# **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**: \_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1.Duc Phu Nguyen | 4. Karanbeer chanana |
| 2.Anthony Korepanov | 5.harmanpreet Singh |
| 3. Jashandeep Singh | 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** |
| **Duc Phu Nguyen** | Test plan and create data structure have completed | **Completed** |
| **Anthony K** | **Created Native Unit Test** |  |
| **All Members** | **Did Reflection questions** | **completed** |
| **Karanbeer chanana** | **Reflection in scrum report and testing of new functions** | **completed** |
| **harmanpreet** | **Wrote the function prototype done with the tester function** | **completed** |
|  |  |  |
|  |  |  |

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 |
| Discussion MS3 | **How to create 4 new functions prototype and 2 functions provided and create test case** | **Each member create one function prototype and create mins 4 test case** |
| doubts | Create traceability matrix and test case template | Each member fill test case into template |
| meeting | **Testing of all the functions correctly done** | **All of us successfully tested functions and understood testing** |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Devision | Work was successfully divided among group |
| Analysing source code | Source code was understood by everyone with better point of view to test |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Duc Phu Nguyen | **Create function vailddestination and min 4 test case** | **10mins** | **Completed** |
| Duc Phu Nguyen | Write scrum-report MS r Reflections question 1 | 15mins | Completed |
| Duc Phu Nguyen | **Create form traceability matrix and test case template** | **20mins** | **Completed** |
| Anthony K | Create function getInput() and min 4 test case, added to scrum report q2, create Native unit test | **1h** | **Completed** |
| Jashandeep Singh | **Created function checkBoxSize(), made 5 test cases for the prototype, did question 3 for reflection, completed test cases for a a function from mapping.c(distance)** | **1.30 hr** | **Completed** |
| Karanbeer chanana | **Creating function checkWeight and made five test cases along with reflection question** | **1 hr** | **completed** |
| Harmanpreet singh | **Created function checkspaceoftruck and wrote 4 test cases and done with tester function** | **1hr** | **completed** |

**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 |
| Anthony K | Develop getInput() implementation |
| Duc Phu Nguyen | Implementation function ValidDestination |
| Jashandeep | Testing/ debugging and developing checkBoxSize function |
| Karanbeer chanana | Develop checkweight() implementation and testing |
| Harmanpreet | Develop checkspaceoftruck and implementing the tests |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Organized group | **More group members showed up in person, created good organization** |
| Git usage | **Everyone used git correctly** |
| Better communication | **Everybody communicated through jira and share their thoughts.** |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Testing of code | **Due to good communication teammates help each other testing the code** |
| Scrum report | **Everybody did their part in scrum report** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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.**

There are several reasons why we use black box testing. Firstly, it's easier to develop because it relies solely on input and expected output, without requiring knowledge of the internal code. Secondly, even if the coding is incomplete, black box testing can still be executed to ensure that the functions meet the requirements. On the other hand, white box testing, which focuses on code function and structure analysis, testing tools. However, these may not be available during the development stage. Therefore, black box testing is suitable a practical choice for validating software functionality.

1. **Explain why we need the function-test matrix and why it is important in a large project.**

The function test matrix is important in a large project because it gives us an overview of which test cases apply for each function under test. It allows the developer who is scanning the document to quickly determine which tests are being applied to which function. It also shows the extent of how much each function is being tested, as we can see immediately the amount of test each function is receiving.

Many teams or team members work on different system components are common in large projects. The function-test matrix supports better communication between developers, testers, and users by providing as a shared point of view. By giving a clear picture of the requirements and the current state of the testing, it makes coordination and collaboration more efficient.

The matrix directly assists quality assurance efforts by systematically mapping functions to their tests. It assists in locating areas of the application that have not been tested, guaranteeing that test efforts are thorough and in accordance with the project's goals. This systematic testing strategy is crucial for spotting problems early in the tests

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 approach has played a role, in shaping software development by introducing an adaptive method to manage projects. By dividing tasks into sprints teams can focus on progress while being able to adjust to changing needs. Collaboration and communication are aspects of Agile ensuring that all team members are involved and contributing to the project’s success, which facilitates decision making and issue resolution.

Furthermore, Agile stresses the importance of goals and priorities to keep the team aligned with the project’s objectives throughout development. This clarity minimizes confusion and keeps everyone focused on delivering value to customers. Additionally Agile promotes improvement, by allowing teams to reflect and adjust after each sprint. This iterative process enables teams to incorporate feedback address issues. Enhance their processes for efficiency and outcomes over time.

To conclude, Agile methodology does not avoid team member inactivity by encouraging involvement and teamwork but also empowers teams to produce top notch results in a dynamic environment that is constantly evolving. By adopting principles teams can confidently tackle complexity. Succeed in their projects. The Agile methodology has demonstrated its effectiveness, in software development. Its popularity has been on the rise in the industry, for some time.