# **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: 6**

**Members Present**:

1. Sahan Gallage
2. Ileperuma Gunarathna
3. Dhanuth Hennedige
4. Sean Li

## 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** |
| **Sahan** | **Create struct headers and test plan file** | **None** |
| **Ileperuma** |
| **Dhanuth** | **Scrum report/reflection** | **None** |
| **Sean** |
|  |  |  |
|  |  |  |
|  |  |  |

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** | **Professor cannot access our GitHub and Jira** |
| **Reason for delay or block** | **Technical difficulty** |
| **Impact on Project** | **Unable to receive a valid grade for MS1** |
| **Solution or work-around** | **Discuss with professor in class** |
|  |  |
| **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 |
| Milestone 2 discussion | Described what milestone 2 is about, that is, creating the structures and test plan going forward, as well as what to expect for milestone 3 to help us delineate what needs to be done on this one | **Understood the code, the kind of structs we need, what we need to test. We’re now definitely on the same page** |
| Division of tasks | **Divided the task among members** | **Divided the task among members** |
| Discussion of addition | What new structs we will require | **The new *shipments* struct header file 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 |
| Structures | The structures intuitively reflect how the problem is described |
| Tests | The kinds of tests we need to perform based on our programming knowledge and the shipping/dispatching/pathfinding nature of 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? |
| Sahan | **Structs/test plan** | **4 hr** | **Yes** |
| Ileperuma |
| Dhanuth | **Scrum/reflection** | **2 hr** | **Yes** |
| Sean |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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 |
| All | Read milestone 3 and perhaps 4, answer any lingering questions, uncertainties, address new issues leading up to milestone 3 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Milestone 2 understood | **By understanding this milestone, we’ll be able to complete it satisfactorily and be ready for the next phase, that is, milestone 3** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Milestone 3 | **We know programming and have good chemistry** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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.

Shipment Struct

c

struct Shipment {

int weight; // in kilograms (1–5000)

int volume; // in cubic meters (0.5, 2, or 5)

struct Point dest; // delivery destination on the map

};

Explanation:

The Shipment struct holds all the essential information for a single delivery:

weight is the shipment’s mass in kilograms, important for planning and safety.

volume is how much space the shipment occupies, used for packing and truck loading.

dest is a Point struct representing the delivery location on the map, which helps with routing and logistics.

Truck Struct

c

struct Truck {

struct Route route; // truck’s delivery route

int currentWeight; // how much weight the truck is carrying

int currentVolume; // how much volume is used in the truck

char name[10]; // truck name (e.g., "BLUE")

};

Explanation:

The Truck struct represents a delivery truck and its current status:

route (a Route struct) tracks the truck’s delivery path.

The attributes currentWeight and currentVolume measure the truck's cargo load to prevent exceeding its maximum capacity.

name functions as a distinct identifier for every truck that enables simple shipment tracking and assignment processes.

Summary:

The two structs utilize minimal data to optimize shipment and truck administration within the project. Their limited data storage helps maintain straightforward code while ensuring easy project maintenance.

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 analyzed the requirements/design by carefully and thoroughly perusing the instructions, in particular paying attention to and recognizing the introduction, milestones 2 & parts of 3, and the existing path-finding code itself. This way we have a clear and definitive idea of what to build (the shipments structure and a foundational test plan), so that we will be able to test everything thoroughly and fulfill the requirements of the project completely – a proper, procedural way of going from start to finish in the development process. Our main goals are to understand and expand the project code to completion (potential challenges) and brainstorming of what tests we need to run to cover all aspects of the project (clear development strategy). Our analysis for this milestone helps us address and arrive at the end of these major goals.
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.  
     
   Our test plan looks at all the different areas that the project program entails, from existing codes and milestone instructions to future test cases that will arise. We need to check that all functionalities not only run perfectly and smoothly individually, but together as a whole. By defining the overall objectives, we can be laser-focused on exactly what needs to be done. Setting in stone the scope directs us to only build what’s required and specified in the project (i.e. weight/size for trucks, the definition of the routes and grid). Finally, and most importantly, designing a rigorous test plan ensures that the code will run validly and smoothly for both the programmers and the end user. All of these gives us a clear sense of concrete direction, enabling us to create powerful and efficient testing.