# **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.Yatin Bawa | 4. Rehatpreet Kaur |
| 2. Muhammetyar Yarov | 5. Aayush Bhogal |
| 3. Nehmat Ladhar | 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** |
| Yatin Bawa | * **Created function specification for function 2** * **Writing Black box test code for function 2 ( customised function )** * **Created black box test cases for a function** * **Worked on reflect question 3** * **Participated in dicussion of scrum report and reflect questions** | **-** |
| Muhammetyar Yarov | * **Created function specification for function 4** * **Worked on tracebility matrix** * **Created black box test cases for a function** * **Participated in dicussion of scrum report and reflect questions** | **-** |
| Nehmat Ladhar | * **Created function specification for function** * **Created black box test cases for a function** * **Worked on reflect question 2** * **Participated in dicussion of scrum report and reflect questions** |  |
| Rehatpreet Kaur | * **Created function specification for function** * **Created black box test cases for a function** * **Worked on reflect question 1** * **Documented scrum report** * **Participated in dicussion of scrum report and reflect questions** |  |
| Aayush Bhogal | * **Created black box test cases for 2 predefined functions** * **Participated in dicussion of scrum report and reflect questions** * **Worked on tracebility matrix** |  |
|  |  |  |
|  |  |  |

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 |
| Getting started with the milestone | **Reviewed the milestone, assigned tasks to the team members and discussed the approach to solve the tasks.** | **Due dates decided for each task** |
| Discussion on function specification | **Discussed about the possible functions that can be made which would be helpful in completing the code in future.** | **Function specifications finalised and members assigned task to make test cases for each function** |
| Discussing the test cases and traceability matrix | **Discussed if all the test cases are valid and how to work on traceability matrix** | **Test cases modified and traceability matrix discussed** |
| Completing the scrum | **Discussed and documented the Scrum report** | **Done with the scrum** |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Members to start working on the assigned tasks with clear instructions. | Success |
| Fucnctions decided to impletemnent and ready for black box test cases to be created | Success |
| Finalising the test cases, working of traceability matrix and skeleton answers to reflect questions | Success |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Yatin Bawa | **Discussed a function, made black box test cases and black box test code** | **2.5 hr** | **Yes** |
| Muhammetyar Yarov | **Discussed a function, made black box test cases and started working on traceability matrix** | **2 hr** | **Yes** |
| Nehmat Ladhar | **Discussed a function, made black box test cases and worked on reflect questions** | **3 hr** | **Yes** |
| Rehatpreet kaur | **Discussed a function, made black box test cases ,worked on reflect questions and documented scrum report** | **2 hrs** | **Yes** |
| Aayush Bhogal | **Discussed black box test cases for predefined functions and worked on traceability matrix** | **2hr** | **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 |
| Yatin Bawa | Whitebox tests with test data for functions , debugging and test execution |
| Muhammetyar Yarov | Coding the functions , test automation and Updated requirements traceability matrix stored in the repository. |
| Nehmat Ladhar | Coding functions , test automation and reflect questions |
| Rehatpreet kaur | Coding functions, test automation , reflect questions and documenting scrum report |
| Aayush Bhogal | Whitebox test cases and Updated requirements traceability matrix stored in the repository. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Functions decided | **Can start working on black box test cases and then traceability matrix** |
| Black box test cases finalised | **Can be added to traceability matrix now** |
| Traceability matrix | **Can start working on reflect and scrum report** |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Functions finalised | **Group participation** |
| Black box test cases modified | **Team contribution** |
| Traceability matrix completed | **Team work** |
|  |  |
|  |  |
|  |  |
|  |  |

**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**:

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.   
     
     
   In this milestone, We can just write blackbox tests but not whitebox tests as in this milestone we have just created function specifications and not the function itself. Black box test are specifically meant for the case when we don’t know what happen inside the function but we can just expect some output based on the parameters passed and function description. This approach is the only possible for testing code at this early stage in software development.This type of testing is based on the external behaviour and requirments and not on the implemetation of the function. Therefore, using black box test cases for this milestone was the most appropiate. This helps in early validation of software for specific requirements. This type of testing provide easy traceability between test cases and specific requirements.
2. Explain why we need the function-test matrix and why it is important in a large project.  
     
   Ans) One of the major factors that ensures Software Testing Life Cycle is **Requirement Analysis.** It is therefore essential that all the business requirements are understood before deciding on the functions and test data. Once we identify what the project requirements entail, we can test for both the functional and non-functional parts of the requirements. To start creating a traceability matrix, the basic requirements of the application are required and to ensure that it is successfully implemented , we ensure that all the test cases that we create align with the business requirements in some way. The traceability matrix is extremely important in large projects as it ensures the following components :

(i) **Valid Function** -- Ensuring the functions align with business requirements and are not created randomly without any implementational usage. While deciding on each function, one ensures that it fulfills the business requirements in some way and then start work on it, this is checked by the matrix.

(ii)**Aspects** – When we check our test cases and business requirements using traceability matrix, it ensures that all the edge cases are covered, and we do not miss out on any test case that is essential in the working of the application.

(iii)**Check** – Traceability matrix can be used as reference by testers about the tests that they have conducted so that when there is a change or modification that must be performed, new test data can be created keeping in mind all the tests that were previously being passed.

To implement the function-test matrix, all the business requirements are written across the top and tests down the left-side. When we put a ‘x’ in the box aligning the test case and business requirements that means that particular test case aligns with the business requirement leading us closer to the fulfillment of the application. There are two types of traceability – forward traceability is where one looks up a requirement and finds the related tests while reverse traceability is where one looks up a test and sees the associated business requirements with it.

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?

Ans) As the software that we know today got more and more complex, there was a need to develop models that would enable multiple people to work on a project simultaneously without waiting for a particular task to be complete. The competition in the market increased, which called for more agile methods that would increase the efficiency of the work along with reducing the time it would take to complete a task. Earlier, a certain developer could not start work because the design part was not completed which led to redundancies and till that time the business requirements of the project changed. These problems lead to the making of agile models that could facilitate the rapidly changing requirements of the application. Agile methods make managing the simpler as –

1)**No wait time** – The whole time can decide upon the requirements that have to be fulfilled and start working on the project and not wait for a particular part of the project to be complete to start their work.

2)**Time management** – With this model, the entire was kept continually busy and everyone could complete their work on their own time and not have a time constraint in the end after a particular task is complete.

3)**Higher Productivity** – People working in a team at a time use human resources effectively as all the people are contributing to the project and not staying idle at any point of time.

4)**Reduced Delivery Times** – Everyone gets assigned work, which increases the speed at which the tasks are completed and hence leads to reduced delivery times.

5)**Easy accommodation** – Any changes in the requirements can be accommodated easily because they typically only change one part of the software. Even if that software has already been implemented, it is still relatively small compared to the size of the entire project and is not that costly to re-implement if necessary.

Completion of large projects should be carried out using Agile models as they always facilitate the inclusion of everyone in the project and does not require a certain task to be completed fully for others to start execution. In the times of such changing business requirements and fast paced software, agile models are the best approach to take.