# **MILESTONE 3** -- 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. Chang Cui | 4. Xiaofei Xu |
| 2. Lok Yin Tai | 5. Xinyang Wu |
| 3. Siu Man Cheng | 6. Ye Tian |

## Milestone 3 Tasks

In this milestone you will create issues to design the functions, design all of the functions you need to complete the project and store the specifications in the repository. As soon as the specifications start to be produced, you can start to design the blackbox tests (what they test, how to perform them and test data). Once tests are written, they can be implemented and added to the repository and any team members not otherwise busy can start to implement the functions. You will also build a function-test matrix that shows the blackbox tests for each function. This will be maintained through the testing cycle as new tests are added.

**Deliverables Due at end of Lab:**

* Completed SCRUM report and reflections

**Deliverables Due at 23:59 6 Days after Lab:**

* A set of function specifications stored in the repository,
* A set of blackbox tests as test documents with test data for the functions.
* Start writing blackbox test code and store in repository. (at least 1 required)
* Start implementing functions and store in repository. (optional)
* A function-test matrix added to the repository.
* Updated Jira project to show activities and progress.

**Rubric**

|  |  |  |
| --- | --- | --- |
| Individual | Group Participation | 75% |
| Teamwork | 10% |
| SCRUM Report | 15% |
| Group | Function Specs (documented, correct, complete, well-written) | 20% |
| Test documents (well-written, complete, good test data) | 20% |
| Test Code (well-designed, written and documented) | 10% |
| Git Usage (used properly with good structure) | 5% |
| Jira Usage (creates issues, tracks progress) | 10% |
| Meets Deadlines | 10% |
| SCRUM report & reflections | 25% |

**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** |
| Chang Cui | SCRUM Report  Reflections  Jira Board Management  GitHub Repository Management | N/A |
| Lok Yin Tai | Test Plan Control Procedures  Test Plan Resources and Responsibilities  Test Plan Deliverables Session  Jira Board Management  GitHub Repository Management | N/A |
| Siu Man Cheng | Data structures  Test Plan Tools  Test Plan Documentation  Test Plan Approvals  Jira Board Management  GitHub Repository Management | N/A |
| Xiaofei Xu | Test Plan Introduction  Test Plan Scope  Test Plan Test Strategy  Jira Board Management  GitHub Repository Management | N/A |
| Xinyang Wu | Test Plan Environment Requirements  Test Plan Execution Strategy  Test Plan Test Schedule  Jira Board Management  GitHub Repository Management | N/A |
| Ye Tian | Test Plan Suspension / Exit Criteria  Test Plan Resumption Criteria  Test Plan Dependencies  Test Plan Risks  Jira Board Management  GitHub Repository Management | 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 |
| Review on last week work | The team reviewed the work and difficulties from the previous week. The team provided comments to help each other. | Identified areas for improvement and appreciate each other. |
| Improvement base on professor comment and grading | The team read the feedback together and discussed strategies to do better for upcoming milestones. | Made immediate revision and seek for professor comment in person. |
| Summary of lecture content | The team shared their understanding and key takeways of the lecture content and how it is related to MS3. | Ensured everyone has a clear understanding on lecture material and requirements on MS3 |
| Discussion on MS3 task distribution | The team discussed the tasks required for Milestone 3 and divided into subtasks for task allocation | Assigned responsibilities and distribute tasks among team members using random drawing |
| Knowledge sharing on milestone 3 deliverables | The team discussed the requirements and expectations for each deliverable | Everyone has same expectation and direction on MS3 to work for the best |
| Discussion on deliverables dependency and due dates | The team discussed the dependencies between deliverables and reviewed the individual due dates | Set up reasonable due date for individual tasks to ensure timely completion with high quality submission |

**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 |
| Tasks are divided into 6 parts based on workload | Ensured a fair distribution of work among team members, with efficiency and productivity. |
| Task distributed using random dice game | Provided a fair and unbiased method for task distribution, eliminating bias in assigning tasks. |
| Clarified the milestone deliverables and document to be submitted | Ensured a shared understanding of project requirements, preventing misunderstandings and facilitating efficient progress. |
| Agreed to Improve team collaboration by using Jira comments | Utilized a centralized platform for communication, enabling effective collaboration and feedback exchange. |
| Jira task ID number will be recorded in test document | Facilitated traceability and reference between tasks and test documentation. |
| Document version control will be executed by using GitHub | Enabled efficient tracking and management of document versions, ensuring accurate and organized version control. |
| Set up a due date for team documents review and feedback | Reserved a buffer period for reviewing and providing feedback on team documents, promoting timely progress and improvements. |
| Established goal to achieving high grading and submit quality deliverables in each milestone | Ensured everyone is focusing on delivering high-quality work, aligning with project objectives and expectations. |

