# **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**: **2**

**Members Present**:

|  |  |
| --- | --- |
| 1.Luca Novello | 4.Eric Yakimoff |
| 2.Philip Ayomide Tijani | 5.Tyler Kay |
| 3.Karishma Singh Mahender | 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 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) | 10% |
| Updated requirements traceability matrix | 10% |
| 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) | 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** |
| Eric | Blackbox test code, function implementations, hook File | **n/a** |
| Luca | Whitebox test code, Whitebox test cases, hook File | **n/a** |
| Karishma | Traceability matrix, scrum report, hookFile | **n/a** |
| Philip | whitebox test cases, scrum report, hookFile | **n/a** |
| Tyler | whitebox test cases, Traceability matrix, hookFile | **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** |  |
| **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 |
| Dividing Work | Discussed the tasks that needed to be completed and the once that needed to be revised. | Updated Jira and scrum report accordingly. |
| Whitebox Tests | Review of the code on hand | Whitebox tests added to repo |
| Blackbox Test code | Review of the function prototype and implementation planning | Blackbox tests updated |
| Test Documents | Requirement analysis | Updated Traceability matrix |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Task distribution | Tasks were dispersed based on comfort with the skillsets needed as well as an emphasis on splitting the work equally. |
| Blackbox Test code | Function Implementation was discussed along with the test casesfor revision |
| WhiteBox Test code | Code was analyzed to create test methods. |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Eric | Task distribution, Update Blackbox test cases, whitebox test code implementation | 10 min | some |
| Luca | Task distribution, whitebox test code, whitebox test cases | 10 min | some |
| Karishma | Task distribution, requirement analysis, code review | 10 min | some |
| Philip | Task distribution, whitebox test cases, code review | 10 min | some |
| Tyler | Task distribution, requirement analysis, code review | 10 min | some |
|  |  |  |  |
|  |  |  |  |

**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 |
| Eric | **Review milestone deliverables, distribute tasks, scrum report, reflections** |
| Luca | **Review milestone deliverables, distribute tasks, scrum report, reflections** |
| Karishma | **Review milestone deliverables, distribute tasks, scrum report, reflections** |
| Philip | **Review milestone deliverables, distribute tasks, scrum report, reflections** |
| Tyler | **Review milestone deliverables, distribute tasks, scrum report, reflections** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Preliminary prioritization of tasks | Allows members to specialize and get comfortable with the work they prefer doing |
| Preliminary development of whitebox test code | Provided a base to write test methods. |
| Traceability matrix update | Clear idea of business requirements for function implementation |
| Blackbox test case review | Provided a base to work off when further structuring Blackbox test code |
|  |  |
|  |  |
|  |  |

**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 | Meeting was productive in establishing a starting point for the week’s tasks |
| Communication | Clear and honest discourse around the parts of the project we were all comfortable with ensured that work would be completed |
| Git | Allows us to share work easier and show our contributions |
| Jira | Allows us to communicate tasks and responsibilities easier |
|  |  |
|  |  |
|  |  |

**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 | **n/a** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Reflections**:

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

1. Why did we wait until the fourth milestone to write the whitebox tests?
2. Whitebox testing is typically conducted after a significant portion of the code has been written and the project structure is well-defined. In this project, we deferred writing whitebox tests until the fourth milestone to ensure that the primary functionalities and core modules were fully implemented and stable. This timing allowed us to gain a deep understanding of the code’s internal workings, which is essential for creating effective whitebox tests. By this milestone, we had already developed and executed blackbox tests, which validated the external behavior of the functions and helped identify and fix major issues. This set the stage for whitebox testing, which focuses on examining the internal logic, paths, and conditions of the code, ensuring comprehensive coverage and detecting any hidden bugs that blackbox tests might overlook. Conducting whitebox tests prematurely, while the code is still evolving, could result in constant test revisions and maintenance, leading to inefficiencies and potential confusion. Therefore, waiting until the fourth milestone ensured that our whitebox tests were robust, targeted, and less prone to frequent changes.
3. How does the Agile methodology ensure that all team members are consistently engaged throughout the software development process, avoiding downtime due to dependencies on others? Provide an example to illustrate your point.

A. The Agile methodology keeps everyone engaged and productive by encouraging regular progress and communication. Agile practices like daily stand-up meetings, sprint planning, and backlog grooming help ensure all team members know the current project status and their tasks. These practices also help identify and resolve dependencies early. For instance, if a developer needs a teammate to finish a feature, the daily stand-up can bring this up. The team can then decide to work together, reassign tasks, or adjust the project plan to keep things moving. Agile's approach of making small, regular improvements means team members often work on different parts of the project at the same time, which helps prevent delays. By keeping the task list well-organized and regularly updated, Agile ensures everyone has clear tasks to work on, reducing downtime and improving teamwork and efficiency.

1. What is a shell script and how are we going to utilize a hook script in this project?

A. A shell script is a text file containing a sequence of commands for a Unix-based operating system's shell to execute. These scripts automate repetitive tasks, manage system operations, and streamline complex processes. In this project, we utilize a hook script as a form of a shell script to automate the testing process before any code is pushed to the repository. The hook script, specifically named pre-push, is placed in the .git/hooks directory. When a team member attempts to push changes to the remote repository, the script automatically runs the suite of tests defined in our project. If any test fails, the push is aborted, preventing potentially faulty code from being integrated. This ensures that only code that passes all tests is pushed, maintaining code quality and stability. Each team member must configure this script on their local machine, customizing paths to the test runner and test DLLs based on their environment. This automation step is crucial for maintaining high standards and reducing manual errors, ensuring that the repository remains reliable and free from integration issues.