# **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**: Team4

**Members Present**:

|  |  |
| --- | --- |
| 1. Seulgi Lee | 4. Alejandro Mercado |
| 2. Khassan Suleimanov | 5. |
| 3. Philip Grahamm | 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** |
| Seulgi Lee | Write the scrum reports - Summary of Tasks Completed or Delayed in the last week, Write the scrum reports, Create a team contract, Write and sign a team contract, Create a GIT account, Create a Jira account, Create and Initialize the GIT repository, Set-up the Jira project, Add your professor to the GIT and Jira accounts, Update Jira with the work performed and planned | Create and Initialize the GIT repository (Seneca firewall wasn’t working so it delayed 3 days) |
| Khassan Suleimanov | Write the scrum reports, Write the Reflection, Write the Reflection, Create a GIT account, Create a Jira account, Update Jira with the work performed and planned | None |
| Philip Grahamm | Write the scrum reports, Write the Reflection, Write and sign a team contract, Create a GIT account, Create a Jira account, Update Jira with the work performed and planned | None |
| Alejandro Mercado | Write the scrum reports, Write the Reflection, Write and sign a team contract, Create a GIT account, Create a Jira account, Update Jira with the work performed and planned | 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** | Create and Initialize the GIT repository |
| **Reason for delay or block** | Failure in network access due to Seneca firewall issues. |
| **Impact on Project** | Caused a 3-day delay in the initial setup and repository configuration, resulting in significant time loss for project progress |
| **Solution or work-around** | Waited until the firewall issue was resolved and then completed the task. |

**Summary of Meeting:**

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

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| An analysis of the problem (no written artifacts produced) | **Understood the core of the program.**  - Number of trucks: 3 (yellow, blue, green)  - City structure: 25x25 grid  - Truck capacity: up to 2500 kg, 100 m³  - Box size: Small: 1m³, Medium: 3m³, Large: 5m³  - Customer requirements: cargo weight, box size, delivery location  - Workflow: Check capacity (weight and size) -> select optimal truck  - Route restrictions: No passing through black buildings  - Route navigation: Shortest route required  - Back-up plan: Next day delivery if not possible  - Operating period: Daily program  - Cargo acceptance: Until truck is full or no more freight.  **Reviewed Source codes.**  **Asked and discussed about necessary actions.** | Everyone understood the core of the project, and we all looked at the code together, talked about what was missing, and analyzed the code to see what the best options were for data structures. This increased understanding among team members and shared awareness of the core of the problem and the possible solutions. |
| A series of data structures created as header files and stored in the repository. | Discussed the issue of adding the new data structures needed.  Analyzed the existing data structures and functions, as well as the interactions between them.  Discussed what new header files should be created and what additional work needs to be done. | We realized that we need a data structure for the vans that takes into account the total weight capacity of the cargo, and the total volume capacity for the boxes. We also need a structure to manage the shipment specs. We also decided who would do the work and how it would be done. |
| Write a test plan and stored in the repository. | Reviewed the test plan document.  Discussed how to create a specific test plan. Checked how to store them in a repository. | We recognized that having a clear test plan will help you in future test runs. Recognize that having a clear test plan will help you in future test runs. |
| Write a test strategy and stored in the repository. | Reviewed the test strategy document.  Checked how to store them in a repository. | We realized that the test plan had a test strategy with it, and since it was the largest task, we assigned two people to it. |
| Completed scrum report including reflection questions answered. | Conducted a meeting based on a Scrum report.  Summarized their overall progress.  Asked each team member to verbally answer reflective questions about the project. | We felt that Scrum increased team members' understanding and engagement with the project, including what had been done, what still needed to be done, and their overall understanding of the project. It also allowed us to get feedback for continuous improvement. Overall, we decided to use the meetings to get ideas and assign someone to oversee organization. |

**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 |
| Understanding the Core of the Program | Recognizing the core structure and problems.  All team members gained a clear understanding of the project’s core through code review to identify issues. |
| Adding New Data Structures | Decision to create new data structures and header files.  Implementing structures to create vans with total cargo weight, and box volumes, as well as the shipments to be sent. |
| Development of Testing Plan and Strategy | Reviewing the testing plan and strategy documents.  Assigned the largest task of formulating the test plan to two team members due to its complexity. |
| Completing Scrum Reports and Reflection | Conducted a meeting based on a Scrum report to summarize progress.  Collected feedback on each team member’s understanding of the project and remaining tasks. |
| Prioritization and Assignment of Tasks using Jira | Decided on priorities for critical tasks and designated responsible persons.  Setting up new tasks, assigning priorities, and designating responsible individuals in Jira. |

