# **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**: **Group C**

**Members Present**:

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

## 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** |
| Chia-Ming Cheng | Set up GitHub repository and Jira project |  |
| Md Arafat Koyes | Wrote scrum report |  |
| Md Asif Karim | Wrote group contract |  |
| Peter Bryson | Meeting invitation |  |

**Summary of Meeting:**

A summary of the main points discussed 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 cannot be completed, the student should indicate why this was not possible.

|  |  |  |  |
| --- | --- | --- | --- |
| Member | Task Attempted | Time Spent | Complete? |
| Chia-Ming Cheng | Add the source code to repository and assign tasks | 30 mins | Yes |
| Chia-Ming Cheng | MS 2 Group Meeting | 30 mins | Yes |
| Chia-Ming Cheng | MS 2 Jira Task Update | 30 mins | Yes |
| Chia-Ming Cheng | MS 2 Proofread and Submission | 30 mins | Yes |
| Md Arafat Koyes | Writing Test Plan | 2 hours | Yes |
| Md Asif Karim | Writing Scrum Report Reflection | 30 mins | Yes |
| Peter Bryson | Data Structures Creating | 2 hours | Yes |
| Whole Group | MS 2 Analysis Problem | 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 |
| Chia-Ming Cheng | Review and update GitHub and Jira, write scrum report |
| Md Arafat Koyes | Write blackbox tests documents |
| Md Asif Karim | Create and add a C++ testing project, write blackbox test code, at least 1 |
| Peter Bryson | Write function specifications |
|  |  |
|  |  |

**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 |  |
|  |  |
|  |  |

**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.

A:

We came up with 3 data structures for our software, one for the delivery, another for the Routes and the third for the Truck Capacities.

**Data Structure for Delivery:**

struct Delivery { int delivery\_id; char destination[100]; float weight; int priority };

This data structure consists of the following members

* **delivery\_id:** This represents a unique integer identifier for each delivery and helps in distinguishing each delivery from the other.
* **destination:** A string holding the address where the delivery is to be made which ensures that delivery reach the correct location.
* **weight:** A float representing the weight of the delivery package in kilograms. This is crucial for logistics and helps determine the load a vehicle can carry as well as if it can take the load in the first place.
* **priority:** An integer indicating the priority level of the deliveries. Priority levels can be high (1), medium (2), or low (3), which helps in determining the order in which the deliveries would be handled.

**Data Structure for Routes:**

struct Route { int route\_id; char start\_point[100]; char end\_point[100]; float distance; };

* **route\_id:** A unique identifier for each route and allows for easy referencing and management of routes.
* **start\_point:** A string representing the starting point of the route and represents where the delivery journey begins.
* **end\_point:** A string representing the destination point of the route and represents where the delivery journey ends.
* **distance:** A float representing the distance of the route in kilometers. This helps in planning and optimizing delivery routes.

**Data Structure for Truck Capacities**

struct TruckCapacity { int truck\_id; float max\_load; char truck\_type[50]; };

* **truck\_id:** This represents a unique identifier for each truck which helps in distinguishing between different trucks.
* **max\_load:** A float representing the maximum load capacity of the truck in kilograms and ensures that trucks are not overloaded.
* **truck\_type:** A string indicating the type of the truck (e.g., small, medium, large). This can help in assigning the right truck for the right delivery based on load requirements.

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

A:

In order to analyze and understand the existing software, we carried out the following steps:

1. **Understanding the Project Requirements**

This involved two important steps of understanding the project description and deliverables. The project involves a delivery system with trucks following specific routes. The map is a 25x25 grid with buildings and open spaces, and deliveries need to be made to specified addresses.

1. **Initial Review of the Code Files**

We had a review of the three provided source codes which are main.c, mapping.c and mapping.h. The main.c contains the main function and serves as the entry point for the program. It is used to demonstrate the functionality of the project by calling functions from the other files. The mapping.c file contains the implementation of functions related to mapping, such as creating and managing maps, calculating routes, or handling spatial data. The mapping.h header file declares the functions and data structures used in mapping.c. It provides the interface for other files to interact with the mapping functionalities.

1. **Detailed Analysis and understanding of the Algorithms**
   1. **Pathfinding Algorithms**: Understanding the pathfinding algorithms implemented in the code, such as the A\* algorithm mentioned in the project instructions.
   2. **Euclidean Distance Calculation**: Review of how the program calculates distances between points on the grid.
2. **Integration with New Data Structures**

After following the above processes of analyzing and understand the existing software and code. we properly commented and documented the code, integrating our new data structures, and ensuring compatibility and consistency across the project.

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

A:

It is important to create a test plan before start working on the project in software development. Planning is the essential part before starting a project. We can make sure by planning that we can ensure the quality which our client exactly and which can work under the condition client wants and work under heavy load also. There are some key aspects we consider during creating our test plan.

1. Testing: We defined which functions, features, and performance we have to test and which is not. Because we will be given a tight schedule for finishing the project if we check everything, we will not able to finish the project without bug and error.

2. Test Environments: We have to test hardware, software and other configurations to create a test environment.

3. Test Risk: We have to identify the risks by testing the projects and have to build a strategy to outcome it.

Milestones in the plan:

1. Testing Approval: The plan of testing must be reviewed by the client and make sure the projects meet all requirement.

2. Test Execution: After meeting with client the members need to start testing.

3. Review: The members have to check that coverage, defect trends when testing is ongoing.

4. Complication: The members have to finish all planed tests and start retesting for failed test cases.

5. Report: After all the test complete, we have to make a final test report.

6. Improvement: If there are any mistakes happen, we will take it as a lesson and try to avoid these kinds of mistakes in future.