# **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**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1. Mihyeon Park | 4. Manraj Singh |
| 2. Ana Masoumi | 5. Veronika Edith Turpo Meneses |
| 3. Hamzeh Khaled Nayef Muhiar | 6. Yashleen Brar |

## 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** |
| Mihyeon Park | GitHub commits and Jira usage, Data Structures, contributed to test plan, filled scrum report | **N/A** |
| Ana Masoumi | GitHub commits and Jira usage, Data Structures, test plan, reflection q1 | **N/A** |
| Hamzeh Khaled Nayef Muhiar | Filled Scrum Report | **N/A** |
| Manraj Singh | Contribution in Test Plan and Scrum Report | **N/A** |
| Veronika Edith Turpo Meneses | Contributed to Test Plan | **N/A** |
| Yashleen Brar | Filled the scrum report, contributed to test plan and reflection question Q2 & Q3. C | **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** | **N/A** |
| **Reason for delay or block** | **N/A** |
| **Impact on Project** | **N/A** |
| **Solution or work-around** | **N/A** |
|  |  |
| **Delayed or Blocked Task** | **N/A** |
| **Reason for delay or block** | **N/A** |
| **Impact on Project** | **N/A** |
| **Solution or work-around** | **N/A** |

**Summary of Meeting:**

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

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| Test Plans | **Discussed on how to write and implement tests in test plans** | **Test Report Completed** |
| Scrum Report | **Discussed on the reflection questions** | **Scrum Report Filled** |
| New header file | **Discussed on creating new data structures for new header file** | **Header File created** |
| Jira | **Created new tasks** | **New tasks created** |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Prioritization of tasks | Equal amount of work distributed to each member |
| Testing decisions | Wrote a test plan consisting of testing decisions to create the algorithm that covers every part of the scope mentioned |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 | **Filled Test Report** | **2 hours** | **Completed** |
| Everyone | **Filled Scrum Report** | **30 mins** | **Completed** |
| Everyone | **Discussed on data structures for new header file** | **1 hour** | **Completed** |
| Everyone | **Created new tasks in Jira and assigned the tasks themselves** | **20mins** | **Completed** |
|  |  |  |  |
|  |  |  |  |

**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 |
| Everyone | Update Jira with new tasks |
| Everyone | Commit new files to GitHub |
| Everyone | Contribute to Scrum Report and Reflection |
| Everyone | Contribute to creating Blackbox tests as test documents |
| Everyone | Contribute to creating function specification |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Test Plan | **Test Plan that will help with implementing test cases** |
| Header File | **New header file with new data structures that will help with testing the project and debugging** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 Report and reflections | **Everyone Contributed** |
| Jira tasks | **Each assigned task for themselves to work on** |
| Meeting | **Everyone attended** |
|  |  |
|  |  |
|  |  |
|  |  |

**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.  
     
   We created a separate header file called struct.h to store and define all the structures needed for the shipment, truck rout etc. Below I mentioned each struct and explained what each one does.
2. void validTruckPaths(struct Shipment shipment, struct Truck truck, const struct Map\* map, struct Route routes [MAX\_ROUTE], int\* size);

The `validTruckPaths` function identifies all valid paths a truck can take based on the shipment details and map structure, storing these paths in an array of routes.

1. int checkDestination(const struct Route\* route, struct Shipment shipment);

The `checkDestination` function verifies if a given route includes the shipment's destination.

1. void routePoints(struct Route route, struct Shipment shipment);

The `routePoints` function lists all points along a route for a specific shipment.

1. int checkRouteBuilding(const struct Route route, const struct Map\* map);

The `checkRouteBuilding` function ensures that no buildings obstruct the identified routes.

1. int bestRoute(struct Route\* routes[MAX\_ROUTE], struct Shipment shipment, int size);

The `bestRoute` function determines the shortest and most efficient route from a list of possible routes for delivering the shipment.

1. int checkTruckOverload(struct Truck truck, struct Shipment ship);

The `checkTruckOverload` function checks if adding a shipment would exceed the truck's capacity.

1. int checkBoxSize(struct Truck truck, float boxSize);

The `checkBoxSize` function verifies if a shipment's box size fits within the truck's dimensions.

1. int validBoxSize(float boxsize);

Lastly, the `validBoxSize` function ensures that a shipment's box size adheres to predefined truck restrictions. Together, these functions help streamline the logistics of delivering shipments efficiently and safely.

1. Describe the process you used to analyze and understand the existing software code.  
     
   We initiated the process of analyzing and comprehending the current software code by conducting individual reviews. Subsequently, we conducted group discussions to exchange insights and resolve any uncertainties. We concentrated on the primary functions and data structures, identifying the key components and their relationships. Additionally, we searched for any existing documentation and comments within the code. Two members collaborated to review the code line by line during pair programming sessions. This collaborative approach enabled us to develop a more comprehensive comprehension of the software's overall architecture and functionality.
2. What aspects did you consider when creating the test plan? What were the milestones you identified in the test plan?  
     
   During the development of the test plan, we considered various factors, such as the software's functionality, potential edge cases, and user scenarios. Our objective was to ensure that all critical functions were addressed and that they functioned as anticipated in various scenarios. Additionally, our objective was to guarantee consistency and repeatability by automating the tests. The milestones that we identified in the test plan were as follows:
3. The initial configuration and setup of the test environment
4. Developing test cases for individual components
5. Incorporating tests to evaluate the overall functionality of the system
6. Conducting assessments and recording findings
7. Iterating and improving tests in response to feedback and identified issues
8. This methodical approach guarantees that the software functions properly and adheres to the necessary standards.