The purpose of the Final Report was discussed.** |
| Scrum Report | **Scrum report is done. Updated for MS 6.** |
|  |  |
|  |  |
|  |  |

**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 |
| Acceptance testing | **Executed successfully to fulfill the purpose of the project** |
| Requirement Traceability Matrix | **Requirements are up to date and tests are fulfilled to meet the requirements.** |
| GitHub | **Useful for version control and keeping track of changes** |
| Jira | **All contributed and tasks are organized** |
| Scrum Report | **Final scrum report has been discussed and outlined.** |
| Update reflection | **Completed updated version of reflection for milestone 6.** |
|  |  |

**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 |
| limited time | **Busy schedule and communication problems** |
| workload | **Other work/assignment commitments with limited time** |
| Testing issues and the program | **Lack of resources** |
|  |  |
|  |  |
|  |  |
|  |  |

**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?

Regularly updating the test matrix with the test report is a cornerstone of effective software testing and quality assurance. The function test matrix serves as a visual guide, ensuring a systematic testing approach by meticulously documenting tested functions and their outcomes. It plays a pivotal role in enabling efficient test planning and execution, providing real-time insights into the tested portions of the program. This real-time visibility allows us to focus on areas requiring modifications, promoting traceability, and establishing clear links between test cases and relevant code segments, fostering accountability. The matrix's role in assessing testing comprehensiveness offers valuable insights into software quality, aiding informed decisions about the software's readiness for release. It serves as a potent risk management tool, identifying areas with limited test coverage and potential vulnerabilities.  
 Beyond its impact on testing procedures, updating the function test matrix contributes to effective communication among team members, fostering collaboration and feedback exchange. This transparent communication enhances team cohesion and overall project success. Moreover, the matrix streamlines regression testing by guiding necessary test reruns following code changes, supporting continuous improvement efforts, and identifying opportunities for enhancing the testing process. The function test matrix's significance extends to regulated industries, offering compliance benefits and valuable insights for continuous improvement. It also guides test automation efforts, improving efficiency and repeatability. Integrated with the test report, the function test matrix ensures a comprehensive oversight of testing activities, affirming the software's quality.

1. 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?  
     
    Using GitHub and Jira really helped us manage our project better. GitHub's strong control over versions and features that let us work together made handling code and collaborating super smooth. Features like pull requests and issue tracking made it easy to assign tasks and fix problems quickly. At the same time, Jira's tools for managing projects gave us a clear picture of our goals, helping us plan, track, and prioritize tasks effectively. Using GitHub and Jira together made our project management organized and helped us handle the project's complexities.

However, our team encountered difficulties managing the workload and adhering to the project schedule, primarily due to communication challenges. While we initially worked well, as the project complexities increased, some team members faced confusion and felt lost. The challenges stemmed from poor communication and a lack of understanding of certain project aspects among group members. The fast-paced nature of the project, coupled with limited time, posed significant hurdles. Despite collective efforts, these constraints impacted on our ability to coordinate tasks and meet deadlines. Stronger communication channels and more time would have greatly assisted in overcoming these obstacles. Although GitHub and Jira were valuable tools, the rapid project pace led to communication gaps, affecting task assignment and conflict resolution. In hindsight, addressing these communication issues early on would have been pivotal for improved project management and a smoother workflow.

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?  
     
    Our team consistently reviewed our performance after each milestone, examining both successes and challenges. This reflective practice was crucial for continuous improvement, providing valuable insights to refine our processes for the next steps. Every milestone served as an opportunity for collective learning, driving enhancements in our overall approach. Collaborative efforts, marked by open communication and shared responsibility, were key contributors. Recognizing the value of learning from mistakes, we actively addressed issues, resulting in tangible improvements in subsequent milestones. Our commitment to adaptability and learning, regardless of the experience's nature, demonstrated our dedication to continual enhancement.   
    Looking back, we recognize the significance of proactive measures to improve communication and meet deadlines for each task. Implementing strategies to identify better communication channels and refining our time management could have optimized our team's efficiency and success in every milestone. Embracing this proactive mindset would undoubtedly have led to even more significant strides in our collaborative endeavors.

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?

Even if the code passes black-box and white-box testing, integration and acceptance testing are essential. Integration testing ensures that different components work together perfectly, revealing potential problems when combined. Acceptance testing validates overall system functionality to ensure it meets business requirements. While black-and-white box testing focuses on specific functionality, integration and acceptance testing provide a holistic view, ensuring robust system behaviour and smooth integration even when individual devices have passed previous testing, ensuring a stable and effective software system. Provides a comprehensive verification approach.

At the beginning, we focused on addressing the issues of the tests that passed, given the constraints on fixing bugs within the limited time frame. Notably, we couldn't achieve a perfect pass due to these time constraints. 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.

Addressing these test failures necessitated a meticulous reassessment of the codebase, leading to additional time and effort invested by the project team. The project team had to invest additional time and effort to diagnose and diagnose these issues, which in turn impacted project timelines, resource allocation, and overall project delivery. This experience served as a valuable lesson, emphasizing the need for a more detailed approach in subsequent testing phases and reinforcing the commitment to delivering high-quality software.