# **Milestone 4 Scrum Report**

All students are expected to attend the scrum meetings and to participate. Failure to do so will result in greatly reduced grades.

**GROUP**: \_\_06\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1. Sean Li | 4. Dhanuth Fernando |
| 2.Sahan Vimukthi | 5. |
| 3. Dineth Damishka | 6. |

## Milestone 4 Tasks

* 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 in the repository, ensuring it shows both passed (green) and failed (red) tests.
* 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) | 10% |
| Updated requirements traceability matrix | 10% |
| Test execution (performed, results recorded, issues created) | 5% |
| Debugging (bugs fixed, documented, Jira updated) | 5% |
| Hook files | 15% |
| Git usage (used properly with good structure) | 5% |
| Jira usage (creates issues, tracks progress) | 15% |
| 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** |
| **Dineth** | Completed implementation of shipment functions (percentage calculations, validation) | None |
| **Sahan** | Developed blackbox tests for shipment and truck functions and documented test results | Delay in finalizing test data for some cases |
| **Dhanuth** | Created whitebox test cases in Excel, including at least 4 sets per function | None |
| **Sean** | Updated requirements traceability matrix; prepared hook files for test automation | Minor delay due to Git merge conflicts |
|  |  |  |
|  |  |  |
|  |  |  |

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** | Finalizing test data |
| **Reason for delay or block** | Waiting for clarifications on shipment specs |
| **Impact on Project** | Testing incomplete for some edge cases |
| **Solution or work-around** | Team discussed and resolved specs during meeting |
|  |  |
| **Delayed or Blocked Task** | Git merge conflicts |
| **Reason for delay or block** | Overlapping commits caused conflicts |
| **Impact on Project** | Delayed integration of hook files and docs |
| **Solution or work-around** | Used communication and manual merges to fix |

**Summary of Meeting:**

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

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| Shipment functions | Reviewed current implementation, ensured alignment with header files and specs | Confirmed shipment\_functions.c correctness |
| Testing approach | Agreed on scope of blackbox and whitebox tests, clarified test case requirements | Assigned members to specific test creation |
| Git and Jira usage | Discussed merge conflicts and proper branching strategy | Decided to adopt stricter commit policies |
| Task priorities | Prioritized finishing tests and updating traceability matrix | All members agreed to focus on testing next week |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Use Manhattan distance for routing | Simplifies route distance calculations for milestone 4 |
| implement at least 6 functions tested | To meet milestone test coverage requirements |
| Each member responsible for hook file | Ensures automation tests can run smoothly |
| Stricter Git commit messages and branch names | Reduce merge conflicts and improve traceability |
|  |  |
|  |  |
|  |  |

**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? |
| Sean | Reviewed shipment\_functions.c implementation | **45 min** | **Yes** |
| Sahan | Worked on blackbox test cases for shipment funcs | **90 min** | **Mostly** |
| Dhanuth | Created whitebox test cases in Excel | **2 hrs** | **Yes** |
| Dineth | Updated traceability matrix and merged Git changes | **1 hr** | **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 |
| |  |  | | --- | --- | | Dineth |  | | |  |  | | --- | --- | |  | Complete any remaining shipment function code | |
| |  |  | | --- | --- | | Dhanuth |  | | |  |  | | --- | --- | |  | Finalize blackbox test cases and run test automation | |
| |  |  | | --- | --- | | Sahan |  | | |  |  | | --- | --- | |  | Finish whitebox test documentation and Excel files | |
| |  |  | | --- | --- | | Sean |  | | |  |  | | --- | --- | |  | Fix any remaining merge conflicts and finalize hook files | |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Shipment functions code finalized and reviewed | Foundation ready for thorough testing |
| Testing responsibilities clearly assigned | Improved accountability and progress tracking |
| Git conflicts addressed with new commit policy | Reduced risk of future delays |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| |  |  | | --- | --- | | Clear communication of task status | Everyone shared updates openly, avoiding surprises | | |  |  | | --- | --- | | Clear communication of task status | Everyone shared updates openly, avoiding surprises | |
| |  |  | | --- | --- | | Collaboration on Git issues | Quick resolution of conflicts through teamwork | | |  |  | | --- | --- | | Collaboration on Git issues | Quick resolution of conflicts through teamwork | |
| |  |  | | --- | --- | | Defined testing approach | Agreement on test scope helped focus efforts | | |  |  | | --- | --- | | Defined testing approach | Agreement on test scope helped focus efforts | |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| |  |  | | --- | --- | | Delays in test data finalization |  | | |  |  | | --- | --- | |  | Waiting for specs slowed progress; improve early clarifications | |
| |  |  | | --- | --- | | Some incomplete whitebox tests |  | | |  |  | | --- | --- | |  | Lack of detailed route info caused blocking; propose early info gathering | |
| |  |  | | --- | --- | | Merge conflicts on Git |  | | |  |  | | --- | --- | |  | Caused delays; will enforce branch discipline and code reviews | |
|  |  |
|  |  |
|  |  |
|  |  |

**Reflections**:

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

1. How did analyzing the internal logic and structure of the code help you design effective white-box test cases?  
     
   When I looked closely at how the code was written and how the functions worked internally, it really helped me think about all the different situations the program might face. For example, knowing how the weight and volume were calculated made me realize I should test cases where the truck is almost full or completely empty. Also, seeing how destination validation worked helped me come up with tests for invalid or out-of-bound locations. This way, I could create test cases that cover many possible paths the code could take, making sure the program behaves correctly even in unusual situations. It made the testing much more thorough and meaningful.
2. How did using automated unit testing tools simplify or enhance your testing process? Reflect on the advantages and potential limitations of automation compared to manual testing methods.  
     
   Using automated testing tools was a big time-saver. Instead of testing everything by hand over and over, I could just run the tests quickly and get instant results. This helped me find mistakes faster and make sure new changes didn’t break old features. Automation made the process more reliable because the tests were run the same way every time, which is hard to do manually. However, automation also needs well-written tests to be effective. Sometimes, it’s tricky to automate tests for complex scenarios, and manual testing might still be necessary. Overall, automation made testing easier and more efficient for our team.
3. How did you document and communicate the bugs you identified? Reflect on the importance of clear and detailed bug reports in ensuring that issues are effectively resolved by the development team.  
     
   Whenever I found a bug, I made sure to write down everything clearly in our bug tracking system. I explained exactly what was wrong, how to reproduce the problem step by step, and what the expected result should have been. Sometimes I included screenshots or error messages to help others understand better. This clear communication is really important because it helps developers fix issues quickly without having to ask a lot of questions. Good bug reports also help keep track of what was fixed and what still needs attention. In a team project, clear bug documentation makes the whole process smoother and faster.