# **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**: C

**Members Present**:

|  |  |
| --- | --- |
| 1.Manav Alpeshbhai Zadafiya | 4.Sunny Vavadiya |
| 2.Mohamed Ashraf Bharot | 5. |
| 3. Fenil Soni | 6. |

## 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** |
| **Manav** | **Milestone 3 planning, function Implementation, review functions and test plan** | **N/A** |
| **Fenil** | **Function Designing and specs documentation in word file.** | **N/A** |
| **Ashraf** | **Function Designing and add function description to header file** | **N/A** |
| **Sunny** | **Black box test** | **N/A** |
| **All** | **Scrum 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 |
| Milestone 3 | **Quick overview of the milestone objectives, which includes the scrum report, Functions, and Black-box Test.** | **All members understood the milestone objective** |
| Scrum report | **topics to document in scrum report and appropriate answers to reflections** | **Scrum report completed with help of all members** |
| Functions designing | **Who will design functions and document its specification in header and word file.** | **Task assigned to two members: Fenil and Ashraf** |
| Black-Box Testing | **how to write black box test code, how to run it, and what test data to use. The repository was updated with the black box test code.** | **The team had understood test plan and task has been assigned to sunny** |
| Implementing functions | **Who will implement the functions into implementation file and update the repository with newly added content.** | **Task assigned to Manav** |
| Building a function-test matrix | **The group created a matrix of functions and tests that displayed the blackbox tests for each function. As additional tests were added or altered, the matrix was updated. The repository housed the matrix.** | **The team had a thorough overview of the testing coverage and outcomes for each function.** |
| Milestone 4 Overview | **What is objective of ms4 and what is expected form us.** | **Everyone get familiar with ms4.** |

**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 |
| Architectural Decision | we decided on a modular design pattern-based architectural strategy to improve code reuse and maintainability. |
| Function Prioritization | The team listed the functions in order of importance and dependencies to make sure that the most crucial ones were implemented first. |
| Testing Strategy | To ensure thorough test coverage and find any potential issues early in the development process, it was decided to combine unit testing, integration testing, and black-box testing. |
| Task Assignment | Each team member was given a specific task to carry out, and a deadline was established for finishing it. Team members were urged to work together and support one another to maintain progress. |
| Documentation Standardization | The group decided on a template and format for function specifications, black-box tests, and test code documentation. The project will be consistent thanks to this standardization, and future maintenance will be simpler. |
| Issue Tracking | The team established a procedure for using Jira to track issues and bugs. To ensure a smooth development workflow, issues will be logged, assigned to team members, and tracked until resolution. |
| Git Workflow | The group decided to use a branching model for coding, adding features to feature branches, testing them, and then merging them into the main branch. |

**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** | **35 min.** | **yes** |
| manav | **Function implementation** | **1 hr** | **yes** |
| Fenil, Ashraf | **Writing function specifications** | **1 hr** | **yes** |
| sunny | **Black box testing** | **45 min.** | **yes** |
| All | **Jira to update status and manage workflow** |  |  |
| All | **Use Git to push and commit changes to repository** |  |  |
|  |  |  |  |

**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 |
| Manav | Implement function definitions |
| sunny | Writing Black box and white box test for functions |
| Fenil | Execute black box test and document result |
| Ashraf | Execute white box test and document result |
| ALL | Update repository and complete preparation for test automation |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Function design | **Add value to software by including functionalities** |
| Understand algorithm for shortest path | **Help to define function and provide effective delivery services.** |
| Functions understanding | **All team members understand functions so they can work with it and test it accordingly.** |
| Potential Issues in projects | **Issues such as vehicles becoming caught at the map's edge or in building corners have been identified. Recognizing and fixing these difficulties strengthens the algorithm, lowering the likelihood of delivery failures owing to vehicles being unable to reach locations.** |
| Refine Plan | **Improve overall plan for more stability and timely deliverables** |
|  |  |
|  |  |

**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 understanding | **Team had a clear understanding of milestone objectives** |
| Effective task division | **Tasks were successfully divided among team members** |
| Comprehensive function design | **Productive discussion on designing required functions** |
| Well-defined test strategy | **Defined strategy for writing black-box test code** |
| Collaboration in implementation | **Members collaborated and supported each other during implementation** |

**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. **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.**

In this workshop, we have just design functions and not actually define its implementation. As white box tests focus on internal structure of functions it is not possible yet to write whitebox tests. Moreover, we were focused on the functionality and behavior of the functions rather than the internal implementation details in this milestone. Blackbox testing is carried out based on the functions' expected inputs, outputs, and needs, without requiring knowledge of their underlying code. We may develop Blackbox tests to ensure that the functions fulfil the function requirements, which outline the intended behavior and functionality. However, Whitebox testing, on the other hand, need knowledge of the underlying structure and code implementation, both of which are currently unavailable at this milestone.

1. **Explain why we need the function-test matrix and why it is important in a large project.**

The function-test matrix is vital in a large project because it gives a systematic and organized way of ensuring that all functions are properly tested. It aids in the mapping of functions to their related test cases, guaranteeing thorough coverage of various circumstances. The function-test matrix also acts as a reference document for tracking testing progress and spotting any gaps or missing test cases. It helps to guarantee that all functionalities are fully tested, lowering the possibility of undetected flaws or faults in the system.

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?**

Team members are consistently engaged and occupied throughout the project under an agile model, such as the one being implemented in milestone 3. This is accomplished through agile development's iterative and incremental nature, in which work is broken into tiny, manageable tasks or user stories. Unlike previous life cycle models, where team members may have to wait for sequential stages to be finished, it also allows team members to work on different user stories or tasks concurrently. As soon as one user narrative is finished, we share with each other, take confirmation by everyone and then, the team may move on to the next text which maximizes the production of our group. We kept on adding new tasks and possibilities for cooperation, this agile method keeps the entire team occupied. It reduced idle time and reduced reliance on certain personnel or project stages. Throughout the development cycle, team members were able to contribute their talents and knowledge, resulting in a more efficient and streamlined process. It makes the project management more straightforward overall by encouraging flexibility, adaptability, and continuous improvement. Because agile is iterative, regular feedback and course corrections are possible, allowing the team to adapt rapidly to changes and provide high-quality products in shorter iterations. Agile approaches also encourage improved team communication, transparency, and cooperation, which can result in more effective project management and successful project outcomes.