# **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**: \_\_\_\_**1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Members Present**:

|  |  |
| --- | --- |
| 1. Taehwa Hong | 4. Hyunjoo Han |
| 2. Jenna Moon | 5. Natalya Pak |
| 3. Farouk Alhassan | 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 4 days after your lab day:**

* A set of AT LEAST 4 function specifications stored in the repository.
* A set of blackbox tests as test documents with test data for the functions you created. At least 4 sets of test data are required for each function. You must have test cases for at least 6 functions (including all your custom function). Stored in the repository.
* Start writing blackbox test code (for the functions above) and store in repository (at least 1 is required for this milestone).
* Start implementing the functions and store them in repository (optional).
* A requirements traceability matrix added to the repository and shows the mapping between the requirements and test cases.
* Updated Jira project to show activities and progress.
* Completed scrum report including reflection questions answered.

**Rubric**

|  |  |  |
| --- | --- | --- |
| **Individual** | Group participation (includes GitHub commits and Jira usage) | 80% |
| Teamwork | 20% |
| **Group** | Function specifications (documented, complete, well-written, added to the project) | 10% |
| Blackbox test cases document (well-written, complete, good test data) | 15% |
| Blackbox test code (well-designed and documented) | 10% |
| Functions implementation (coded in the C project & well documented) | 10% |
| Requirements traceability matrix (complete, added to GitHub) | 10% |
| Git usage (used properly with good structure) | 5% |
| Jira usage (creates issues, tracks progress) | 10% |
| Scrum report & reflections | 20% |
| Meets deadlines | 10% |

**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** |
| **Taehwa Hong** | **SCRUM, Function specifications** | **N/A** |
| **Jenna Moon** | **SCRUM, Complete and add a requirements traceability matrix.** | **N/A** |
| **Farouk Alhassan** | **SCRUM, Create blackbox tests as test documents.** | **N/A** |
| **Hyunjoo Han** | **SCRUM, Function specifications** | **N/A** |
| **Natalya Pak** | **SCRUM, Write blackbox test code.** | **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 |
| Function Specification | **Function specifications and document discussed together according to new functions written on header file added last week.** | **Function discussed together** |
| SCRUM | **SCRUM report planned** | **SCRUM is completed and will be updated** |
| Testing Functions | **Testing Functions were discussed as it is black box testing** | **Testing Functions discussed but not executed yet** |
| Testing Docs | **Testing Document has been assigned** | **Testing document has been assigned and will be completed to add to repo** |
| Jira | **Task assignment and schedule has been set up in Jira** | **Completed** |
| Git | **Git update to each branch** | **Completed** |
|  |  |  |

**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 | Reasonably equal number of tasks have been assigned to each member of team. |
| Testing Decision | Planned to do tests on position detection and map generating. |
| Testing Functions Implementation | Black box will cover basic functionality. |
| Function Specifications | Added function specifications on existing header file and documentation completed. |
|  |  |
|  |  |
|  |  |

**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 | **Analysis discussion of testing requirements in program and discussed** | **1hr** | **Yes** |
| ALL | **Analysis discussion of Function Specs in program** | **1hr** | **Yes** |
| ALL | **Scrum report** | **40min** | **Yes** |
| ALL | **Jira and GitHub project page updated and assigned** | **30 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 | Attending weekly team meetings |
| ALL | Jira Management and making updated comments |
| ALL | SCRUM report and reflection |
| ALL | Function Implementation |
| ALL | Test execution |
| ALL | Debugging |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Function specs | **Functions specs discussed as a group using the data structure implemented in MS2** |
| Testing documentation | **Documentation planned as a group during the meeting** |
| Testing codes | **Black box testing code implemented according to discussion in meeting.** |
|  |  |
|  |  |
|  |  |
|  |  |

**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 | **Keeping track of changes and useful for version control** |
| Meeting | **All attended meeting** |
| Documentations | **All documented that what should be done, testing and functions specs** |
|  |  |
|  |  |
|  |  |

**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 | **N/A** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Reflections**:

Answer the following questions using your own words. Make sure that each answer comprises a minimum of 100 words.

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.

The main reason we're prioritizing black-box testing over white-box testing in this phase is because we don't have a lot of detailed information about the inner workings of the system we're testing. Since our understanding of how the system operates internally is limited, it makes more sense to create tests based on our expectations of how the system should behave. Black-box testing allows us to assess the system's performance from a user's perspective without delving into all the technical details. This approach helps us catch significant issues early on and address them before they escalate into larger problems in the code. Later on, we may use white-box testing to dig deeper into the internal details of the code.

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

The function-test matrix is an important organizational tool in large software projects, utilizing a spreadsheet or table to systematically track features, options, and testing areas. It is important because it ensures comprehensive test coverage, traceability, risk assessment, and optimized resource allocation. By correlating software functions with corresponding test cases, the matrix facilitates thorough verification against expected standards and requirements. In large projects, it plays a key role in prioritizing, reusing, and documenting test cases, preventing omissions, and saving valuable time and resources.

The matrix serves as a communication tool, encouraging collaboration among team members and contributing to effective regression testing. Additionally, it is essential for compliance and documentation, capturing client requirements and serving as a traceability matrix for future reference. This structured representation helps avoid missing or repeating test cases, making sure that all functionalities align with specified standards. In essence, the function-test matrix is a strategic and efficient tool, enhancing the reliability and thoroughness of testing procedures in large-scale software development projects.

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?

In an agile model, such as the one being utilized, team members avoid idleness by working in iterative cycles known as sprints. Unlike traditional life cycle models that may leave team members waiting for project components to be completed, agile divides the project into manageable parts with specific goals. This approach keeps the entire team consistently engaged and focused. Regular communication and collaboration among team members ensure everyone is aware of progress and can actively contribute. Agile's emphasis on adaptability and continuous improvement simplifies project management by providing clear goals, fostering teamwork, and allowing for frequent feedback. The incremental nature of agile development ensures that the team remains busy, making project management more straightforward and responsive to changing requirements.