# **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**: 4

**Members Present**:

|  |  |
| --- | --- |
| 1. Jo Eric - 137057188 | 4. Peralta Joe - 114751209 |
| 2. Nguyen Huu Linh - 118197227 | 5. Dominguez Daniel - 110835188 |
| 3. Jang Hyeri - 115328221 | 6. Sadat Morsal - 102693215 |

## 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 at end of Lab:**

* Completed SCRUM report and reflections

**Deliverables Due at 23:59 6 Days after Lab:**

* A set of function specifications stored in the repository,
* A set of blackbox tests as test documents with test data for the functions.
* Start writing blackbox test code and store in repository. (at least 1 required)
* Start implementing functions and store in repository. (optional)
* A function-test matrix added to the repository.
* Updated Jira project to show activities and progress.

**Rubric**

|  |  |  |
| --- | --- | --- |
| Individual | Group Participation | 75% |
| Teamwork | 10% |
| SCRUM Report | 15% |
| Group | Function Specs (documented, correct, complete, well-written) | 20% |
| Test documents (well-written, complete, good test data) | 20% |
| Test Code (well-designed, written and documented) | 10% |
| Git Usage (used properly with good structure) | 5% |
| Jira Usage (creates issues, tracks progress) | 10% |
| Meets Deadlines | 10% |
| SCRUM report & reflections | 25% |

**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** |
| Jo, Eric | * **Updated Jira to show activities and progress.** * **Collaborate and helped in completing the scrum report** * **Test Plan 10-13, and Test Plan integration** | **There are no tasks delayed or blocked** |
| Nguyen, Huu Linh | **Test Plan 1-6**  **Completing the reflection** | **There are no tasks delayed or blocked** |
| Jang, Hyeri | **Test Plan 7-9**  **Completing the reflection** | **There are no tasks delayed or blocked** |
| Peralta, Joe | * **Collaborate and helped in completing the scrum report** * **Data Structure Creation** | **There are no tasks delayed or blocked** |
| Dominguez, Daniel | **Test Plan 14-17** | **There are no tasks delayed or blocked** |
|  |  |  |
|  |  |  |

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** | **There are no tasks delayed or blocked** |
| **Reason for delay or block** | **There are no tasks delayed or blocked** |
| **Impact on Project** | **There are no tasks delayed or blocked** |
| **Solution or work-around** | **There are no tasks delayed or blocked** |
|  |  |
| **Delayed or Blocked Task** | **There are no tasks delayed or blocked** |
| **Reason for delay or block** | **There are no tasks delayed or blocked** |
| **Impact on Project** | **There are no tasks delayed or blocked** |
| **Solution or work-around** | **There are no tasks delayed or blocked** |

**Summary of Meeting:**

A summary of the main points discusses in the meeting and the outcomes of the discussions.

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| Task Distribution | **Function specifications and implementation has been assigned to be completed by Joe Peralta. Blackbox test document has been assigned to be completed by Hyeri Jang. Blackbox test code implementation has been assigned to be completed by Nguyen Huu Linh, and Eric Jo. Function matrix has been assigned to be completed by Daniel Dominguez.** | **Each member that was assigned a task has agreed to do and complete their assigned task.** |
| Understanding of the assigned task | **Each member must understand the requirements needed to successfully complete their assigned tasks.** | **Each member has reviewed the necessary requirements to complete their assigned task.** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Collaboration | If some tasks are highly unhandled, the other member can give a hand to complete the tasks. |
| Contribution of every member to build 4 functions | Last time, we missed details and resulted in just two data structures. To save time, we agreed on sharing ideas on creating additional structures. |
| Regular communication about the progress in group chat | We decided to regularly communicate in group chat to keep track of the progress of each member’s task. |
| Detail-oriented approach | If someone found a detail that other members missed, he or she should notify them of it. |
|  |  |
|  |  |
|  |  |

