# **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**: *\_\_\_\_D\_\_\_\_*

**Members Present**:

|  |  |
| --- | --- |
| 1. Manas Gandotra | 4. Gurmehak Kaur Uppal |
| 2. Ajaypartap Singh Maan | 5. |
| 3. Arshnoor Kaur | 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** |
| **Manas Gandotra** | **Whitebox test data, traceability matrix, hook file** | **No** |
| **Gurmehak Kaur Uppal** | **Completed Blackbox and function implementations** | **No** |
| **Arshnoor Kaur** | **Finished coding blackbox test code** | **No** |
| **Ajaypartap Singh Maan** | **Implemented functions and main code** | **No** |
|  |  |  |
|  |  |  |
|  |  |  |

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 |
| Finish implementing/coding the functions | **Discussed the progress and any issues faced during implementation** | **All functions are implemented and documented** |
| |  | | --- | | Finish coding blackbox tests and documentation |  |  | | --- | |  | | **Reviewed the blackbox test cases, their results, and corresponding documentation** | **Blackbox tests are coded, documented, stored in the repository, and executed** |
| Whitebox tests and documentation | **Discussed the whitebox test cases, their implementation, and corresponding documentation** | **Whitebox test cases are documented, implemented, and stored in the repository** |
| |  | | --- | | Traceability Matrix |  |  | | --- | |  | | **Updated the matrix with new requirements and test cases** | **Matrix is up to date and stored in the repository** |
| |  | | --- | | Hook files |  |  | | --- | |  | | **Discussed the automation of test cases for each team member** | **Hook files completed and stored in the repository** |
| Test execution | **Discussed the execution of tests and recording of results** | **Tests executed, results recorded in Jira and on test documents** |
| Debugging | **Addressed bugs found during testing and their resolution** | **Bugs fixed, documented, and Jira updated** |

**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 |
| All deliverables should be ready 6 hours prior to the deadline | The submission of deliverables prior to the deadline ensures there is enough time for error checking and helping each other with remaining tasks. |
| Any possible delays in submission or inability to join the meeting should be reported in advance | Effective communication between group members is crucial. Reporting tasks in progress on Jira gives time for other team members to come up with a solution or work together on the problem as soon as possible. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| ARSHNOOR KAUR | **Overviewed the milestone 4 pdf thoroughly, reflection discussed** | **30 MIN** | **Yes** |
| GURMEHAK KAUR UPPAL | **Overviewed the milestone 4 pdf thoroughly, deliverables discussed** | **30 MIN** | **Yes** |
| MANAS GANDOTRA | **Overviewed the milestone 4 pdf thoroughly, Scrum Report discussed** | **30 MIN** | **Yes** |
| AJAYPARTAP SINGH MAAN | **Overviewed the milestone 4 pdf thoroughly, Jira management discussed** | **30 MIN** | **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 |
| Ajaypartap Singh Maan | Create integration test document (well-written, complete, good test data) + Complete scrum report & reflections |
| Arshnoor Kaur | Finish coding integration tests (well-designed, written, and documented) + Test execution (performed, results recorded, issues created) |
| Manas Gandotra | Implement integration tests (well-designed, and documented) + Debugging (bugs fixed, documented, Jira updated) |
| Gurmehak Kaur Uppal | Update requirements traceability matrix + Whitebox test code (well designed and documented) |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Milestone 4 towards completion | |  | | --- | | **The deliverables for MS4 are on track to be successfully submitted.** |  |  | | --- | |  | |
| Milestone 4 tasks assigned to each member | **Milestone 4 will be completed before the deadline with all deliverables documented and tested.** |
| The project requirements understood in detail | **Each team member is aware of their assigned tasks and the project requirements for smooth execution and timely submission.** |
|  |  |
|  |  |
|  |  |
|  |  |

**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 Report** |  |  | | --- | |  | | **Everyone participated efficiently and answered reflection questions thoroughly.** |
| Jira Task Assignment | **Members chose their roles effectively and collaborated well on task progress.** |
| Blackbox Testing | **The blackbox test cases were thoroughly reviewed, and the results were accurately documented and executed.** |
| Whitebox Testing | |  | | --- | | **The test cases were well-documented and implemented with good test data, leading to effective verification of functions.** |  |  | | --- | |  | |
| Traceability Matrix | **The matrix was updated promptly and accurately, reflecting the current requirements and test cases.** |
| Team Collaboration | **Team members worked together cohesively, helping each other complete tasks and address any issues promptly.** |
|  |  |

**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. **Why did we wait until the fourth milestone to write the whitebox tests?**

A: We waited until the fourth milestone to write the whitebox tests to ensure that all functions were implemented, stable, and thoroughly understood. Initially, we focused on blackbox testing, which allowed us to verify the external behaviour of our functions and identify major issues from the user’s perspective. This phase provided a comprehensive view of the project, enabling us to address external issues and ensure that the code met its expected behaviour. Whitebox testing involves examining the internal structure of the code, which requires a stable and well-documented codebase. By this stage, significant changes and updates to the internal code had been made, reducing the risk of redundancy and rework. Prioritizing blackbox tests first ensured that major bugs were already addressed, making the code reliable and minimizing potential issues during whitebox testing. This sequential approach, where blackbox testing is followed by whitebox testing, ensures a better understanding of the code, enhances test coverage, and helps in identifying intricate internal logic issues effectively.

1. **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: Agile methodology ensures continuous engagement of team members through iterative development and frequent communication. In Agile, work is divided into small, manageable increments known as sprints, which typically last one to four weeks. This approach encourages regular check-ins through daily stand-ups, where team members discuss their progress, plans, and any blockers they face. These stand-ups promote transparency and help in early identification of dependencies, allowing the team to address them promptly.

For example, in our project, every milestone comprised distinct activities that were delegated to various team members, like developing test cases, implementing functions, and updating documentation. We made sure that nobody was standing about aimlessly waiting for someone else to finish their portion of the project by segmenting it into smaller jobs and keeping Jira updated on our progress. The team may rearrange work priorities or work together to eliminate the bottleneck if a barrier was found, such as a reliance on a function that hadn't been created yet, guaranteeing ongoing productivity.

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

A: Programming created for an operating system's shell, often known as the command line interpreter, is called a shell script. It is a set of instructions that the shell is to follow in order to automate tedious jobs and streamline intricate command sequences. The versatility of shell scripts in software development lies in their ability to handle files, manage system activities, and run applications.

In this project, we are utilizing hook scripts, which are specific types of shell scripts, to automate parts of our workflow. A hook script is triggered by certain actions in a version control system like Git. For example, a pre-commit hook script can be used to run tests or linters before changes are committed to the repository. This ensures code quality and prevents errors from being introduced into the main codebase. By using hook scripts, we can enforce consistent coding standards, run automated tests, and integrate continuous integration practices seamlessly, enhancing the overall efficiency and reliability of our development process.