# **MILESTONE 3** -- SFT221 SCRUM Report and Reflection

All students are expected to attend the SCRUM meetings and to participate. Failure to do so will result in greatly reduced grades.

**GROUP**: \_\_\_\_\_\_\_\_\_\_A\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1. Heungbin Oh | 5. Hiu Fung Chan |
| 2. Tien Vu To | 6. Trung Kien Phan |
| 3. Arshia Keshavarz Motamedi | 7. Ying Cheung Ellis Fung |
| 4. Aditya Tambe |  |

## 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 added to a new header file and stored in the repository.
* A set of blackbox tests as test documents (in an Excel file) 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.
* **Create and add a C++ testing project to your solution.**
* 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 (in the C++ project) well-designed and documented | 15% |
| Functions implementation (coded in the C project & well documented) | 10% |
| Requirements traceability matrix (complete and added to GitHub) | 10% |
| Git usage (used properly with good structure) | 10% |
| Jira usage (creates issues, tracks progress) | 10% |
| Scrum report & reflections | 20% |
| **Deadline** | 20% deduction for each day you are late |  |

**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** |
| Arshia Keshavarz Motamedi | **writing traceability-matrix** | **-** |
| Heungbin Oh | **Coming up with test cases** | **-** |
| Tien Vu To | **Documenting the test cases** | **-** |
| Aditya Tambe | **Creating the scrum report, assisting to others if required additional help** | **-** |
| Hiu Fung Chan | **black box cases studying for functions** | **-** |
| Trung Kien Phan | **Function Implementation** | **-** |
| Ying Cheung Ellis Fung | **Studying the Shortest Path Algorithm and function design** | **-** |

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 |
| Noting down the traceability of matrix | **It was then distributed amongst each other based on the expertise each member has in.** | **Everyone raised the issue according to task pick by their expertise** |
| Creating the test cases | **It was then distributed amongst each other based on the expertise each member has in.** | **Everyone raised the issue according to task pick by their expertise** |
| Documenting the test cases | **It was then distributed amongst each other based on the expertise each member has in.** | **Everyone raised the issue according to task pick by their expertise** |
| Creating the black box functions | **It was then distributed amongst each other based on the expertise each member has in.** | **Everyone raised the issue according to task pick by their expertise** |
| Implementation of function which are being created | **It was then distributed amongst each other based on the expertise each member has in.** | **Everyone raised the issue according to task pick by their expertise** |
| Analyzing and study the shortest path of algorithm | **It was then distributed amongst each other based on the expertise each member has in.** | **Everyone raised the issue according to task pick by their expertise** |
|  |  |  |

**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 |
| Noting down the traceability of matrix | Task was executed as we mutually agreed and told the responsibilities of each task which was being held by all by themselves. |
| Creating the test cases | Task was executed as we mutually agreed and told the responsibilities of each task which was being held by all by themselves. |
| Documenting the test cases | Task was executed as we mutually agreed and told the responsibilities of each task which was being held by all by themselves. |
| Creating the black box functions | Task was executed as we mutually agreed and told the responsibilities of each task which was being held by all by themselves. |
| Implementation of function which are being created | Task was executed as we mutually agreed and told the responsibilities of each task which was being held by all by themselves. |
| Analyzing and study the shortest path of algorithm | Task was executed as we mutually agreed and told the responsibilities of each task which was being held by all by themselves. |
| Noting down the traceability of matrix | Task was executed as we mutually agreed and told the responsibilities of each task which was being held by all by themselves. |

**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? |
| Arshia Keshavarz Motamedi | **Coming up with the traceability matrix, by going through the documentation provided by professor** | **10hrs** | **Y** |
| Heungbin Oh | **Coming up with the thorough planning of what all need to be tested and what no need to test** | **9hrs** | **Y** |
| Tien Vu To | **Once all the test case were planned, it was time for documenting it and was thus executed by him.** | **8hrs** | **Y** |
| Aditya Tambe | **Was assigned to create the scrum and monitor and help others since the complexity of task was increased.** | **4hrs** | **Y** |
| Hiu Fung Chan | **Coming up the black box test cases and studying thoroughly the functions** | **9hrs** | **Y** |
| Trung Kien Phan | **Was assigned to work on the implementation of the function** | **8hrs** | **Y** |
| Ying Cheung Ellis Fung | **Was assigned to work on the shortest path for algorithm and contributed to working for black box test cases and implementation of it since its complexity was increased as compared to previous** | **9hrs** | **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 |
| Arshia Keshavarz Motamedi | Noting down the traceability of matrix |
| Heungbin Oh | Creating the test cases |
| Tien Vu To | Documenting the test cases |
| Aditya Tambe | Creating the black box functions |
| Hiu Fung Chan | Implementation of function which are being created |
| Trung Kien Phan | Analyzing and study the shortest path of algorithm |
| Ying Cheung Ellis Fung | the shortest path for algorithm and contributed to working for black box test cases and implementation of it |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| As we mutually decided that we will raise the issue and assign to ourselves accordingly | **This made it easier for all of us to focus on the task which was been issued by us and no need to of other to remind what is been assigned to each other** |
| Holding and ensuring to fulfill the commitment | **By doing so, no interference was encountered, as well task executed smoothly since all things went smoothly from test planning to execution to programmers to black box testing creators** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Creating issues | **We selected the topic we have expertise on** |
| Meeting the deadlines | **As per agreement and topic chose by us, it became easy to deliver the topics on time by providing and submitting it on the time discussed and said in the meeting** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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.   
     
     
   We are creating the black box tests and not the white box tests because at this stage we are prioritizing and focusing on the functionality of software and not on the internal factors of our delivery system. Also, we are creating as per the requirements of what we would require in our system to make it run. On the other hand, the white box tests require the functions to be implemented and we can see the source code for creating test cases and test data. However, the function implementation is not completed yet and we will implement the source code of the functions in our upcoming milestones.
2. Explain why we need the function-test matrix and why it is important in a large project.  
     
   Function- test matrix is essential for working on large project because; since we are working on a delivery system which itself is so big with all huge data to be managed so as this will help us to map each and every function which is been created in the project and to its correspondence tests. Additionally, this helps to identify which are tests or which might be the function which won’t be requiring test or test cases by saving time and energy which otherwise we need to invest. Thus, ensuring efficiency and quality improvement of the development of software as well as making it easier for our testers to process their testing in a more efficient manner.
3. 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?

The agile model helps in repeating or iterating the testing throughout the development of the lifecycle of our software. This keeps a heads up for all our group members and keeps them engaged with the work which is equally distributed into small issues created. In this way our programmers and testers can have sprints and thus can finish the work early and before the deadline. This helps in playing safe since we do get server issue, or device issue, by doing so reduce the risk as well as reduce the complexity of the project too. On the other hand, we feel this would be more towards complex as it requires to be in contact with the assigned person, tracking and tracing the updates and thus creating the certain adjustments but on the contrary it provides more of the flexibility and easy to tackle the bug fixation too.