# **Milestone 2 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**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_8\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |
| --- |
| Kate De Leon |
| Jarod Jian Kang Hery Chen |
| Ronak Jung Rayamajhi |
| Carson Ji |
| Kemono Onomek |

## Milestone 2 Tasks

Some of the software for the project has already been written for you and is available on Blackboard. You must use this in your project and every team should add it to the source code for their repository. Anything in the main function is simply for demonstration purposes and can be replaced. The software you are being given has not been tested and you will need to test it.

You need to study the problem and the code provided for you and then:

* Add any new data structures you will require This will require a thorough analysis of the problem and the existing software. This should be done by creating a new header file in the directory where the rest of the source code has been placed. You do not want to go back and modify it later if you can avoid it as it will slow the project.
* Create a test plan for the project by replacing the text in the supplied test plan template with your test plan.

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

* An analysis of the problem (no written artifacts produced).
* A series of data structures created as header files and **stored in the repository**.
* A test plan 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** | Data structures (complete, correct, and well-designed, updated in the project, and added to the repository) | 25% |
| Test plan (complete, well-written) | 25% |
| Git usage (used properly with good structure) | 10% |
| Jira usage (creates issues, tracks progress) | 20% |
| Scrum report & reflections | 20% |
| **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 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** |
| Kate De Leon | * Created a data structure for the header file. * Assisted in creating the test plan. * Participated in scrum meetings. | * None |
| Jarod Jian Kang Hery Chen | * Created test plan and assisted in making test strategy. * Participated in scrum meetings. | * None |
| Ronak Jung Rayamajhi | * Update the GitHub repository with all the new changes. * Scheduled task to all the members in Jira. * Participated in scrum meetings and assisted in test plan. | * None |
| Carson Ji | * Created test plan. * Participated in scrum meetings. | * None |
| Kemono Onomek | * Worked in reflection in scrum meeting. * Assisted in creating data structure. * Assisted in test plan creation. | * None |

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 discussed in the meeting and the outcomes of the discussions.

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| Test Plan | About completing the test plan. | Test plan completed |
| Header file | About making a new header file | Header file made |
| Jira | Schedule task properly | Done |
| GitHub | Update the repository | Done |
| Scrum report | About completing the report | Scrum completed |

**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 Priority | All the members were assigned with equal amount of work. |
| Data Structure creation | Discussed and determined the most optimal data structure for the header file. |
| Test plan | Resolved several issues regarding the test plan after multiple discussions. |

**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 cannot be completed, the student should indicate why this was not possible.

|  |  |  |  |
| --- | --- | --- | --- |
| Member | Task Attempted | Time Spent | Complete? |
| Everyone | Test plan | 1 hour | Yes |
| Everyone | Scrum report | 30 minutes | Yes |
| Everyone | Data Structure for new header file | 40 minutes | Yes |
| Everyone | Jira task assignments and GitHub repository update | 20 minutes | Yes |
| Everyone | Analyzing problems regarding the project | 30 minutes | 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 |
| Ronak Jung Rayamajhi | Jira management, GitHub repository management |
| Everyone | Scrum report and reflection |
| Everyone | Function specs |
| Everyone | Test the functions |
| Everyone | Attend the group meeting on July 10, Wednesday at 8 pm |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Test plan | Successfully created the test plan and test strategy. |
| Header file | Successfully created a new header file. |
| Scrum report | Completed the report |
|  |  |

**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 |
| Participation | Everyone attended and actively discussed throughout the meeting |
| Test plan | Planned the test plan creation collaboratively with contributions from the entire team. |
| Scrum | Everyone worked together. |

