# **MILESTONE 4** -- 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. |
| 2.Anthony Korepanov | 5. |
| 3.karanbeer chanana | 6. |

## Milestone 4 Tasks

**Deliverables due 4 days after your lab day:**

* Finish implementing/coding the functions.
* Finish implementing/coding blackbox tests. Store in repo, executed, results in Jira (and on corresponding test documents, and debugged.
* A set of whitebox 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.
* Whitebox tests implemented (in the C++ testing project), stored in repository, executed, results in Jira and on corresponding test documents, and debugged (at least 1 SET is required).
* Updated requirements traceability matrix stored in the repository.
* Completed hook file (for EACH team member) for test automation stored in the repository.
* Completed scrum report including reflection questions answered.

**Rubric:**

|  |  |  |
| --- | --- | --- |
| **Individual** | Group participation (includes GitHub commits and Jira usage) | 80% |
| Teamwork | 20% |
| **Group** | Implemented functions and main (well-designed, and documented) | 10% |
| Finish coding blackbox code (well-designed, written, and documented) | 5% |
| Whitebox test case document (well written, complete, good test data) | 10% |
| Whitebox test code (well designed and documented) | 20% |
| Updated requirements traceability matrix | 5% |
| Test execution (performed, results recorded, issues created) | 10% |
| Debugging (bugs fixed, documented, Jira updated) | 5% |
| Hook files | 10% |
| Git usage (used properly with good structure) | 5% |
| Jira usage (creates issues, tracks progress) | 10% |
| Scrum report & reflections | 10% |
| **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** | **4 function prototypes** |  |
| **Duc Phu Nguyen** | **Updating traceability matrix** |  |
| **Duc Phu Nguyen** | **Create test case template** |  |
| **Anthony K** | **4 Blackbox tests in unit test** |  |
| **Anthony K** | **Create code backbone, Link functions together** |  |
| **Anthony K**  **Karanbeer chanana** | **Implement getInput()**  **Implement checkweight , 4 blackbox test in unit tests, 8 whitebox tests for 2 fundtions and scrum report with hookife** |  |

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 |
| MS4 | **Implement 4 functions** |  |
|  | **Write white box test for 4 function** |  |
| Organize issue | **Determine areas of confusion and make time-management for group** | **Attempt to set meeting** |
| Git usage | **How to use git correctly and commiting changes** | **Better communication on git** |
| Unit testing | **Blackbox testing of unit test** | **Everyone was able to do blackbox tests as unit test** |
| Scrum report | **Content of scrum report** | **Easy to maintain scrum report** |
|  |  |  |

**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 |
| Re-work function parameters | Check functions do not need elaborate pointers, they only need the value they are testing |
| Unit testing | All the blackbox function required to be unit tested for better results |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 | **Implement function validdestination** | **2hrs** | **complete** |
| Duc Phu Nguyen | **Write whitebox test case** | **15mins** | **complete** |
| Duc Phu Nguyen | **Write reflection** | **30mins** | **complete** |
| Karanbeer chanana | **Blackbox test cases** | **45 min** | **complete** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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 calculate route functions for main() method, link mapping.c functionality to our functions. |
| Karanbeer chanana | Developing of functions and testing them |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Determined correct cases for unit-test | **Cleared up confusion between which test cases to implement first in unit test.** |
| White box test cases determined | **All the white cases were devided among members correctly** |
| Correct git and jira usage | **Git was correctly used and jira by members to commit and communicate** |
| Coding of functions | **Code for functions were made correctly by members** |
|  |  |
|  |  |
|  |  |

**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 |
| Unit test was fixed | **Group member analyzed issue with unit test** |
| Better communication among members | **All the members communicated well in meeting** |
| Coding was done for functions | **Code for all the function was made correctly** |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Issues regarding time management | **Some parts get started late and may not have enough time to complete if they are complex** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Reflections**:

Answer the following questions using your own words. Make sure that each answer comprises a minimum of 100 words.

1. After you run your blackbox and whitebox tests you are asked to record the results in both the original test document as well as in Jira. Explain why it is a good idea to record the results in both places.
2. Why did we wait until the fourth milestone to write the whitebox tests?  
     
   There are several reasons to wait until now to write the Whitebox tests. Firstly, we need to focus on establishing fundamentals in the early stages rather than implementing functions. This approach allows developers to gain a clear understanding of the logical structure before go into internal testing. Secondly, Whitebox tests involve developers having a comprehensive understanding of data structures, internal logic before steps inside functions, wait until this stage can help developer gain insight system complexity. Addition, waiting until later stages to write Whitebox tests ensures that they remain relevant and effective, capturing any changes or optimizations made to the code along the way.
3. Pick one of the functions you created and list its name. For this function did you produce more blackbox or whitebox tests? Explain why your answer (more blackbox or more whitebox) happens for most functions.

The "checkWeight" function has more blackbox tests than whitebox tests. There are six blackbox tests and four whitebox tests

Blackbox tests try different inputs to see if the function gives the right output, without caring about how the function works inside. They're like trying out different things to see if the function can handle them.

Whitebox tests, on the other hand, look inside the function to check specific situations based on how it's coded. You don't need as many whitebox tests because if you've checked that the function's logic works for some key weights, it's likely to work for others too.

So, there are usually more blackbox tests because they check the function from different angles, while whitebox tests focus on specific code logic.

1. Explain the purpose of the automation hook for GIT and explain how it can improve the quality of the software in the project.

The automation hook in GIT is a BASH script that gets executed when the developer calls a certain GIT command. In our case, we are making a script to get executed when we call GIT PUSH. The hook is called pre-push. The automation hook allows us to call our unit test before pushing the local project to the repository. This is an extra layer of functionality that makes sure the code adheres to the correct functionality assigned in the unit test before being committed to the repository on GitHub. This improves the quality of the software because it stops possible bugs from being committed to the repository, which causes problems when bugs get nested deeper and deeper in the repository after multiple commits. It saves time and energy for the testers and developers by detecting potential bugs early before making more code.