# **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**: \_\_\_\_\_\_\_\_5\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1.Prabhjot Singh | 4.Prince Prince |
| 2.Dhruv Kakadiya | 5.Siya Khanna |
| 3. Sampreet Klair | 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** |
| **Prabhjot Singh** | **Scrum Report, Task allocation and reflection** | **-** |
| **Dhruv Kakadiya** | **Created the matrix to show function working and function description** | **-** |
| **Sampreet Klair** | **Managing Jira and Git** | **-** |
| **Prince Prince** | **Designing the BlackBox tests** | **-** |
| **Siya Khanna** | **Functions definitions** | **-** |
|  |  |  |
|  |  |  |

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** | **Nothing** |
| **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 Definitions | **Function requirements and function definitions** | **Done** |
| Blackbox Tests | **Discussed about the Blackbox tests** | **Done** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Managing Project as usual | Done |
| Scrum Report, Tasks allocation and reflection | Done |
| Blackbox tests | Done |
| Function Defintions | Done |
|  |  |
|  |  |
|  |  |

**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? |
| Prabhjot Singh | **Reflection discussion** | **10 min** | **Done** |
| Siya Khanna | **Function Definitions** | **40 min** | **Done** |
| Prince Prince | **Blackbox tests** | **30 min** | **Done** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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 |
| Prabhjot Singh | Scrum report, Reflection, task allocation and assistance |
| Siya Khanna | Function definitions |
| Dhruv Kakadiya | Help in Blackbox testing |
| Prince Prince | Blackbox Testing |
| Sampreet Klair | Managing Jira and Github repo |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Successful completion of various tasks | **Success towards completion** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Discussions | **Team plays** |
| Blackbox testing | **Knowledge of everyone** |
| Functions definitions | **Group discussions** |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Nothing | **Team co-ordination** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 write Blackbox tests overs Whitebox testing because:

* Early Testing stages: The software, in this case, is in its early stages of development. Writing the white box testing in this case, will be worth nothing.
* Blackbox testing allow us to independently test the software and testers can create the tests based on the specifications and requirements.
* When we want to validate the users interactions and overall user experience, Blackbox testing provides a valuable perspective without the need of detailed code analysis.

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

We need the function test matrix because it helps us to relate the function’s purpose with its intentional use as we can correlate the functions and backbox test ensuring the project proper functioning.

Here is why it is important in the large project:  
A. It gives a clear overview of which function have associated test cases and which are not.

B. Large projects often are really complex, and it helps to manage the risks where defects might have severe consequences.

C. It gives us the ability to regression testing where we can test new functions.

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?

Solving the problem of idle team members:

1. Firstly, the agile projects are divided into smaller portions which provide a chance for the other members to work simultaneously.
2. Secondly, agile projects are cross functional which involves people who are masters in different areas like testing, designing and a lot more.
3. Agile projects involve team members can focus on various parts of the project.

Making simpler or complex:  
Actually it makes project development easier as different people work on their own parts and the focus on continuous delivery stakeholders allows them to see the progress.