**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 |
| None | None |
|  |  |

**Reflection Questions:**

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

1. In this milestone you were asked to design the data structure for the project. Print the data structure below then explain each item.
2. #ifndef FINDER\_H
3. #define FINDER\_H
4. *// Structure representing a truck*
5. struct Truck {
6. int truck\_ID; *// Unique identifier for each truck*
7. float volume\_limit; *// Maximum volume the truck can carry*
8. float weight\_limit; *// Maximum weight the truck can carry*
9. int shipments; *// Number of shipments the truck can handle*
10. };
11. *// Structure representing a building*
12. struct Building {
13. int building\_ID; *// Unique identifier for the building*
14. float x; *// X-coordinate of the building's location*
15. float y; *// Y-coordinate of the building's location*
16. int is\_obstacle; *// Flag indicating an obstacle (1 if true, 0 if false)*
17. };
18. *// Structure representing the path request*
19. struct PathRequest {
20. int start\_building\_ID; *// Unique identifier for the starting building*
21. int destination\_building\_ID; *// Unique identifier for the destination building*
22. };
23. #endif

This is the data structure for the project. All the items are explained with the help of comments.

1. Describe the process you used to analyze and understand the existing software code.

We organized a meeting to discuss the project, where everyone shared their understanding of the code and provided ideas and suggestions for the testing process. This collaborative effort helped us gather more information and deepen our knowledge about the project, resolving any doubts among the team. We thoroughly examined the header file and configuration files to grasp the implementation details and pinpoint potential areas for improvement or issues. Our analysis and discussions enabled us to gain a better understanding of the code at hand.

1. What aspects did you consider when creating the test plan? What were the milestones you identified in the test plan?

When we made the test plan, we thought about what parts of the software needed checking and how best to do it. We planned out which tests would cover all the things the software is supposed to do and how it should perform. The test plan was created with several key considerations to ensure comprehensive coverage and effective testing of the delivery algorithm. These aspects include:

1. **Understanding Requirements:**
   * A thorough analysis of project requirements, functional details, and relevant documents to identify the main features, functions, and expected behaviors that need to be tested.
2. **Test Strategy:**
   * Different types of testing, such as system tests, user acceptance tests, documentation tests, and risk-based tests, to ensure the system meets all functional and non-functional requirements.
3. **Environment Requirements:**
   * Specifications for the testing environment, including hardware and software requirements, to ensure consistent and reliable testing conditions.
4. **Execution Strategy:**
   * Defined entry and exit criteria to determine when the testing can start and when it is considered complete, along with severity levels to categorize defects.
5. **Test Design and Overview Process:**
   * Steps for creating and reviewing test cases, including the creation of a traceability matrix to ensure all requirements are covered.
6. **Control Procedures:**
   * Regular reviews, bug review meetings, and a formal change request process to manage and track testing progress and issues.
7. **Functions to Be Tested:**
   * Specific functions such as allocation shipment, capacity calculation, shortest path calculation, and message generation to ensure they perform as expected.
8. **Resources and Responsibilities:**
   * Allocation of resources and responsibilities among team members to ensure efficient execution of the test plan.
9. **Risk Management:**
   * Identification and mitigation of potential risks related to schedule, technical aspects, management, personnel, and requirements.

**Milestones Identified in the Test Plan**

The test plan outlines several key milestones to track progress and ensure timely completion of testing activities:

1. **Test Case Development:**
   * Creating test cases based on identified requirements and scenarios, ensuring they cover all positive, negative, limit, and unusual situations.
2. **Test Environment Setup:**
   * Establishing the necessary test environment, including hardware, software, and simulated data, to support effective testing.
3. **Initial Testing Phase:**
   * Executing initial test cases and documenting results, focusing on critical functions and initial performance tests.
4. **Regular Reviews and Bug Review Meetings:**
   * Conducting regular reviews to evaluate progress and address any issues or gaps in the testing process.
5. **User Acceptance Testing:**
   * Verifying that the system meets user expectations and performs correctly in real-world scenarios.
6. **Documentation and Reporting:**
   * Generating and reviewing test reports, documenting defects, and communicating findings to stakeholders.
7. **Final Testing Phase:**
   * Completing all test cases, ensuring all critical and high-severity defects are resolved, and preparing for final system validation.
8. **Approval and Sign-off:**
   * Obtaining approvals from key stakeholders on the test plan and related documents before proceeding with the final stages of testing.

These milestones help ensure that the testing process is organized properly.