# **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 6**

**Members Present**:

1. Sahan Gallage
2. Ileperuma Gunarathna
3. Dhanuth Hennedige
4. Sean Li

## 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** |
|  |  | **Parts of milestone 3 not completed in an acceptable state, upon review** |
|  |  | **Timely completion of milestone 4** |
|  | **Start of milestone 4** |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

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** | **Milestone 4** |
| **Reason for delay or block** | **Parts of milestone 3 not completed in an acceptable state, upon review** |
| **Impact on Project** | **Unable to move forward in a timely manner** |
| **Solution or work-around** | **To be discussed with professor** |
|  |  |
| **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 |
| What needs to be done | **Milestone 4 deliverables, according to the rubric, and what needs to be on GitHub and Jira** | **Introduction to milestone 4** |
| Issues with milestone 3 upon review | **Issues discovered in milestone 3’s submission** | **To be corrected for milestone 4** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Distributing tasks for milestone 4 | Each member should have an estimated equal workload for milestone 4 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 | **Issues with milestone 3, what needs to be done for milestone 4** | **1 hour** | **Partially** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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 |
| Reflect and review | Look into the existing state of the project’s issues, review what has been done and what still needs to be done to move the project forward in a timely manner, and address the overall progress of the project. Discuss with professor if necessary. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Review of milestone 3 | **Understand milestone 3’s issues as a prerequisite to attempt milestone 4** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Topics described above | **No issues with communication and team integrity** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| N/A |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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?  
     
   In order to write white box tests, you obviously have to understand what’s inside the code. You can watch out for special cases and the particularly logical flow of the functions in order to unravel and test them. Sometimes you may be able to tease out the internalities of a code through black box testing, but only by looking into the code itself can you write white box tests.
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.

Automated testing allows you to focus on the testing, with the scripts taking care of the tests. The advantage is that it takes care of safeguarding submission to repository for you in a cinch. The limitations is that there are some stuff which the testing framework may not be able to do (like taking user inputs or checking or catching unexpected crash/infinite loop) without complex configuration, and that’s where careful manual testing comes in.

1. 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.  
     
   Documenting bugs allow you to easily show and describe issues to the whole team, and saves its log for review and future reference. Having a clear and detailed bug report is just as important as having a well-written program, giving the developers a clear picture of how to identify, replicate, investigate, and ultimately solve the issue, cleanly and quickly.