# **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**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_7\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1. Janipan Sivaguru | 4. |
| 2. | 5. |
| 3. | 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** |
| **Janipan** | **Everything** |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

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 |
| N/A meeting was not held | **N/A meeting was not held** |  |
| N/A meeting was not held | **N/A meeting was not held** |  |
| N/A meeting was not held | **N/A meeting was not held** |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| N/A meeting was not held | **N/A meeting was not held** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Janipan | **Scrum report** | **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 |
| Janipan | Every task |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| N/A meeting was not held | **N/A meeting was not held** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| N/A meeting was not held | **N/A meeting was not held** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| N/A meeting was not held | **N/A meeting was not held** |
| N/A meeting was not held | **N/A meeting was not held** |
| N/A meeting was not held | **N/A meeting was not held** |
| N/A meeting was not held | **N/A meeting was not held** |
|  |  |
|  |  |
|  |  |

**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.   
     
     
   It may be thought more difficult to analyze an issue and design software without actually writing the code than to write the software itself. The complexities of the problem, potential edge cases, and effective algorithmic solutions are all carefully taken into account during this process. It requires a thorough comprehension of the problem domain to guarantee that the functions that are designed fit in perfectly with the goals of the project. The challenge stems from the requirement for thorough planning and foresight to avert possible problems during the implementation stage. This process may be more difficult than quick coding since it necessitates a thorough understanding of the issue and the capacity to anticipate difficulties.
2. Explain why we need the function-test matrix and why it is important in a large project.  
     
   In a big project, the function-test matrix is crucial for a several number of reasons. It offers a method & structures approach to guarantee thorough test coverage. Every feature or function is mapped against multiple test cases, which specify which tests must be run for every function. This matrix is useful for:

Monitoring Test Coverage: Teams can monitor which features or functions have been tested and which ones still need to be tested by using the matrix. It guarantees a comprehensive analysis of every aspect of the software.

Determining Test Dependencies: Various features or functions may be dependent on one another in a large project. By assisting in the identification of these dependencies, the matrix guarantees that tests are carried out in the appropriate sequence to take interdependencies into account.

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?

Agile project management can be easier or more difficult. simpler due to the iterative process's capacity for adaptability and change response. More complicated because efficient coordination and communication are needed to manage frequent iterations, coordinate tasks, and maintain a continuous feedback loop. Overall, especially in dynamic project environments, the advantages of decreased idle time and flexibility frequently outweigh the difficulties.