# **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. Leo Ru | 4. Frank Fu |
| 2. WaiSun Lam | 5. Kam Chun Stanley Tung |
| 3.Xinyang Ma | 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** |
| Frank Fu | **Function specifications** |  |
| Leo Ru  WaiSun Lam  Xinyang Ma | **Blackbox test cases, blackbox test code** |  |
| Kam Chun Stanley Tung | **Reflection, function-test matrix** |  |
| Leo Ru  WaiSun Lam  Xinyang Ma  Kam Chun Stanley Tung  Frank Fu | **Scrum report** |  |
| Xinyang Ma | **Update Jira** |  |
|  |  |  |
|  |  |  |

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 specifications | **Discussed/agreed upon additional function specifications** | **Approved** |
| Blackbox test cases | **Reviewed and discussed** | **Approved** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| To implement additional functions for input validation and error checking | Based on the sample output, user input (shipment) must be validated before finding the right truck. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Frank Fu | **Function specification for input validation functions** | **30min** | **completed** |
| XinYang Ma  Wai Sun Lam  Leo Ru | **Test cases for validation function** | **30min** | **completed** |
| Stanley Tung | **Reflection** | **1.2h** | **completed** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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 |
| Frank Fu | Implement functions |
| Xin Yang Ma  Wai Sun Lam  Leo Ru  Stanley Tung | Blackbox/ white box tests |
| Stanley Tung | hook ,Function-test matrix |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Tasks assigned for next milestone/ Completion of scrum report 3 | **Everyone on the same page and knows what needs to be done** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| quickly decided on division of labor without conflict | current phase of assignment has clearly outlined deliverables that made it easy for group members to understand the necessary tasks as well as likely workloads involved |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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, we are asked to create a set of function specifications instead of function implementation. The main objective is to focus on the behavior and functionality of the software. This objective aligns with the principles of blackbox testing.

Blackbox testing is an approach where the tester does not require knowledge of the internal workings or structure of the software. The test involves inputting certain data and checking whether the output aligns with the specification.

On the other hand, writing whitebox tests often require more time and resources, like testers with a deep understanding of the software. We can leave it to a later stage when we gain more knowledge about the software.

1. Explain why we need the function-test matrix and why it is important in a large project.  
     
     
   Function-test matrix is important because it ensures traceability. In a large project, the volume of functions and tests can be overwhelming, and it's easy to miss out testing some of the requirements. A function-test matrix can help prevent this.

Additionally, function-test matrix gives visual representation of which parts of the system have been tested and which parts are still untested. These facilities enable easier communication among development teams.

Function-test matrix can also serve as a form of documentation, showing the history of testing activities and how they've corresponded to the system's functions over time.

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?

Agile models break down development into smaller, manageable pieces. Each piece focuses on delivering a working piece of software that adds to the overall product. For this reason, work is continuously flowing and prevents team members from idleness. For example, once some of the project requirements are clear, team members can start designing functions to meet those requirements, without any need to wait for full project requirements to start working. If there is any change in requirement, team members will iterate and embrace the change.

The agile model can make managing the project simpler because it allows for flexibility and continuous adjustment. Issues are typically identified earlier in the development process, making them easier to handle. Agile model promotes cooperation and active communication, which helps to keep everyone informed and invested in the project.