# **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. Song Nhat Nguyen | 4. Prince Ghumaan |
| 1. Nguyen Dang Khoa Huynh | 5. Benson Liu (Can not contact) |
| 1. Mohamed Mohamed | 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** |
| **Song Nhat Nguyen** | **Function specs, Testing excel, code (blackbox testing), Traceability Matrix and reflection 1.** | **N/A** |
| **Nguyen Dang Khoa Huynh** | **Scrum Report, check all the files to meet the requirements, updated Testing Plan, assign tasks on Jira and upload files on GitHub repo.** | **N/A** |
| **Mohamed Mohamed** | **Reflection 2.** | **N/A** |
| **Prince Ghumaan** | **Reflection 3.** | **N/A** |
|  |  |  |
|  |  |  |
|  |  |  |

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** | **N/A** |
| **Reason for delay or block** | **N/A** |
| **Impact on Project** | **N/A** |
| **Solution or work-around** | **N/A** |
|  |  |
| **Delayed or Blocked Task** | **N/A** |
| **Reason for delay or block** | **N/A** |
| **Impact on Project** | **N/A** |
| **Solution or work-around** | **N/A** |

**Summary of Meeting:**

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

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| Function Spectification | **Function spec doc written by Song Nhat Nguyen according to new functions written on header file (Truck.h) last week. Nguyen Dang Khoa will check it again to make sure the structure functions correctly and meet the requirements.** | **Function Spec doc completed** |
| SCRUM Report | **Complete it after the files and reflection questions are completed** | **SCRUM Report is done** |
| Testing Functions | **Testing functions were examined as a form of black box testing.** | **Evaluating Functions Testing was examined as a form of black box testing. Functions have been written but not yet run (Execution scheduled for MS4).** |
| Black Box Testing (Exel File) | **Testing completed** | **This file finished and uploaded to GitHub repo.** |
| Jira | **Assigning Tasks on Jira** | **Completed** |
| Git | **Git updated for each branch** | **Completed** |
| Traceability Matrix | **Based on test and fill in Traceability Matrix** | **Completed** |
| Testing Plan | **Updated questions 5, 10, 11, 12, 14, 15, 16, 17.** |  |

**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 |
| Testing Decision | We have resolved to conduct experiments on map generation and position identification throughout the evaluation phase. |
| Testing Functions Implementation | The black box will encompass fundamental functionalities such as basic mapping, findTruck,Shipment, addPackage, capacity, pathCalculation, addPackage and countPackage. |
| Function Specs | Added function specs on existing header file(Truck.h), and documentation completed |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Song Nhat Nguyen | **Created set of functions and update current header file(Truck.h). Created Black Box test code in excel file. Doing reflection 1. Create test files. Fill in Traceability Matrix** | **4 hours** | **Completed** |
| Nguyen Dang Khoa Huynh | **Update testing plan, Checking all files, Scrum Report. Assign tasks on Jira and upload file on GitHub** | **3.5 hours** | **Completed** |
| Mohamed Mohamed | **Reflection 2** | **1 hour** | **Completed** |
| Prince Ghumaan | **Reflection 3** | **1 hour** | **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 |
| Nguyen Dang Khoa Huynh | Manage tasks on Jira |
| ALL | Meeting in Monday Class and arrange a meeting on Wednesday |
| ALL | Reflection questions, SCRUM report |
| ALL | Test Execution including White box testing codes and MS3 Blackbox testing |
| ALL | Debugging |
| ALL | Function implementation |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| N/A | **N/A** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

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

**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.   
     
   Let's back to the definition of blackbox, so blackbox is a technique that you will test function without knowing anything inside. What you need to know is only about the function and a parameter and which expected result that the function will return to user.

Then you will compare the function with business or user requirement by making a lot of test cases. Basically Blackbox is when you play as a user and try to figure as many as possible test cases that user might use it in reality. This is why we don't need to know about the declaration inside black box. It is more like a game that 1 + 1 = 2 and you have to create a case for the calculation to make sure that 1 + 1 = 2 and the one doing the maintenance will take your document about test cases and working with whitebox later.

1. Explain why we need the function-test matrix and why it is important in a large project.  
     
   We need the function test matrix for reasons such as testing and ensuring that the program works with many different test cases. It also helps stop new bugs from happening when changing code by retesting other functionalities ensuring they still work after code updates. In large projects it is very useful because of the amount of information you have to keep track of, but with this you can ensure everything always works as intended and any errors will be efficiently dealt with.
2. 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 model cuts down on downtime by breaking the project into smaller tasks that can be done at the same time. In Milestone 3, for example, some team members work on writing function specifications while others write blackbox tests, start implementation, or make changes to the requirements traceability matrix. Because of this, everyone can play their part, and the process doesn't get stuck in a certain order like it does in traditional models like the Waterfall model.  
  
To keep track of tasks and progress on our project, we use Jira and GitHub. This helps the agile process. For instance, we used Jira to give out tasks like writing test cases for certain functions and GitHub to code together and keep track of commits. With these tools, team members can start working on new tasks as soon as they finish the ones they are already doing. This cuts down on downtime.  
  
This way of working keeps the team interested, but it makes management more difficult. Coordinating tasks, keeping communication open, and avoiding doing the same work twice can be hard. Also, putting all the finished parts together needs a lot of planning. But Jira and a function-test matrix make it easier to keep track of dependencies and make sure everything is lined up.  
  
Overall, agile makes project management more fluid and difficult, but it also encourages teamwork and output, which makes the benefits greater than the problems when managed correctly.