# **Milestone 6 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**: **Group C**

**Members Present**:

|  |  |
| --- | --- |
| 1. Chia-Ming Cheng | 4. Peter Bryson |
| 2. Md Arafat Koyes | 5. |
| 3. Md Asif Karim | 6. |

## Milestone 6 Tasks

This is the final milestone where you will run the acceptance tests and fix any remaining bugs found. In addition, you will produce a testing report which lists all the tests conducted, the results and whether the bugs were fixed, and the final test passed. You will also review the test matrix to ensure every test has been performed and passed. You can change the colour of the test in the matrix to show it was run and passed. At the end, all tests in the matrix should have been passed.

The final test report can be tabular like this:

|  |  |  |  |
| --- | --- | --- | --- |
| Function/acceptance/requirement | Test Run | Bugs Fixed | Passed |
| Distance | TF001 | Did not handle negative coordinates | 🗹 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Deliverables due 4 days after your lab day:**

* Final testing report listing tests conducted, bugs fixed, and the final tests passed.
* Execute acceptance tests (results in Jira), and debug.
* Updated requirements traceability matrix 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** | Complete solution code running and executing successfully | 15% |
| Test execution (performed, results recorded, issues created) | 10% |
| Updated requirements traceability matrix | 5% |
| Final test report | 30% |
| Debugging (bugs fixed, documented, Jira updated) | 5% |
| Git usage (used properly with good structure) | 5% |
| Jira usage (creates issues, tracks progress) | 15% |
| Scrum report & reflections | 15% |
| **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** |
| Chia-Ming Cheng | Group meeting invitation and assign tasks, updated GitHub and Jira, wrote integration test document, wrote scrum report, proofread, and submission |  |
| Md Arafat Koyes | Wrote white box test code |  |
| Md Asif Karim | Completed hook file |  |
| Peter Bryson | Wrote integration tests code, acceptance tests implemented |  |

**Summary of Meeting:**

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

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| Task arrangement | Every member knows their job | On time |
|  |  |  |
|  |  |  |

**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 |
| Assign appropriate tasks to each member | To ensure all team members can understand the project |
|  |  |
|  |  |

**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? |
| Chia-Ming Cheng | MS 6 Group meeting invitation and assign tasks | 30 mins | Yes |
| Chia-Ming Cheng | MS 6 Jira task and GitHub update | 30 mins | Yes |
| Chia-Ming Cheng | Writing scrum report | 30 mins | Yes |
| Chia-Ming Cheng | MS 6 Proofread, review and submission | 1 hour | Yes |
| Md Arafat Koyes | Execute acceptance tests and debug | 1 hour | Yes |
| Md Asif Karim | Writing final testing report | 1 hour | Yes |
| Peter Bryson | Updated requirements traceability matrix | 1 hour | 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 |
| N/A |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Every member knows their job | The project is progressing perfectly |
|  |  |
|  |  |

**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 |
| Task arrangement | Every member is good at communicate |
|  |  |
|  |  |

**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. Although we wrote a report on the testing that shows which tests were run and passed or failed, we also updated the traceability matrix. What are the advantages of updating the traceability matrix in addition to writing the test report?

A:

Traceability: The traceability matrix directly links requirements to their respective test cases. By updating it, you ensure that every requirement has been covered by test cases and that changes to requirements are reflected in testing coverage.

Coverage Verification: You can verify that all requirements are adequately tested, and any missed requirements are identified early.

Consistency: The traceability matrix provides a clear and consistent view of which requirements have been tested and how, complementing the detailed information in the test report.

Impact Analysis: When requirements change, the traceability matrix helps assess the impact on testing. It shows which test cases need to be updated or which new tests need to be created.

Documentation of Changes: It documents the relationship between requirements and test cases, making it easier to manage and track changes throughout the project lifecycle.

1. Teamwork on a project like this is vital to its success. How well did your team work together? If you worked well, what contributed to its success? If it did not work well, what contributed to the problems?

A:

Regular Updates: The team had frequent check-ins and updates, ensuring everyone was informed about project status, changes, and issues.

Effective Channels: Utilized effective communication tools and channels (e.g., Slack, Microsoft Teams) to facilitate discussions and information sharing.

Open Feedback: Encouraged an environment where team members could provide and receive constructive feedback, which helped in refining processes and solving problems.

Support and Help: Team members were supportive of each other, offering help when needed and working collaboratively to overcome challenges.

Coordination: Coordinated tasks and dependencies well, ensuring that various aspects of the project were synchronized and moving forward.

Addressing Issues: Any conflicts or disagreements were addressed promptly and constructively, preventing them from escalating and affecting the project.

1. In every milestone you were asked what worked and did not work along the way. Were you able to incorporate what you learned to improve your team’s performance on the next milestone? Did your team learn from their mistakes and improve? If so, why? If not, why?

A:

Feedback Mechanisms: Implemented effective mechanisms for capturing feedback and lessons learned from each milestone (e.g., retrospectives, post-milestone reviews).

Action Plans: Developed and followed actionable plans based on feedback to address identified issues and capitalize on what worked well.

Adjustments: Made specific adjustments to processes, tools, or roles based on lessons learned, leading to improved efficiency and effectiveness in subsequent milestones.

Knowledge Sharing: Encouraged knowledge sharing and discussions about what worked and didn’t work, fostering a culture of continuous learning.

Training and Development: Provided additional training or resources if necessary to address skill gaps or knowledge deficiencies identified during milestones.

Process Improvements: Refined project management practices, communication strategies, and team coordination methods based on past experiences.

1. Did you end up testing the code to the point where you were convinced it worked correctly? Were there any tests that had not pass at the end? If so, what was the impact of this on the project?

A:

Coverage: Ensure that all functional and non-functional requirements were covered by test cases. This includes unit tests, integration tests, system tests, and user acceptance tests.

Test Cases: Verify that test cases were executed for all critical features, edge cases, and potential failure points.

Pass Rate: Analyze the percentage of test cases that passed versus those that failed. A high pass rate generally indicates confidence in the correctness of the code.

Bug Tracking: Review any bugs or issues identified during testing and ensure they were addressed or mitigated appropriately.