**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? |
| Chang Cui | Test documents | 50 mins | Yes |
|  | Test Traceability Matrix | 10 mins | Yes |
|  | Jira Board Management | 10 mins | Yes |
|  | GitHub Repository Management | 10 mins | Yes |
| Lok Yin Tai | Functions Spec | 60 mins | Yes |
|  | Jira Board Management | 10 mins | Yes |
|  | GitHub Repository Management | 10 mins | Yes |
| Siu Man Cheng | Functions Spec | 60 mins | Yes |
|  | Jira Board Management | 10 mins | Yes |
|  | GitHub Repository Management | 10 mins | Yes |
| Xiaofei Xu | Reflection | 60 mins | Yes |
|  | Jira Board Management | 10 mins | Yes |
|  | GitHub Repository Management | 10 mins | Yes |
| Xinyang Wu | Test Code | 60 mins | Yes |
|  | Jira Board Management | 10 mins | Yes |
|  | GitHub Repository Management | 10 mins | Yes |
| Ye Tian | Test documents | 50 mins | Yes |
|  | Test Traceability Matrix | 10 mins | Yes |
|  | Jira Board Management | 10 mins | Yes |
|  | GitHub Repository Management | 10 mins | 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 |
| Chang Cui | * Test documents * Test Traceability Matrix * Jira Board Management * GitHub Repository Management |
| Lok Yin Tai | * Function Specs * Jira Board Management * GitHub Repository Management |
| Siu Man Cheng | * Function Specs * SCRUM Report * Jira Board Management * GitHub Repository Management |
| Xiaofei Xu | * Reflections * Jira Board Management * GitHub Repository Management |
| Xinyang Wu | * Test Code * Jira Board Management * GitHub Repository Management |
| Ye Tian | * Test documents * Test Traceability Matrix * Jira Board Management * GitHub Repository Management |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Comprehensive clarification on the interrelation of tasks | Enabled better planning and coordination, with task dependencies consideration to ensure everyone has enough time to work on their own tasks. |
| Tasks for MS3 divided into six parts based on workload and difficulty | Optimized productivity and individual contributions by promoting fair distribution of work. |
| Agreed on the methodology to improve based on MS2 feedback | Learned from past mistakes and improve upcoming submission. |
| Clarify the deliverables and all requirements | Eliminated misunderstanding and ensured everyone are clear with requirements and instructions. |
| Mutual objective to strive for quality work | Defined a shared understanding of project expectations and goals, strive for 100/100 for each work. |

**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 |
| Clear clarification on tasks dependencies for MS3 with individual due dates | Ensured every member fully understood the assigned tasks, with individual due dates. The team can effectively plan and work ahead of deadline. |
| Equal workload and tasks allocation for MS3 | Every member can do their best with balanced workload and time allocation. |
| Establish same goal and target on quality submission | Every member shared understanding and commitment to achieving a high-quality submission. |
| Valuable knowledge sharing and discussion on upcoming deliverables | Gained mutual support by open discussions and insights sharing and promoted an effective teamwork. |
| Active collaboration and problem solving | Encouraged teamwork and collaboration in problem-solving to achieve positive team culture and improve work efficiency |

**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 | The meeting went well and smooth. All planned topics are discussed with no outstanding item.  We shared our own knowledge and opinions towards the project. |

**Reflections**:

1. In this milestone, we write the blackbox tests but not the whitebox tests. Explain why we can write the blackbox tests but not the whitebox tests.
2. Lack access to the source code: black box test concentrates on testing system’s functionality based on its external behavior and characteristics, and we don’t need to know what is inside the function. We just need to know if we put a certain value into the function then a certain value should come out of the box. In this milestone, we have prototypes of functions, but we don’t have functions’ implementation. In this case, we only can do black box testing instead of white box testing.
3. Black box testing is normally the first approach to designing unit tests. In this milestone, we use black box testing to validate if the system functions can work correctly and the system’s behavior can meet our requirements.
4. Explain why we need the function-test matrix and why it is important in a large project.  
     
   The function-test matrix helps us to know the relationship between different system’s functions and related test cases. It provides an organized and structured way to show us the specific tests needed to be done for each function.

It is important in a large project because:  
1) Large project normally has various functions, and it’s hard to make sure that all aspects of the system are tested. The function-test matrix can help to identify specific tests required for each function to ensure comprehensive test coverage across the entire system.

2) It provides traceability between requirements and corresponding test cases. Using function-test matrix can make sure all specified functionalities are sufficiently tested and test cases are satisfied with the project’s requirements.

3) It can be a communication tool for team members. Developers and testers can know which functions have been tested, testing results and issues in testing through using function-test matrix. It contributes to team collaboration and coordination.

1. Other life cycle models left team members idle while waiting for parts of the project to be completed. Describe how an agile model, like the one we are using, avoids this problem and keeps the whole team busy all the time. Does this make managing the project simpler or more complex and why?
2. Iterative development: we are using Scrum and Kanban, which divide our whole project into small iterations. Our team members can focus on different parts of our project at the same time. After they complete their iteration, they can move on to the next one while waiting for feedbacks.
3. Cross-functional teams: In agile models, team members have different skills, which makes the team can work on different parts of the project. For example, some team members can work on coding, while other members can work on designing testing and writing reflection simultaneously. In this case, the whole team can participate in the entire project.
4. Team collaboration: Through group meetings, using Jira to assign tasks, review and get feedback from group members, every member can know the progress of our project and can contribute their opinions and expertise.

Simpler or more complex:

Agile model offers a good way to ensure each group member know the progress of project clearly and track tasks easily. Therefore, group members can identify and address issues efficiently. Its iterative nature can ensure all members work on their individual part simultaneously and make continuous improvement. However, agile model needs active and ongoing management, which may be challenging when managing larger projects. Project leader should pay attention to ensure effective coordination among group members, promoting communication and managing priorities efficiently.