# **MILESTONE 2** -- SFT221 Scrum Report and Reflection

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. Nguyen Dang Khoa Huynh | 4. Prince Ghumaan |
| 2. Song Nhat Nguyen | 5. Benson Liu (Can not contact) |
| 3. Mohamed Mohamed | 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 is 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, & project updated) | 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** |
| **Song Nhat Nguyen** | **Create new header file with data structures, reflection question 1, Test Plan** |  |
| **Nguyen Dang Khoa Huynh** | **Complete Scrum Report, Make sure all the tasks completed, check Header File and perform Problem Analysis.** |  |
| **Mohamed Mohamed** | **Reflection 2** |  |
| **Prince Ghumaan** | **Reflection 3** |  |
|  |  |  |
|  |  |  |
|  |  |  |

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 |
| Group Tasks | **Designated tasks for group members** | **Every member in team knew what to do and when to complete it** |
| Separate Tasks | **Determine who will do which parts** | **Each member in team knew what to do in this Milestone 2** |
| Scrum | **Complete the table, and reflection questions** | **Scrum report finished** |
| Test Plan | **Find the issues when testing perform.** | **Test Plan completed** |
| Jira | **Assign Tasks Schedule in Jira** | **Completed** |
| Git | **Git update to each branch** | **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 |
| Testing Decision | Optimized algorithm for shortest path possible. |
| Testing Functions | Shipment Allocation Function, Shortest Path Calculation Function, Capacity Calculation Function. |
| Decided to use branches on GitHub repo for every Jira programming task | To maintain organization and coherence in the main branch, the team utilized branching in the GitHub repository and will persist in using branches for various issues and programming tasks aligned with Jira tasks |
| Pull request and merging | The team determined that if a person submits a pull request from a separate branch, it must be reviewed and approved by another member. This choice maintains the integrity of the main branch and, via thorough peer code reviews, minimizes less bugs will be introduced or added to the main branch. |
|  |  |
|  |  |
|  |  |

**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? |
| Song Nhat Nguyen | **Created Header file Truck.h, Tested Plan, and performed reflection question 1** | **3 hours** | **Completed** |
| Nguyen Dang Khoa Huynh | **Scrum report, checked file Truck.h, make sure all tasks completed, performed problem analysis.** | **2 hours** | **Completed** |
| Mohamed Mohamed | **Performed Reflection 2** | **1 hours** | **Completed** |
| Prince Ghumaan | **Performed Reflection 3** | **1 hours** | **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 |
| Song Nhat Nguyen | Write descriptions for two functions. Create test cases for functions. |
| Nguyen Dang Khoa | Fill out the scrum report for MS3. Push all new files to GitHub and organize branches. Assign new tasks on Jira. Support Song Nhat Nguyen for testing cases. |
| Mohamed Mohamed | Answer reflection questions for MS3. Support Song Nhat Nguyen for testing cases. |
| Prince Ghumaan | Answer reflection questions for MS3. Support Song Nhat Nguyen for testing cases. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Header File | **H file creation will help with testing the project and to identity potential bugs and errors(Truck.h)** |
| Problem Analysis | **Helped members understand about the program and code walk through.** |
| Discussed testing strategies and approaches | **Will help the team to have proper mindset and plan for testing going forward in the project.** |
| Set up custom labels for Jira tasks and assigned tasks | **Custom labels facilitate the differentiation and organization of jobs according to milestone numbers.** |
| Set up branching techniques and learned about it | **Will assist the group members in structuring code and collaborating effectively on the GitHub repository.** |
|  |  |
|  |  |

**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 |
| Brainstorming for and discussing Test Plan | **Team members demonstrated respect for one another's viewpoints and contributed rational and efficient methodologies for test strategies, test execution, and overall test planning.** |
| Learn about important GitHub methods & concepts such as Branching | **The team members who had more awareness and expertise in GitHub Branching instructed other team members on Branching in GitHub repositories and implemented a sample.** |
| General teamwork & work distribution | **All team members were cooperative and assisted one another in executing tasks.** |
|  |  |
|  |  |
|  |  |
|  |  |

**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 have been asked to analyze a problem and design software (functions) to complete the solution without writing the software.
   1. Is this process more difficult than just writing the software to complete the project? If so, why is it more difficult? If not, why is it easier than just writing the software?

This process is either hard or easy. I can tell that the creation of functions only is easy at first because it basically based on the project description, however, even if I can do whatever I want with the functions, it still having to meet the requirement of the project. Also, I had to think more than just the project description to see is there any function that can help to make the program more efficient. This is the hardest part of just doing the functions without doing actual software. But on the other hand, writing the software is harder if we are counting on the small details inside such as the algorithm. There is a mile between creating thing in our knowledge and actual writing the software, because what we think is not what the machine thinking, so we have to take a lot of time to transfer it to the computer thinking.

* 1. Describe two advantages of developing software in this manner rather than just moving on to writing the functions without writing specifications first.

First, it is easy to manage. Especially if we are doing it in a group, we can discuss a lot of things with the functions and can eliminate or keep the function base on the situation is bad or good. Is the function will be capable with the project or not? This will help us to delete the function if is not suited with the program without worrying about the other functions that have a connection with the function that we just deleted.

Secondly, I can look at the functions with the vision of the user. It is hard to have this vision if we just doing the software because when we dive into the writing software, the only important left is the time, and will the software have bug or error or not, basically the software will be ignored a lot of function that user may want to have it. Because the developer is so busy on the other thing. This first manner is so good to help a developer or a team to look at what they want to give for their customer outside of the basic requirement from the company.

1. Why is it a good idea to create a test plan? Describe at least 3 advantages of test plans.

It is a good idea to create a test plan because it allows clarity and structure which can organize the things needed to be done like, what needs to be tested and how to approach the testing. It also improves efficiency and saves time by having set deadlines and assign members their tasks. Finally, there’s easier team management and accountability for all team members and with this it gives each member a set of responsibilities and a role they are suited for.

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

To learn about the software, I looked at main.c, mapping.h, and mapping.c. First, I looked at main.c to learn how the functions that fill out the map, set routes, and show map output are called by the program. The mapping.h file contained a detailed structure for the map, points, and routes. The mapping.c file carried out the main tasks, such as making the map grid with buildings and routes and finding the shortest route between points.   
  
Looking at these files and learning about their parts helped me understand how data is managed and sent between functions. GetBlueRoute sets up routes, and populateMap sets up the map structure. By going over these functions very carefully, I understood what they were supposed to do and how the modules worked together.   
  
With this information, I helped make truck.h, a header file that has structures for Truck, Package, and Path. Following the logic in the previous files, this header file sets out how the truck's capacity, package details, and routing paths will be organized and defined. Since we combined transportation features with map data, the truck.h file was set up in a way that made it easier to manage truck and package data.   
  
After figuring out the structures, I used the template to help make a test plan. The software's features were checked to make sure they met all the needs, especially those related to truck capacity limits and route validation. The test cases included limits on the truck's weight and volume, the accuracy of the route, and boundary conditions. After looking at, organizing, and testing the software, I knew how each part affects the whole and made sure it would work the way it was supposed to.