# **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**: \_\_\_\_B\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1. Isabela Jorge Bulla | 4. Devarsh Patel |
| 2. Ketia Teta | 5. |
| 3. Abdullah Al Mahfuz | 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** |
| **Isabela** | **Scrum Report** |  |
| **Ketia** | **White box test cases** |  |
| **Devarsh** | **Traceability** |  |
| **Abdullah** | **Implementation of black box test cases** |  |
| **ALL** | **Reflections** |  |
|  |  |  |
|  |  |  |

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 |
| What tasks should be done by each team member | **Assignment of tasks.** | **Clear understanding of what needs to be done.** |
| Where to upload files in the GitHub repository | **Where should each file be uploaded.** | **Clear understanding of where to upload the files.** |
| Discussion about hook files | **Since the professor specified that we shouldn’t do these for MS4, we removed it from the Jira tasks.** | **Removal of hook file tasks from Jira.** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Keep the same “tasks” moving forward | Each team member is going to keep working on the same “topic”, for example Isabela is doing the implementation of the code, Ketia is creating the test cases, Abdullah is implementing the test cases and Devarsh is doing the documentation. |
| Remove hook files from deliverables | Remove hook files’ tasks from the Jira board. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Isabela Jorge Bulla | **Scrum report** | **10min** | **YES** |
| ALL | **Definitions of tasks** | **20min** | **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 |
| ALL | Agreed to divide tasks after the tasks have been discussed and understood in the class next week. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Task definitions | **Clear understanding of project and what needs to be done.** |
| Removal of hook files from deliverable | **Removal of hook files from Jira board.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Discussion of tasks | **Everyone participated in the main discussion.** |
| Timing of the meeting | **Meeting in person after class.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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?

White box testing we require to look into the code to create test cases according to it. So, by the fourth milestone, the code is more stable and complete. This allows us to write white box tests with a comprehensive understanding, ensuring that tests are accurate and meaningful. Also, black box testing is done in 3rd milestone as it focuses on verifying the functionality of the software without looking at the internal code structure. This helps identify major issues and ensure that the software meets the requirements. White box tests require detailed knowledge of internal logic, data structure, and algorithms. So, implementation details are clear and well-documented, making it easier to perform white box testing. Black box testing does not cover all the paths so white box testing ensures that all the paths including edge cases are tested properly. In summary, waiting until the fourth milestone to write white box tests ensures that the code is stable, functionality is verified, and implementation details are clear.

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.

Agile methodology breaks down the project into small and manageable parts which ensures that every member has defined tasks to work on. Regular meetings allow team members to share updates, discuss problems, and get a meaningful solution keeping everyone on the same page. The agile method also gives the freedom to reschedule the expected due dates if any problem occurs. Regular sessions ensure that the backlog is up-to-date providing a steady stream of work for the team. Agile is more useful than the traditional method as it is flexible and reduces downtime. For example, we as a group read all the instructions of the upcoming milestone before the class. Then generates the tasks as an issue in Jira. After the class, each member picks tasks based on their expertise and interest. Timely meetings and discussions on Jira help us keep track of everything. In the end, everybody checks the combined work and push it to GitHub. This whole process is a part of agile methodology.

1. What is a shell script and how are we going to utilize a hook script in this project?  
     
   A shell script is a text file that contains a series of commands for the operating system to execute. It's normally used for simple, small programs that don't require advanced logic. In this milestone, the shell script will automate the process of running all unit tests and generating a message indicating whether all tests passed or if any failed. The shell script is used in a hook file. This means that every time someone attempts to push new code to the GitHub repository, all tests are run, and the push is aborted if any test fails. This ensures the program's integrity and prevents defective code from being pushed to the GitHub repository.