# **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.Seyam Chowdhury | 4.Anuraj Singh Osan |
| 2.Basil Tariq | 5. |
| 3.Anton Samofalov | 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** |
| Seyam Chowdhury | **Problem analysis** | **N/A** |
| Basil Tariq | **Review and comment the code** | **N/A** |
| Anton Samofalov | **Test plan and data structure implementation** | **N/A** |
| Anuraj Singh Osan | **Scrum report and reflections** | **N/A** |
|  |  |  |
|  |  |  |
|  |  |  |

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** | **N/A** |
| **Reason for delay or block** | **N/A** |
| **Impact on Project** | **N/A** |
| **Solution or work-around** | **N/A** |
|  |  |
| **Delayed or Blocked Task** | **N/A** |
| **Reason for delay or block** | **N/A** |
| **Impact on Project** | **N/A** |
| **Solution or work-around** | **N/A** |

**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 all required functions for the milestone** | **Decided on specifications and stored in repository** |
| |  | | --- | | Blackbox Testing |  |  | | --- | |  | | |  | | --- | | **Reviewed test case designs** |  |  | | --- | |  | | |  | | --- | | **Reviewed test case designs** |  |  | | --- | |  | |
| |  | | --- | | Traceability Matrix |  |  | | --- | |  | | |  | | --- | | **Discussed mapping test cases to requirements** |  |  | | --- | |  | | |  | | --- | | **Discussed mapping test cases to requirements** |  |  | | --- | |  | |
| |  | | --- | | Visual Studio Setup |  |  | | --- | |  | | |  | | --- | | **Team faced issues with Visual Studio setup for testing** |  |  | | --- | |  | | |  | | --- | | **Agreed on team troubleshooting approach** |  |  | | --- | |  | |
| |  | | --- | | Repository Permissions |  |  | | --- | |  | | |  | | --- | | **Addressed permissions issues for pushing code** |  |  | | --- | |  | | |  | | --- | | **Set up permissions with instructor guidance** | |
|  |  |  |
|  |  |  |

**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 |
| |  | | --- | | **Use Visual Studio for testing** |  |  | | --- | |  | | |  | | --- | | Streamlines testing and allows use of assertions |  |  | | --- | |  | |
| |  | | --- | | **Implement a traceability matrix** |  |  | | --- | |  | | |  | | --- | | Ensures alignment of requirements with tests |  |  | | --- | |  | |
| |  | | --- | | **Start with simpler functions first** |  |  | | --- | |  | | |  | | --- | | Allows for faster progress and feedback |  |  | | --- | |  | |
| |  | | --- | | **Use structured black box test documents** |  |  | | --- | |  | | |  | | --- | | Enables clear tracking and review of test data and results |  |  | | --- | |  | |
| |  | | --- | | **Assign tasks based on expertise** |  |  | | --- | |  | | |  | | --- | | Optimizes team strengths for quicker completion | |
|  |  |
|  |  |

**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? |
| Seyam Chowdhury | **Updated created Jira issues and traceability matrix** | **1 hour** | **yes** |
| Basil Tariq | **Created black box test cases – xcel sheet** | **1.5 hours** | **yes** |
| Anton Samofalov | **Worked on function specifications and**  **Function implementation** | **2 hours** | **yes** |
| Anuraj Singh Osan | **Reflections and the Scrum reports** | **1.25 hours** | **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 |
| Seyam Chowdhury | Traceability matrix and Jira project |
| Basil Tariq | Completing black box testing |
| Anton Samofalov | Completing functions |
| Anuraj Singh Osan | Scrum report creation and reflections |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Defined function specifications | **Team has clear tasks to work on** |
| Blackbox testing approach agreed upon | **Ensures testing consistency and clarity** |
| Traceability matrix in place | **Allow requirements to be tracked against tests** |
| Established troubleshooting strategy | **Minimizes delays caused by technical setup issues** |
| Repository and permissions resolved | **Team can now collaborate without interruptions** |
|  |  |
|  |  |

**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 |
| Clear task assignment | **Reduced ambiguity and improved accountability** |
| Coordination on visual studio | **Established troubleshooting guidelines** |
| Effective discussion on testing | **Led to well-structured and organized black box test approach** |
| Repository setup | **Enabled all team members to access and contribute to the code** |
| Traceability matrix initiation | **Facilitated mapping of the requirements to test cases** |
|  |  |
|  |  |

**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 |
| Visual studio setup issues | **Team agreed to test configurations beforehand next time** |
| Delay in repository access | **Set permissions early to avoid delays in contributions** |
| Confusion over black box design | **Assign a lead to clarify testing structure before meeting** |
|  |  |

**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.   
     
   Blackbox tests can be written because they focus on testing the system's functionality based on its inputs and expected outputs, without requiring knowledge of the internal code structure. At this stage, functional requirements or specifications are available, enabling testers to design test cases that validate whether the program meets its intended behavior. Since blackbox testing does not rely on the implementation details, it can be performed even if the internal code is not accessible or fully developed. In contrast, whitebox tests require understanding the internal code, which may not be feasible or available yet.
2. Explain why we need the function-test matrix and why it is important in a large project.  
     
   The function-test matrix is essential in a large project because it systematically maps functions or features to their corresponding test cases, ensuring comprehensive coverage of all requirements. It helps identify untested functions, reducing the risk of undetected bugs. In a large project, where multiple teams work on various components, the matrix improves coordination by providing a clear view of what has been tested and what remains to be verified. Additionally, it facilitates tracking progress, simplifies test maintenance, and ensures alignment between testing efforts and project goals, which is crucial for managing complexity and delivering a reliable product.

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?

The agile model avoids leaving team members idle by breaking the project into smaller, iterative cycles called sprints, where teams work on deliverable parts of the project concurrently. This allows continuous development, testing, and integration, keeping all members engaged with tasks such as coding, testing, or refining requirements. Agile promotes cross-functional teamwork and adaptive planning, ensuring that progress is incremental and constant. While this keeps the team busy, it can make project management more complex due to the need for frequent coordination, prioritization, and handling evolving requirements. However, it also offers flexibility and quicker feedback, leading to higher-quality outcomes and better alignment with client needs.