# **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**: \_\_\_\_\_\_\_\_\_\_\_\_\_E\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1. Kemal Batu Turgut | 4. Tu Yin Hnit Aung |
| 2. Kusum Acharya | 5. Roy Bryan D. Franck |
| 3. Krish Sanjaybhai Patel | 6. Shovana Shrestha |

## Milestone 3 Tasks

In this milestone you will create issues to design the functions, design all 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 is 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 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** |
| Kemal Batu Turgut | Function specifications & testing |  |
| Krish Sanjaybhai Patel | Function specifications & testing |  |
| Roy Bryan D. Franck | Function implementation |  |
| Tu Yin Hnit Aung | Function specifications & testing |  |
| Kusum Acharya | Requirements traceability matrix, Scrum Table (major outcomes, things went well, did not go well, reflection 2) |  |
| Shovana Shrestha | Scrum Report and reflect questions 1 & 3 |  |
|  |  |  |

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 discussed in the meeting and the outcomes of the discussions.

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| Dividing the tasks | **We equally divided the tasks so that each member can finish their work on time.** | **It promotes clear communication and collaboration among team members.** |
| Coding and Testing | **We discussed about the additional functions & blackbox testing focusing on its importance in ensuring that software meets all the needed requirements.** | **We make sure that our project meets every requirement by outlining the importance of extra functions and blackbox testing, which are prioritized for thorough verification.** |
| Deadlines | **The team discussed the upcoming deadline to ensure timely delivery without compromising quality work.** | **We had a meeting just after our class, so everybody had enough time to finish their tasks.** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Prioritizing coding and testing | Function implementation and blackbox testing are critical parts of this week's milestone, so prioritizing coding and testing first ensures timely delivery of a high-quality product by finding and addressing errors early in the development process. |
| Adoption of Agile Testing Practices | We are using agile model for this project which enhances flexibility ensuring thorough testing and validation throughout the process. |
| Task Allocation | We divided each task among the members, allowing for effective resource use and ensuring that each team member has clear responsibilities, hence increasing productivity and accountability. |
| Deadline Review and Adjustment | When all tasks are completed on schedule, project timeframes can be properly assessed by all the team members for review and can be adjusted as necessary to manage any expectations or changes in advance. |
|  |  |
|  |  |
|  |  |

**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? |
| Kemal Batu Turgut | Function specifications & testing | **2 hours** | **YES** |
| Krish Sanjaybhai Patel | Function specifications & testing | **2 hours** | **YES** |
| Roy Bryan D. Franck | Function implementation | **2 hours** | **YES** |
| Tu Yin Hnit Aung | Function specifications & testing | **2 hours** | **YES** |
| Kusum Acharya | Requirements traceability matrix, Scrum Table (major outcomes, things went well, did not go well, reflection 2) | **2 hours** | **YES** |
| Shovana Shrestha | Scrum Report and reflect questions 1 & 3 | **2 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 |
| Kemal Batu Turgut | Function specifications & testing |
| Krish Sanjaybhai Patel | Function specifications & testing |
| Roy Bryan D. Franck | Function implementation |
| Tu Yin Hnit Aung | Function specifications & testing |
| Kusum Acharya | Requirements traceability matrix, Scrum Table (major outcomes, things went well, did not go well, reflection 2) |
| Shovana Shrestha | Scrum Report and reflect questions 1 & 3 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |

|  |  |
| --- | --- |
| Task Division | The tasks for this week were successfully divided between each member. Each member was given an option to choose the task they would like to try, and every one’s opinion was voiced and taken into consideration before finalizing it. |
| Participation of Members | **The meeting was in person, so all the members were present for discussion about the milestone. This ensured that all team members had the opportunity to contribute to the discussion and decision-making process, fostering a collaborative environment.** |
| Decision Making and Learning | **The decision making was clearer at this meeting and the most important tasks of the milestone this week were highly prioritized. When discussing tasks, we learn something. Each member was free to ask about any concerns they had during the milestone and the team was ready to help. This not only helped the person asking but also the ones answering and listening.** |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |

