# **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**: **SFT221 NEE group1**

**Members Present**:

|  |  |
| --- | --- |
| 1. Kim Ming Chau | 4. Gordon Tan |
| 2. Hak Kan Poon | 5. Yuhong Fan |
| 3. Elvin Karikari | 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** |
| Gordon Tan | **Create Data structures in header file and upload to GitHub, Reflection 2 & 3** |  |
| Elvin Karikari | **Test Plan 1 - 4** |  |
| HK Poon | **Test Plan 5 - 8** |  |
| Yuhong Fan | **Test Plan 13 - 17** |  |
| Kim Ming Chau | **Test Plan 9 – 12, Reflection 1 & others in Scrum report** |  |
|  |  |  |
|  |  |  |

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 |
| Function Spec | **Functions would be added in MS3** | **Gordon draft a list in Github** |
| Black box doc | **What should be included in the doc** | **Member to prepare their parts for review** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Focus to create the function spec first | After outline the content of function spec, team members could work on designing black box test document separately. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Gordon Tan | **Prepare all the function in the header file & TB11** | **2 hrs** | **Y** |
| HK Poon | **TB01-TB03** | **2 hrs** | **Y** |
| Kim Ming Chau | **TB04-TB07** | **2 hrs** | **Y** |
| Yuhong Fan | **TB08-TB10** | **2 hrs** | **Y** |
| Elvin Karikari | **Scrum report & reflection** | **2 hrs** | **Y** |
|  |  |  |  |
|  |  |  |  |

**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 |
| Gordon Tan | Implemented Functions (well-designed, written and documented) |
| Elvin Karikari | Whitebox tests (well-designed, written and documented) |
| Yuhong Fan | Test Execution (performed, results recorded, issues created) |
| HK Poon | Debugging (Bugs fixed, documented, Jira updated) |
| Kim Ming Chau | Git Usage (used properly with good structure) |
| HK Poon | Jira Usage (creates issues, tracks progress) |
| Kim Ming Chau | Scrum Report & Reflections |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Content of function spec | **Outline major functions to finish the program** |
| Timeline to complete this Milestone | **Have a common ground how to co-operate with each other and the leadtime of its deliverable.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Co-operation on test cases | **Team members willing to help each other when we have difficulty to complete the task. Such as providing suggestion how to tackle, reviewing the document.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 milestone of testing the truck route management system, our chosen approach entails focusing on the development of blackbox tests rather than whitebox tests.

1. User-Centric Evaluation: By emphasizing blackbox testing, our primary goal is to assess the system's functionality and behavior from the perspective of end-users. We aim to validate that the truck route management system fulfills user requirements and operates as expected, without diving into the internal intricacies of the system's code or design.

2. Limited Visibility of Internal Implementation: At this stage, access to the internal codebase or detailed implementation information may be restricted. Hence, it is not practical or feasible to conduct whitebox testing, which typically requires a deep understanding of the system's internal structure and logic.

3. Requirement Compliance: This milestone primarily revolves around verifying whether the truck route management system adheres to the specified requirements and objectives. Blackbox testing enables us to validate the system's conformance to these requirements, ensuring that it delivers the desired functionalities and produces accurate outputs.

4. Early-Stage Functionality Evaluation: Given the current development phase, our focus is on quickly evaluating the overall functionality of the truck route management system and ensuring that it fulfills its intended purpose. Blackbox testing allows us to efficiently test the system's behavior without extensive knowledge of the internal workings or detailed code analysis.

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

The function-test matrix is essential in large projects like our truck route management system for the following reasons:

1. Comprehensive testing: It ensures thorough coverage of truck route functionalities by mapping each function to corresponding test cases.

2. Requirement traceability: The matrix establishes a clear link between system requirements and test cases, ensuring all requirements are addressed.

3. Prioritization and planning: It helps prioritize critical functions for focused testing, optimizing testing efforts.

4. Collaboration and communication: The matrix fosters effective collaboration among team members involved in testing the truck route system.

5. Adaptability: It allows for updates and regression testing as the truck route system evolves.

Overall, the function-test matrix streamlines testing, enhances communication, and ensures the reliability of our truck route management 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?

In an agile model like the one we're using, the issue of team members being idle while waiting for project components to be completed is effectively addressed. Agile methodologies, such as Scrum, employ practices that keep the entire team engaged and productive, avoiding idle time. Here's how an agile model accomplishes this:

1. Iterative Development: Agile projects are divided into smaller iterations or sprints, where each sprint focuses on delivering a usable product increment. This approach ensures that team members always have work in progress, allowing for continuous development and keeping everyone busy and actively involved.

2. Cross-Functional Teams: Agile teams are made up of individuals with diverse skills and expertise. This enables parallel work on different aspects of the project. While some team members may be coding, others can be involved in design, testing, or requirements gathering. This collaborative setup ensures that everyone has meaningful tasks to work on, maximizing productivity.

3. Collaboration and Communication: Agile methodologies emphasize frequent collaboration and communication among team members. Daily stand-up meetings, sprint planning sessions, and retrospectives foster an environment of open communication and coordination. This reduces idle time by identifying any obstacles or bottlenecks promptly and allows the team to address them together.

4. Backlog Prioritization: Agile projects maintain a prioritized product backlog, which is a dynamic list of user stories or requirements. Team members can choose the next highest-priority item from the backlog once they complete their tasks. This approach ensures a smooth workflow and minimizes downtime between tasks.

Using Agile methodologies provides clear visibility into the project's progress through visual tools and regular meetings. This transparency helps track progress, identify issues early on, and make necessary adjustments. The iterative nature of agile allows for incremental development and continuous feedback incorporation, simplifying project management.

In summary, an agile model like the one we're using effectively eliminates idle time and keeps the team engaged. It simplifies project management through transparency and flexibility but introduces challenges related to coordination and adaptability.