**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? |
| Seulgi Lee | An analysis of the problem. | 30 min | **Yes** |
| Khassan Suleimanov | An analysis of the problem. | 30 min | **Yes** |
| Philip Grahamm | An analysis of the problem. | 30 min | **Yes** |
| Alejandro Mercado | An analysis of the problem. | 30 min | **Yes** |
| Seulgi Lee | Tested storing documentation or source code in the Git repository. | 10 min | **Yes** |
| Khassan Suleimanov | Tested storing documentation or source code in the Git repository. | 10 min | **No (Mac issue)** |
| Alejandro Mercado | Tested storing documentation or source code in the Git repository. | 20 min | **Yes** |
| Seulgi Lee | Exchanged opinions on the answers to the reflection questions. | 20min | **Yes** |
| Khassan Suleimanov | Exchanged opinions on the answers to the reflection questions. | 20min | **Yes** |
| Philip Grahamm | Exchanged opinions on the answers to the reflection questions. | 20min | **Yes** |
| Alejandro Mercado | Exchanged opinions on the answers to the reflection questions. | 20min | **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 |
| Seulgi Lee | Complete scrum report. |
| Seulgi Lee | Write the Reflection, and when all of team members finished, combine them into a group reflection. |
| Seulgi Lee | Made timeline and board about Milestone2 using Jira and assign team members. |
| Seulgi Lee | Update Jira with the work performed and planned |
| Khassan Suleimanov | Tested storing a documentation or source code in the Git repository. |
| Khassan Suleimanov | Write a test strategy. |
| Khassan Suleimanov | Stored a test strategy in the repository. |
| Khassan Suleimanov | Update Jira with the work performed and planned. |
| Philip Grahamm | Tested storing a documentation or source code in the Git repository. |
| Philip Grahamm | Write a test plan. |
| Philip Grahamm | Stored a test plan in the repository. |
| Philip Grahamm | Update Jira with the work performed and planned. |
| Alejandro Mercado | A series of data structures created as header files. |
| Alejandro Mercado | Stored a series of data structures created in the repository. |
| Alejandro Mercado | Update Jira with the work performed and planned. |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Understanding the core of the program. | Reviewed the source code of a given program and understood the key issues, enabling team members to better collaborate and solve problems. |
| Recognize what new data structures need to be added. | Recognized what new data structures are needed to move the project forward, bridging the gap between tasker and other team members and minimizing future delays due to rework. |
| Reviewed documentation for test plan and strategy | Guided documentation of test plans and strategies Increased awareness and clarity of direction for future projects |
| Agree on what goes into the Scrum report and the reflection. | Drive continuous improvement and better team alignment for more effective project management |
| Prioritizing and assigning tasks using Jira. | Enhanced task management and accountability, ensuring critical tasks are addressed timely. |

**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 |
| Held scrum meeting after class | Held a Scrum meeting immediately after class. It was time-efficient, allowed everyone to meet in person, and we could ask the professor questions directly if needed. |
| An analysis of the problem | Conducted an analysis of the problem together. By sharing what we know and don't know, our overall understanding of the project improved significantly. |
| Tested storing documentation or source code in the Git repository | Ensured that future uploads to the Git repository by each member will work smoothly. |
| Exchanged opinions on the answers to the reflection questions | Sharing different perspectives helped us come up with better answers and improve the quality of our responses. |

**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 |
| Test storing a documentation or source code in the Git repository. | When running a test to save documentation or source code to a Git repository, one team member's MAC didn't work properly and we couldn't find a workaround, so we decided to run the test on a separate Windows device. |