|  |  |
| --- | --- |
| Problem Solving | There were a few members who had a minor disagreement with one another. Both sides were allowed to voice their side of the story and the disagreement was resolved. |
| Active Participation | **All team members actively participated in the discussion, sharing their ideas, concerns, and suggestions. This ensured that a variety of perspectives were considered when making decisions, leading to more comprehensive and thoughtful outcomes.** |
| Task Allocation | **Every member had the chance to select work according to their interests and ability, and the task distribution procedure was handled without any major difficulties. This improves team members' overall morale and productivity by encouraging a sense of responsibility.** |
| Effective Time Management | **The time of the meeting was effectively managed, preventing unnecessary delays and facilitating productive discussions. This ensures that meetings are about the project and discussions related with it which increases the productivity of the team.** |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |

|  |  |
| --- | --- |
| Communication | While communication was indeed better at this meeting, the team was still lacking the pivotal depth required for smooth execution of the project. Even though all the members were present, there were instances where important discussions were left out, like for example, the timeline of the members completing their tasks because the works correlate with one another. |
| Potential Challenges or Risks Discussion | **There was inadequate communication regarding possible risks or challenges in the meeting. The group did not, however, allot enough time to talk about potential risks or create mitigation strategies. Additionally, it is important that the team emphasizes transparent communication regarding possible risks, ensuring that all members are aware of any possible challenges and ready to address them in a proactive manner.** |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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.   
     
   **ANS:** Here, in this milestone, we focus on writing blackbox tests rather than whitebox tests for several reasons. To begin, blackbox testing allows us to evaluate the software's functionality from an end-user perspective without getting into its internal implementation details. This strategy is beneficial in the early stages of development when the internal structure may still be evolving. By considering the program as a blackbox and evaluating only its inputs and outputs, testers may confirm that it meets the specified requirements and acts as expected without being restricted by the specifics of how it produces these outcomes. This early testing phase, in which we do not know the internal details, helps in identifying any differences between the expected behavior stated in the requirements and the actual behavior of the software. It also enables us to identify bugs in the software's functionality at an early stage, allowing for quick fixes and improvements.

Finally, blackbox testing fosters flexibility and independence in the development process. By testing the program according to its standards and requirements, we may organize different modules and test them separately, allowing teams to work independently on different sections of the program and making integration easier later. Overall, by focusing on blackbox testing during this milestone, we can ensure that the software's functioning is consistent with its purpose and requirements, providing a foundation for future development and testing initiatives.

1. Explain why we need the function-test matrix and why it is important in a large project.  
     
   **ANS:** To provide thorough test coverage across all capabilities of a system or software application, the function-test matrix is an essential tool in software testing. It acts as a link between the features or operations of the software and the appropriate tests that must be run to confirm that they are correct. It is important in a large project for the following reasons:
2. **Comprehensive Test Coverage:** Software in big projects usually has a lot of features or

functions. The function-test matrix facilitates the tracking and organization of the tests that correspond to each function. This guarantees that every feature is tested extensively resulting in fewer openings for possible bugs to go through unnoticed.

1. **Effective Test Planning and Execution:** Test planning becomes more systematic with a function-test matrix. The tests that must be executed for each function or feature can be readily identified by testers. This makes it easier to allocate time and resources efficiently, allowing testers to focus on important aspects and properly prioritize testing efforts.
2. **Traceability and Documentation:** The matrix makes it evident how the requirements or specifications and the accompanying tests relate to one another. This is essential to guarantee that the software performs as intended and to comply with industry regulations. It also helps with maintenance, debugging, and testing for errors by acting as useful documentation for later use.

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?

**ANS:** An agile model, such as the one that we're using in this project, takes a very different approach to project management than traditional life cycle models. Instead of sequential models such as waterfall, which allow team members to wait for their tasks on the project while other team members accomplish their tasks, agile models encourage continuous collaboration and iteration among team members. By using this agile model, teams can avoid extended periods of inactivity by dividing the project into smaller sections and prioritizing the steady delivery of functional software. Whether it's developing, testing, reviewing, or refining requirements, team members are always working on projects. This makes sure that everyone in the team stays productive throughout the course of the project.

However, while the team's constant activity can enhance performance and adaptability to changes, it also presents challenges with project management. Agile projects require careful planning and collaboration to handle dependencies, prioritize activities, and adjust to changing requirements. To guarantee alignment and advancement, team members must also collaborate and communicate often to successfully finish the project on budget and on time. Consequently, in contrast to conventional linear life cycle models, managing the project successfully in this dynamic context might be more difficult, even though the agile model reduces idle time and encourages continuous activity.