# 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**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1.Yatin Bawa | 4. Rehatpreet Kaur |
| 2. Muhammetyar Yarov | 5. Aayush Bhogal |
| 3. Nehmat Ladhar | 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 4 days after your lab day:**

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

**Rubric:**

|  |  |  |
| --- | --- | --- |
| **Individual** | Group participation (includes GitHub commits and Jira usage) | 80% |
| Teamwork | 20% |
| **Group** | Project code running and executing successfully | 10% |
| 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 | 20% |
| Meets deadlines | 5% |

**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** |
| **Yatin Bawa** | * **Fixing bugs and updating code** * **Acceptance testing** * **Test report completion** |  |
| **Muhammetyar Yarov** | * **Helping in acceptance testing** * **Compiling result** |  |
| **Aayush Bhogal** | * **Documenting reflect questions** * **Helping in acceptance testing** |  |
| **Rehatpreet kaur** | * **Documenting scrum report** * **Completing test report** * **Helping in acceptance testing** |  |
| **Nehmat ladhar** | * **Updating tracebility matrix** * **Helping in acceptance testing** * **Helping in bug fixing** |  |
|  |  |  |
|  |  |  |

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 |
| Assigning roles | **Successfully roles assigned to all team membes and everyone started working on the roles assigned** | **Working of ilestone started** |
| Discussing Bugs | **Bugs discussed and process to fix them initiated** | **Fixing code started** |
| Discussion about acceptance testing | **Ran Acceptance testing by passing different inputs** | **Accptance testing done** |
| Dicussion on Scrum and relfect questions | **Reflect questions discussed** | **Documentation initiated** |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Members to start working on the assigned tasks with clear instructions. | Success |
| Discussion on bugs | Success |
| Acceptance testing done | Success |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Yatin Bawa | **Fixing code , acceptance testing** | **1 hr** | **Partially** |
| Muhammetyar Yarov | **Acceptance Testing** | **40 hr** | **Partially** |
| Nehmat Ladhar | **Updating Tracebility matrix, acceptance testing** | **40 min** | **Done** |
| Rehatpreet kaur | **Docmenting scrum report , acceptcance testing** | **40 min** | **Done** |
| Aayush Bhogal | **Documenting reflect questions , acceptance testing** | **40 min** | **Done** |
|  |  |  |  |
|  |  |  |  |

**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 |
| Trying to fix Bugs | **Can start working on acceptance testing** |
| Acceptance testing | **Checking reliablity of the software to be delivered** |
| Tracebility matrix | **Checking the requirements our software meet with the bussiness requirements.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Alloting tasks to team members | **Team contribution** |
| Bug fixing | **Team work** |
| Tracebility matrix | **Group participation** |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Bug fixing | **All the bugs were not fixed, There was some issues with the code supplied by college** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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?

The test report lists all the tests performed during the testing process but fails to deliver how it is related to the customer requirements which does not show us if the testing covers all the required criteria. Traceabilty matrix plays a crucial role in large projects as it shows how each test relates to each requirement. It helps the progress of the overall project by ensuring –

1) The function test matrix helps establish a clear link between individual tests and specific customer requirements. This traceability ensures that every aspect of the software is tested against the defined criteria, providing a clear view of how well the system meets its intended purpose.

2) The matrix allows for a systematic analysis of test coverage, highlighting areas where testing may be lacking or where additional tests are needed. This ensures that all critical functionalities are thoroughly tested, reducing the risk of undiscovered issues in the final product.

3) By comparing the function test matrix with the project requirements, it becomes easier to identify any gaps in testing coverage. This is crucial for addressing potential issues early in the development cycle and ensures that all aspects of the software are scrutinized for correctness.

4) The matrix aids in assessing the impact of changes to the software. When updates or modifications are made, the function test matrix helps identify which tests need to be re-executed to validate the changes and maintain the integrity of the overall system.

5) Through the function test matrix, potential risks associated with untested or under-tested functionalities can be identified. This proactive approach allows teams to prioritize testing efforts based on the criticality of features and potential impact on the end-user experience.

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?

Our team's performance was a dynamic interplay of strengths and challenges. Effective teamwork was characterized by clear communication, well-defined roles, and collaborative decision-making. These strengths were crucial in navigating project complexities and achieving milestones. However, we also faced challenges like scheduling conflicts and varied priorities, which occasionally led to workflow disruptions. Our response to these challenges was proactive, involving regular meetings and fostering a culture of open communication. This approach was instrumental in addressing misunderstandings and realigning our team's efforts towards common goals, ultimately contributing to a productive and harmonious team environment.

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 journey through the project was marked by a continuous cycle of action, reflection, and adaptation. After each milestone, we engaged in reflective discussions, analyzing what worked well and what didn't. This reflective practice enabled us to identify areas for improvement, particularly in communication and task management. Implementing these insights, we observed a noticeable enhancement in our team dynamics and project outcomes. This commitment to learning and adapting not only improved our immediate project performance but also equipped us with valuable skills for future 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?

Our comprehensive testing process was pivotal in ensuring the final success of the project. Initially, we encountered several critical issues that were diligently addressed over time. This process involved an iterative cycle of testing, identifying problems, and implementing solutions, highlighting our team's commitment to quality. The challenges we faced in testing were significant but surmountable, leading to a robust and reliable software product. Ultimately, our perseverance and attention to detail in the testing phase were key factors in overcoming these hurdles and achieving a successful project outcome.