# **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**: 6

**Members Present**:

|  |  |
| --- | --- |
| 1. Shwe Yee Lin Aung | 4. Nevan Sargeant |
| 2. Cynthia Fotso | 5. Tsz Wa Wong (Locus) |
| 3. Jackey Zhou | 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 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** |
| **Jackey** | **Reflection Q3, Black Box test cases** |  |
| **Nevan** | **Reflection Q1, Black Box test cases** |  |
| **Locus** | **Unit test cases, Requirements Traceability Matrix** |  |
| **Cynthia** | **Reflection Q2, Black Box test cases** |  |
| **Shwe** | **Scrum Report, Function Specification, Function Implementation, Requirements Traceability Matrix** | **Function Specification and Implementation** |
|  |  |  |
|  |  |  |

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** | **Function Specification and Implementation** |
| **Reason for delay or block** | **The code was unable to run on Shwe’s laptop** |
| **Impact on Project** | **There was a delay** |
| **Solution or work-around** | **Cynthia was able to fix the problem** |
|  |  |
| **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 |
| Time management and Task allocation | **Tasks required to be completed are identified and allocated to each member according to availability. The tasks were allocated based in urgency and importance. We also took note of the limited time that we had to complete this milestone and worked carefully but swiftly.** | **A general allocation of tasks for each member was completed by collaboration and discussion among the group about which person was allocated to which task. We would aim to finish the most or all of the milestione** |
| Task documentation | **To keep a through documentation of task assignments, progress updates and any decisions made during the task allocation process** | **It serves as a reference point and help maintain accountability.** |
| Task Completion | **Discussion is done to have a general summary and confirmation of completed tasks by each group member.** | **Acknowledge the contributions of each team member to the project or initiative.** |
| Collaboration | **Parties involved: The collaboration involved team members working together on the milestone and communicating through teams and Jira** | **As a result of the discussion, the team agreed on clear project milestones, assigned responsibilities to each team member, established regular progress review meetings, and designated communication channels. Additionally, potential risks and mitigation strategies were identified to ensure smoother project execution.** |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Shwe does the scrum report, function specifications and implementations, Traceability Matrix | She wanted to have a hand at coming up with new functions and documenting the process of the collaboration for the milestone. |
| Cynthia did black box test cases and a reflection | She wanted to share her insights on the project and what she had done in reflection. By not considering the internal workings, it ensures that testing is done from the user's perspective, which helps identify issues that users might encounter in real-world scenarios. |
| Nevan did black box test cases and a reflection | He did so to encourage himself to think critically which helped with analysis, synthesis, and evaluation. Black box testing allows testers to work independently of developers. Testers do not need access to the source code, which means they can start testing as soon as the software is available, without waiting for the code to be completed or stable. |
| Jackey did black box test cases and a reflection | By identifying what worked well and what could be improved, he can apply these learnings to future situations, leading to continuous improvement and growth. |
| Locus did function testing and black box test cases, Traceability Matrix | Reflective practices encourage him to articulate their thoughts and experiences effectively. This improved communication skill can facilitate better collaboration, conflict resolution, and relationship-building in future collaboration. |
|  |  |
|  |  |

**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? |
| Locus | **Function testing and unit test cases, Traceability Matrix** | **80 mins** | **yes** |
| Shwe | **Scrum report function specification and implementation, Traceability Matrix** | **80 mins** | **yes** |
| Jackey | **Black box test cases and a reflection** | **60 mins** | **yes** |
| Cynthia | **Black box test cases and a reflection** | **60 mins** | **yes** |
| Nevan | **Black box test cases and a reflection** | **70 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 |
|  | To be assigned in the meeting on teams during the next week |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| The code that was not initially running due to some bugs worked | **The overall process to finish the milestone using the fixed code in the upcoming week’s tasks are going smoothly, black box test cases could be done on the code** |
| Jira updates | **There is a documentation of who does what for each week tracked which makes it easier for the professor to see who made each contributions** |
| Github | **The files completed by each group members are uploaded for easier access by the professors and easier collaboration of the team going forward** |
| Reflections | **Reflection encourages the group members to enhance their self-awareness, communication and collaboration which leads to overall improved performance and productivity within a team** |
| Scrum Report | **Help establish accountability within the team by documenting commitments and progress towards those commitments. Team members are accountable for completing their assigned tasks within the agreed-upon time frame** |
|  |  |
|  |  |

**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 |
| Allocation of tasks | **Group members identified what they were comfortable with doing and set off to do what was required for each task.** |
| Completion of tasks | **Since group members did what they were comfortable with and asked for clarifications during the group call, the progress went smoothly for each deliverable.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 Constraints | **Some of the deliverables in the project took longer time than expected** |
| Fixing bugs | **The code was not running as expected due to some bugs and it was fixed, additionally, code implementation is improved upon by more than one member and hence, it became more suitable as an end deliverable** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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.   
     
   We write the black-box tests instead of the white-box because this allows us to guess what the function tests can possibly do because when using black-box  the only thing that we can really do is make speculations on the functions that is not fully developed. While White-box tests require testers to know the code logic and structure in order for the test to happen. White-box can only also work if the code is developed to nearly complete in order for the test to be used correctly, if the code is not nearly completed then using the white-box testing would be useless. In conclusion, using black-box testing is more easy when it comes to early project testing as it is only being tested from a user point of view.
2. Explain why we need the function-test matrix and why it is important in a large project.

The function-test matrix also known as the requirements traceability matrix is a document used in software testting to map requirements to the corresponding test cases. It enables everyone ensure that all requirements are covered by the tests and it also enables the verification of the software. It's importance is seen in the below points;  
- Requirement coverage: in a large project having numerous requirements, the function-test matrix serves as a roadmap to ensure that all requirements are adequately tested. Ignoring it might lead to the risk of missing critical functionalities which can lead to gaps in test coverage, hence potential issues in the software.  
- TRaceability: It brings out a clear link between each requirement and its associated test cases. it is crucial for quality assurance as it helps to trace the test results back to specific requirements. It facilitates transparency throughout the testing process which enhances good decision making.  
- It also enables to identify dependencies and relationships between requirements and tests which enhances risk mitigation by focusing on areas of highest importance.  
- Change management: with requirements changing throughout the projects lifecycle, it is crucial to manage changes effectively to ensure that testing remains aligned with the actual specifications. the function-test matrix will allow the teams to update the test cases accordingly.  
- In large projects, the function-test matrix serves as a central document that provides a shared understanding of the testing objectives and progress. This will enhance effective documentation and communication between teams fostering collaboration and coordination.

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?

I would say that the Agile model solves the problem of team members being idle while waiting for specific project milestones by changing the approach to project management. Agile model will break the project into smaller, more solvable problems instead of focusing on deliverables and their due dates. Agile model allows team members to assign themselves additional work. The agile model method allows work to be divided into dependent pieces, and larger tasks can be broken down into smaller problems. By using agile method team members select a manageable piece of work to avoid idle time. It allows flexibility and continuous adjustment. Agile model promotes cooperation and active communication which helps to keep everyone informed and invested in the project. Therefore makes the project simpler.