**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.

**<Existing data structure provided>**

|  |  |
| --- | --- |
| **Code** | **Explain each item** |
| struct Map  {  int squares[MAP\_ROWS][MAP\_COLS];  int numRows;  int numCols;  }; | **`struct Map` is a structure that represents information about a map.**  **int squares[MAP\_ROWS][MAP\_COLS];**  => An array representing a 25x25 map, which stores information about buildings, empty spaces, routes, etc.  **Int numRows;**  => The number of rows that represent the size of the map. This means that the map can be dynamically resized using that value.  **Int numCols;**  => The number of cols that represent the size of the map. This means that the map can be dynamically resized using that value. |
| struct Point  {  char row;  char col;  }; | **`struct Point` is a structure that represents a specific location on the map.**  **Char row;**  => Row to represent specific location on a map.  **Char col;**  => Col to represent specific location on a map. |
| struct Route  {  struct Point points[MAX\_ROUTE];  int numPoints;  char routeSymbol;  }; | **`struct Route` is a route made up of multiple specific locations.**  **Struct Point points[MAX\_ROUTE]**  **=**> An array that can hold up to 100(depends on `MAX\_ROUTE`) Point structures, storing the points that make up the path.  **Int numPoints**  => A variable that stores the number of points (locations) included in the path.  **Char routeSymbol**  => A character used to visually distinguish routes. This character is used when displaying routes on a map. (Blue, Green, Yellow) |

**<New data structure to be created> - This is an early version and may vary in implementation.**

|  |  |
| --- | --- |
| **Code** | **Explain each item** |
| struct Van {  int capacity;  int weight;  }; | **`struct van` is information about each van.**  **int currentVolume;**  => current volume usage  **int weight;**  => current weight of the truck |
| struct Shipment {  double weight;  double size;  struct Point destination;  }; | **`struct Shipment` is the information submitted by the customer for the shipment.**  **double weight;**  => weight of the shipment  **double size;**  => size of the required box  **struct Point destination;**  => Shipment destination |

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

**1) Our team identified the problem and requirements that the program should solve.** This gave us a clear idea of what the program should do and what features it should include.

**2) We read the code we were given piece by piece to see how it worked.** We tried to modify some of the code and added the 2 remaining routes so we could see how the program actually worked with the 3 routes being displayed simultaneously.

**3) We ran unit tests** to make sure each feature worked properly and that there were no bugs, and if we found any, we fixed them.

**4) We used the Visual Studio debugger on the program** to understand the flow of the entire application. The debugging tool allowed us to see in detail how variables change and in what order are the function calls.

**5) We discussed any complicated parts of the code or parts** that we didn't understand with our teammates. We would all sit down and review the code, sharing different opinions and trying to find the best solution.

This systematic approach helped us understand the code better and gave us a good foundation to improve it to meet our project goals.

1. **What aspects did you consider when creating the test plan? What were the milestones you identified in the test plan?**  
     
   **When creating the test plan, we considered:**

**1) Test Objectives and Purposes**: Based on an understanding of the project, we clarified the objectives of the test and provided an overview of how the testing will be conducted.

**2) Test Scope**: We explicitly defined which features will be tested and which will be excluded, clearly delimiting the scope of the testing.

**3) Test Schedule and Sequence**: We planned the overall schedule of the tests, providing estimated times for when testing will start and end.

**4) Test Strategy**: We described the approach for various types of tests, such as functional tests, integration tests, and white box tests, thereby establishing a comprehensive test execution strategy.

**5) Environment Requirements**: We defined the hardware and software environments needed for testing and clarified the necessary conditions before setting up the test environment.

**6) Execution Strategy**: We outlined the entry and exit criteria for the tests, categorized the severity of potential defects, and planned the testing process in detail.

**7) Roles and Skills of the Testing Team**: We specified the resources needed and the responsibilities of each team member, ensuring clarity in roles throughout the testing process.

**In the test plan, we identified the following milestones:**  
**1) Completion of Test Plan**: Finalize the test plan document, covering the test strategy, environment, data, and schedule.

**2) Setup of Test Environment**: Ensure all necessary hardware and software for testing are set up.

**3) Preparation of Test Data**: Prepare and verify all required test data.

**4) Execution of Initial Tests**: Conduct exploratory and initial functional tests.

**5) Defect bugs and Fixes**: Report and fix any defects found during initial tests.

**6) Completion of Functional Testing**: Create and execute test cases for all main software functions.

**7) Completion of Edge Case Testing**: Develop and run tests for edge cases to verify software behavior under unusual conditions.

**8) Completion of Performance Testing**: Perform tests to check the software's performance and efficiency.

**9) Completion of Regression Testing**: Run regression tests to ensure existing features work correctly after updates.

**10) Completion of Automated Testing**: Develop and execute automated tests to improve testing efficiency.

**11) Completion of User Experience Testing**: Conduct user acceptance testing to ensure the software meets end-user needs.