# **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**: 2

**Members Present**:

|  |  |
| --- | --- |
| 1. Luca Novello | 4.Eric Yakimoff |
| 2. Philip Ayomide Tijani | 5. Tyler Kay |
| 3. Karishma Singh Mahender | 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** |
| Eric | Analyze problem, Scrum report, Data structures | n/a |
| Luca | Analyze problem, Scrum report, Data structures | n/a |
| Karishma | Analyze problem, Test-plan document | n/a |
| Phillip | Analyze problem, Test-plan document | n/a |
| Tyler | Analyze problem, Test-plan document | n/a |

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** |  |

**Summary of Meeting:**

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

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| Dividing Work | Discussed the tasks that needed to be completed. Discussed who preferred what to get a broad sense of what work would be done by who. Further clarified responsibilities through JIRA. | Would use JIRA to allocate tasks amongst ourselves. Scrum Report Completed |
| Test Plan | Main outlook for how to setup the tests | Test plan completed |
| Analyzed Problem | Shared ideas for how to approach a solution | Data structures 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 distribution | Tasks were dispersed based on comfort with the skillsets needed as well as an emphasis on splitting the work equally. |
| Data Structure creation | Problem was analyzed by the group and then refined through individual contributions. |
| Testing approach | The test plan was initially discussed as a group and then further refined through individual contributions. |

**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? |
| Eric | Analyze problem, task distribution, data structures | 10 min | Some |
| Luca | Analyze problem, task distribution, data structures | 10 min | Some |
| Karishma | Analyze problem, task distribution, lead on test-plan | 10 min | Some |
| Philip | Analyze problem, task distribution, test-plan | 10 min | Some |
| Tyler | Analyze problem, task distribution, test-plan | 10 min | Some |

**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 |
| Eric | Review milestone deliverables, distribute tasks, scrum report, reflections |
| Luca | Review milestone deliverables, distribute tasks, scrum report, reflections |
| Karishma | Review milestone deliverables, distribute tasks, scrum report, reflections |
| Philip | Review milestone deliverables, distribute tasks, scrum report, reflections |
| Tyler | Review milestone deliverables, distribute tasks, scrum report, reflections |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Preliminary prioritization of tasks | Allows members to specialize and get comfortable with the work they prefer doing |
| Preliminary development of the data structures | Provided a base to work off when further structuring data.h |
| Preliminary development of the Test Plan | Provided a base to work off when further structuring of the test plan. |
|  |  |
|  |  |
|  |  |
|  |  |

**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 meeting | Meeting was productive in establishing a starting point for the week’s tasks |
| Communication | Clear and honest discourse around the parts of the project we were all comfortable with ensured that work would be completed |
| Git | Allows us to share work easier and show our contributions |
| Jira | Allows us to communicate tasks and responsibilities easier |

**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 |

**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.

Answers are in green

#ifndef DATA\_H

#define DATA\_H

#define GRID\_SIZE 25 // Grid length and width

#define TRUCK\_MAX\_WEIGHT 2500 // max truck weight

#define TRUCK\_MAX\_VOLUME 100 // max truck volume

// Struct for each Cell in the grid, Map will be a 2d array of Cells and position will be indexes

struct Cell

{

int isBuilding; // returns true if cell is a building

int isPath; // returns true if cell is a path

char color; // color of cell if on path ['B'='BLUE', 'Y'='YELLOW', 'G'='GREEN', 'N'='N/A']

};

// Struct for a Package or Shipment

struct Package

{

double weight; // weight of package

double volume; // volume of package

int destinationRow; // Row number of package destination

char destinationCol; // column character of package destination. Will need to get converted to an array index later in function (through ASCII).

};

// Struct for a Truck

struct Truck

{

char routeColour; // the color of the path taken by the truck ['B'=blue, 'Y'=yellow, 'G'=green]

double currentWeight; // the current weight of the truck

int currentVolume; // the current volume of the truck

int \*route; // an array holding a list of indexes for the route, might create struct for destination and make this an array of that struct

int routeLength; // The length of the route

};

#endif

1. Describe the process you used to analyze and understand the existing software code.  
     
   To analyze and understand the existing code the main process we used was looking at individual functions at first, and then looking at the whole project. This was particularly effective because we were able to understand each function at a high level, and while doing so make assumptions about how the functions are integrated.

Then our assumptions would be quantified once we examined more functions. By the end we were able to clearly see the full picture and understand how the entire program operated, and also how the variables and existing data structures integrated into the program. Understanding the functions and their relationship with other functions made understanding the data structures easier, as each element of the structure was given purpose and use through the functions.

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

Creating the test plan for the delivery system project involved critical considerations to ensure the system's efficiency and reliability. We focused on understanding the core requirements—allocating delivery locations based on route proximity and truck capacity. The test scope was clearly defined to include essential functionalities such as route assignments, the pathfinding algorithm, and capacity-based package assignments, while excluding post-allocation interactions. A multi-level testing strategy was adopted, covering unit, integration, system, and acceptance testing to comprehensively evaluate every aspect of the system.

The milestones for our test plan were carefully planned to make sure we progressed in an organized way. First, we made sure the test plan was complete and approved. Next, we set up a strong testing environment. After that, we laid out specific schedules for different types of testing: unit, integration, system, and acceptance tests. We regularly checked on our progress and made adjustments if needed. Finally, we reviewed everything, documented the results, and got formal approval to confirm that everything was ready for the final deployment.