# **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**: 2

**Members Present**:

|  |  |
| --- | --- |
| 1. Joon | 4. Heqing |
| 2. Doris | 5. |
| 3. Xiaopeng | 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** |
| **Joon** | **Function design with blackbox test cases, function description and unit test** |  |
| **Doris** | **Function design with blackbox test cases, function description** |  |
| **Xiaopeng** | **Function design with blackbox test cases, function description** |  |
| **Heqing** | **Function design with blackbox test cases, function description** |  |
| **All** | **Scrum report** |  |
|  |  |  |
|  |  |  |

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 |
| Tasks | **Distribute tasks evenly** | **Decided everyone will continue to work on their functions** |
| Meeting | **Whether this MS requires us to meet in person** | **Decided on communicating via Teams** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Everyone will work on their own function implementation as well as test cases | Working on the functionality in parallel will increase productivity and efficiency |
| We will study each others functions | While we are working on our own functions, it is also important that each member understands the architecture of the project, thus it is required to study each others implementations for peer review |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| All | **Setting up unit tester file to work for this MS** | **10m** | **y** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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 |
| Joon | Implement function, design white box test cases as well as unit test them, update prepush file and store in hooks files, debug and update Jira |
| Doris | Implement function, design white box test cases for implemented function plus one given function, blackbox unit test, reflection Q1 |
| Xiaopeng | Implement function, design white box test cases for implemented function plus one given function, black box unit test, reflection Q2 Q3 |
| Heqing | Implement function, design white box test cases for implemented function plus one given function, black box unit test, reflection Q4 |
| All | Update traceability matrix |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Learned about each others functions | **Each member understands the architecture of the project well** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Scrum meeting swiftly distributed tasks | **We are used to distributing our tasks and determining what is a fair workload** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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. 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.  
   The original test document serves as a comprehensive record of all the tests that were conducted. When checking for certain test for certain functions, it’s easily found in the testing document. We also record all the bugs we found in the process in Jira as it provides a centralized platform for tracking bugs. Everyone is informed of all the bugs that were found during testing, and team leader can assign engineers to fix certain bugs. The status of each bug and progress of fixing bugs is also shown graphically to the team which improved the visibility. Overall. By recording test results in Jira, we establish clear traceability between the tests conducted and any related issues or bugs that are identified.
2. Why did we wait until the fourth milestone to write the whitebox tests?  
   For software project, we have to split the whole project to several parts. Different parts are focus on different aspects of the project. We have to define requirement, design test plan, implement features, and testing. Until the fourth milestone, it is right time to write the whitebox tests that typically involves examining the internal structure, design, and implementation details of the software. And, Writing whitebox tests at this step allows the team to verify the interactions between different modules and ensure the integrity of the integrated system.
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 function I'll pick is “int validDestination(struct map\* maps, int row, char col)”

For this function, I would typically produce more blackbox tests. It tells us about the inputs and outputs of the function, and not its internal implementation, blackbox testing is more appropriate. In blackbox testing, I test the function based on its specification without looking at the internal code. I design test cases based on the expected behavior of the function given certain inputs, without considering how the function achieves that behavior. Therefore, in most cases, blackbox testing is preferred as it allows us to test the function based on its specified behavior and requirements, without needing to know the details of its internal implementation.

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

Git automation hooks are scripts that run automatically when certain important actions occur.

1. Code review and code quality inspection: By integrating code quality inspection tools (such as linters, static code analysis tools, etc.) in pre-commit hooks, automatically ensure that all submitted code follows the project's coding standards and style guide . This helps reduce errors and inconsistencies in your code, thereby improving code quality.

2. Automated testing: Unit tests or integration tests can be automatically run before committing or pushing code to the warehouse (for example, using pre-commit or pre-push hooks). This

ensures that only code that passes all tests is committed or pushed, helping to detect and fix bugs early.

3. Ensure code security: Automatically check for potential security holes or unsafe code practices by integrating security scanning tools into corresponding hooks (such as pre-commit or pre-push). This helps identify and mitigate security risks in advance.