# **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**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_B\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1. Jeet Patel | 4. Jeetkumar Patel |
| 2. Samarth Shah | 5. Yash Shah |
| 3. Sahil khatri | 6. Arthav Patel |

## 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** |
| **Jeet Patel** | **Create c++ test project** |  |
| **Arthav** | **Create excel file blackbox testing** |  |
| **Yash** | **Create function specs** |  |
| **Samarth** | **Blackbox testing code for utilizationscore and validate functions** |  |
| **Sahil** | **Blackbox testing code for assigntruck and getbestroute** |  |
| **Jeetkumar** | **Make tracability 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** | **Uploading on github , delay by sahil** |
| **Reason for delay or block** | **Studentvpn not getting connected** |
| **Impact on Project** | **1 hour delay on commit on github** |
| **Solution or work-around** | **Restart the globalprotect vpn and relogibn using Seneca credentials** |
|  |  |
| **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 |
| Distribution of blackbox testing for individual functions | **Who will write the code for testing of functions specified by yash.** | **Divided tasks for blackbox testing to sahil and Samarth and yash.** |
| Creating c++ project for testing | **Who will create the c++ testing project** | **Jeet assigned the task for creating c++ project and merging all individual test code for functions in one.** |
| Traceability matrix | **Between Arthav and Jeetkumar , who will do the tracability matrix and who will do documenting the test.** | **Jeetkumar will do tracability matrix and arthav will do documentation.** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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? |
| Jeet | **Create c++ project and commit using tortoisegit** | **15 mins** | **Yes** |
| Yash | **Start working on blackbox test data** | **20 mins** | **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 |
| Arthav | Implement function code for functions specified, specific functions assigned will be decided next week. |
| Jeetkumar | Implement function code for functions specified, specific functions assigned will be decided next week. |
| Samarth | Implement function code for functions specified, specific functions assigned will be decided next week. |
| Others | To be discussed on monday |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Clarity on work distribution of each member keeping equal distribution in mind | **Better and equal contribution by all members leaving no room for conflict among members.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Understanding and respecting each other’s time schedules. | **Meeting at a time when everyone agrees and no one remains left.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 were to create only functions specifications and not the function code , so without having the code, we can just assume from the function specifications that what value will be return when input is made. On the other hand, Whitebox testing involves testing the internal structure, code, and logic of the system. It requires knowledge of the system's implementation details and typically involves techniques such as code coverage analysis, unit testing, and structural testing. Whitebox tests

are closely tied to the specific implementation of the system and are typically written by developers to ensure code correctness, identify bugs, and cover edge cases. Writing Whitebox tests would require a deeper understanding of the code implementation, which might not be the immediate priority at this stage. It is more efficient to focus on completing the function specifications, designing the Blackbox tests, and implementing them to verify the desired behavior. Once the system's functionality is established and the code implementation is more stable, whitebox testing can be pursued in subsequent milestones to ensure code coverage, handle edge cases, and perform more detailed checks on the internal workings of the system.

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

A function-test matrix is like a map that helps keep track of which parts of a project have been tested and which ones haven't. It's important for big projects because there are so many different things that need to be checked. Here's why it's useful:

* It makes sure that everything that needs to be tested gets tested properly. It shows where the tests are and helps find any spots that haven't been checked yet.
* It connects the things we need to do with the tests we run, making sure everything is done the right way. This helps people know if the project meets all the requirements.
* It helps organize all the tests in one place, making it easier to keep track of what's been done and what still needs to be worked on.
* It helps everyone involved in the project understand what's being tested and how far along we are. This way, we can all work together better and fix any problems faster.
* It helps us check if new changes to the project mess up things that were working fine before. This saves time and makes sure everything keeps running smoothly.

Overall, the function-test matrix is super important for keeping track of what's being tested, making sure everything works as it should, and helping everyone involved in the project work together smoothly.

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 our project, we're using an agile model to keep everyone busy and avoid waiting around. We split the work into small parts called sprints, so we're always making progress. Our team is versatile, with everyone pitching in on different tasks, ensuring there's always something to do. We can adjust our plans quickly if needed, focusing on the most important tasks first. We use tools like Kanban boards to track progress and make decisions. While it's simple to see what's going on and adapt to changes, managing workload balance and ensuring smooth integration of different parts can be challenging. Overall, agile methods help us stay productive and responsive, but we need to stay on top of coordination and communication to keep everything running smoothly.