## 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**: \_\_\_\_\_\_4\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1. Judd Niemi | 4. Riaz Hossain |
| 1. Mustafa Siddiqui | 5. |
| 1. Abdiwali Warsame | 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.

Note: Your professor will **only grade** the **master** or **main** branch, unless you indicate otherwise.

**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) | 20% |
| Git usage (used properly with good structure) | 15% |
| Jira usage (creates issues, tracks progress) | 25% |
| 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 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** |
| **Riaz Hossain** | **Helped complete scrum report + reflection 1** | **none** |
| **Mustafa Siddiqui** |  | **none** |
| **Abdiwali Warsame** |  | **none** |
| **Judd Niemi** | **Helped complete scrum report + reflection 2 + 3** | **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 |
| MS2 workload distribution | **Talked about the workload and which member should do which task** | **Members were assigned their tasks for MS2** |
| Submission deadline | **Finalized the submission date and time to set a deadline for all tasks** |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Mustafa Siddiqui | **Source code completion**  **testing** | **60 mins** | **yes** |
| Riaz Hossain | **Scrum report + reflections** | **35 mins** | **yes** |
| Judd Niemi | **Scrum report + reflections** | **60 mins** | **yes** |
| Abdiwali Warsame | **Creating test plan** | **60 min** | **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 |
| Mustafa Siddiqui | SOURCE code + testing |
| Riaz Hossain | Scrum report and reflections |
| Judd Niemi | SOURCE code + testing |
| Abdiwali Warsame | Creating test plan |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Tasks assigned to members | **Work evenly distributed** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Everyone showed up on time to discuss the workload | **Very active group** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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.  
     
   Date Structures used:

struct Shipment {

double weight;

double volume;

struct Point destination;

struct Truck {

struct Route route;

struct Shipment cargo[MAX\_SHIPMENTS];

int numShipments;

double currentWeight;

double currentVolume;

int truckNumber;

struct ShipmentInput {

double weight;

double boxSize;

char destination[10];

int isValid;

struct DeliveryResult {

int success;

int truckIndex;

int needsDiversion;

double distanceToGo;

In this milestone, we designed multiple data structures to represent the components of a delivery system. The Shipment struct holds information about a single package, including its weight, volume, and destination, represented by a Point. This allows us to track individual packages and their delivery details efficiently. The Truck structure models a delivery truck. It includes a route that the truck follows, an array cargo to store up to 50 shipments, and counters like numShipments, currentWeight, and currentVolume to manage the truck’s capacity. The truckNumber field identifies each truck by a unique number (0 for Blue, 1 for Green, 2 for Yellow). The ShipmentInput struct collects and validates user input for new shipments. It includes weight, boxSize, and a string destination, with an isValid flag indicating if the input passed validation. The DeliveryResult structure stores the outcome of trying to assign a shipment to a truck. Fields like success indicate whether a truck was found, truckIndex identifies which truck, needsDiversion shows if the truck must go off route, and distanceToGo records how far off route it must travel

1. How did analyzing the project requirements and design before starting the coding process help you identify potential challenges or define a clear development strategy?  
     
     
     
   we were able to identify possible issues early on, like how to integrate the given blackboard code and identify what extra data structures we needed. We carefully reviewed the projects requires and designs before starting any coding. It helped us comprehend the connections between input validation, truck capacity checks and pathfinding. This helped us avoid wasting lots of our time on the project.
2. How did creating a test plan help you ensure comprehensive test coverage for the project? Reflect on how defining objectives, scope, and test cases in advance influenced the effectiveness and efficiency of your testing process.  
     
   By creating a test plan, it helped us ensure that test coverage would be comprehensive by allowing us to define clear objectives, scopes and specific test cases before we started coding. This helped us know what functions and parts of the code would need testing. This helped us ensure that we did not overlook anything important. By making it a plan for tests in advance it helped us with the test process and saved time. It also helped with team expectations.