**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? |
| Eric | **Assigned individual tasks to each member** | **10m** | **Yes** |
| Eric | **Assigning a task to multiple people** | **10m** | **Yes** |
| Joe | **Added information about the topic, discussion summary, and outcome in the Summary Meeting section of the scrum report. In addition, he helped in the section called Things that did not go well in this meeting section to include information about time constraints.** | **30m** | **yes** |
| Joe | **Helped in filling out the summary tasks completed, the delayed section in the scrum report, and reflection.** | **40m** | **yes** |
| Hyeri | **Filled out the Major Outcomes of Meeting section** | **10m** | **yes** |
| Daniel | **Set up function-test matrix document, completed reflection question #2** | **20m** | **yes** |
| Linh | **Helped completing the summary of tasks accomplished and delayed section in the Scrum report. Things That Did NOT go Well in This Meeting and Reflection** | **20** | **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 |
| Joe | Function Specification and Implementation |
| Hyeri | Test document for Blackbox tests |
| Linh | Blackbox test code |
| Eric | Test document for Blackbox tests, Blackbox test code |
| Daniel | A function-test matrix |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Tasks/Roles Assigned | **By assigning tasks and roles to each team member, efficiency was increased.** |
| Scrum Report Completed | **The scrum report was prepared based on the results of the scrum meeting.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Assigning individual tasks to each member | **Members who joined the meeting showed their passion to participate in their roles.** |
| Assigning a task to multiple people | **We all agree not to just assign a single task to only one person but multiple members if needed.** |
| Attendance | **In a meeting, everyone is comfortable and solves problems together. The meeting atmosphere was comfortable** |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Time Constraint | **We did not have enough time to collaborate more for the planning of milestone 3. Each of us has a different time schedule which makes it harder to complete each meeting successfully. But mostly the meeting is excellent enough to cover some of the tasks and requirements that needed to be done, especially the ones that are most important.** |
| Understanding deliverables clearly | **In the last milestone, we misunderstood a deliverable, so mark was deducted. For the next time, we should focus on the prof’s words and give questions if we cannot get some elements clearly.** |
| Advancing Towards Optimal Solutions | **Following our productive discussion with the professor, we have identified a few remaining challenges. Currently, our team is dedicated to further enhancing the situation and striving for optimal solutions.** |
|  |  |
|  |  |
|  |  |
|  |  |

**Reflections**:

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.

- The reason we cannot write Whitebox tests is that the member who was assigned to function specification and implementation is different from one who assigned to creating test cases. Other members don’t have enough time to wait until the member completed code implementation and go through the entire code to generate Whitebox tests. Code reading is not needed to generate Blackbox tests, which means the other members can produce test cases as much as possible before the code implementation is completed. Blackbox tests guarantee time efficiency in this milestone.

1. Explain why we need the function-test matrix and why it is important in a large project.  
    - We need a function-test matrix to track function requirements and corresponding test cases, it ensures that the main functionality of the software is tested properly and thoroughly.

- It is important because it provides a clear overview of which test cases cover which functions and/or requirements. This in return helps the testers identify gaps or redundancies in the test coverage and ensures that all the necessary functionality is thoroughly tested. It is also important to keep this document up-to-date as new requirements and/or test cases are identified. It serves as a reference for test execution, enabling testers to track their progress and ensure comprehensive coverage.

- This also allows the project stakeholders to assess the level of testing coverage and make any informed decisions about the testing process if needed.

- Overall, the function-test matrix increases confidence in the quality of the software, by aligning test cases with the desired functions in the form of efficient and systematic testing, especially in large projects with many functions and features.

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 modeling focuses on dividing business requirements and problems into manageable chunks that can be distributed and assigned to other developers or engineers in parallel. It helps in making the project’s life cycle more efficient and well-structured by involving each member continuously in the life cycle. This helps in reducing the time spent completing the whole project. It also introduces collaborative project management where each member shares ideas, implements tasks, and solves problems in a continuous process without waiting for other developers to finish their own tasks.

- It makes managing the project simpler and less complex because each member involved in completing the project contributes, collaborates, and shares ideas continuously without waiting for others to finish their own tasks. The project is much more organized because each member is involved in the process, and no one is slacking.

- Agile emphasizes collaboration and communication among team members. Instead of a sequential handoff approach, team members continuously share ideas, discuss implementation strategies, and provide feedback. This ensures that everyone is actively involved in the project and can contribute their expertise throughout the development process.