# SCRUM Report and Reflections – 2

**Members Present**:

|  |  |
| --- | --- |
| 1. Divya Devendrasinh Rana | 4. Prabhjot Singh Longia |
| 2. Harsh Pahurkar | 5. |
| 3. Mukul Sharma | 6. |

## Milestone 2 Tasks

* 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.

**SCRUM Report**

**Summary of Tasks Completed or Delayed in the last week:**

|  |  |  |
| --- | --- | --- |
| **Member** | **Tasks Completed** | **Tasks Delayed/Blocked** |
| **Divya** | **Formulated and maintained a well-structured test plan for the milestone.** |  |
| **Harsh** | **Helped create SCRUM report, mediate the weekly meet and jot down important points.** |  |
| **Mukul** | **Helped ideate, create, and solidify the team’s understanding on the problem through working on reflect questions.** |  |
| **Prabhjot** | **Primarily worked on formulating, creating data structures for the milestone.** |  |
|  |  |  |
|  |  |  |
|  |  |  |

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:**

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| Significance of a test plan | **It is important to plan out variety of tests to ensure holistic and efficient testing.** | **Test Plan is discussed throughout the group and finalized.** |
| Creating data structures and header files | **Delivery.h is to be made where new code for structs, required for keeping track for existing attributes of code, is kept. This header file will also contain certain macros.** | **Discussed what exactly are data structures. The team also rewatched parts of the lecture.** |
| Problem-statement | **Thorough understanding of the task given i.e creating test plan and data structures.** | **Reflection questions were answered, allowing team to have a thorough understanding of the milestone.** |
| Analysis and Design brainstorm | **Header files are to be put separately where data structures are defined as well as other technical aspects for execution of milestone.** | **Data Structures are implemented in the header file with everyone being well-versed with how they work.** |
|  |  |  |
|  |  |  |
|  |  |  |

**Summary of Decisions Made:**

|  |  |
| --- | --- |
| Decision | Rationale |
| Creating new data structures such Shipment, Truck and Diverted Routes | Creating shipment, truck and diverted routes to keep track of shipment, truck, diverted route. |
| Test plan will be made by Divya, then double checked by the team after | The team agrees to thoroughly go through the plan after the base has been set to make sure it is solid. |
| Reflect will be made by Mukul, then double checked by the team after | Reflect questions have been gone through by the team to better understand the content being taught in the milestone. |
| A new header file “delivery.h is introduced to define macros and new structs | This new header file is implemented into the code to specify the newer code about to be added as well as macros such as MAX\_WEIGHT and MAX\_VOLUME to make the functioning of newer code efficient. |
| The team will double check the SCRUM report after it is finished | Everybody contributes and adds to the SCRUM reports after it has been made. |
|  |  |
|  |  |

**Tasks Attempted During Meeting:**

|  |  |  |  |
| --- | --- | --- | --- |
| Member | Task Attempted | Time Spent | Complete? |
| Divya | **Made a solid test plan** | **1 hour** | **YES** |
| Harsh | **Worked on the SCRUM report** | **1 hour 30 mins** | **YES** |
| Mukul | **Worked on and built the base for reflection.** | **1 hour** | **YES** |
| Prabhjot | **Created data structures and the extra necessary code for the milestone** | **1 hour** | **YES** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**SCRUM Tasks Selected for Next Week**:

|  |  |
| --- | --- |
| Group Member | Task Description |
| Divya | Will be responsible for Black box tests. |
| Harsh | SCRUM Report to be prepared. |
| Mukul | Reflection to be created and maintained. |
| Prabhjot | Functions to be implemented properly in the code. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| New data structures in the header file are discussed | **Discussion for keeping track of trucks and shipments in the program, which was required in the program.** |
| Reflect questions were discussed | **The team better understood the assignment and the required learnings by discussing test plan and making new data structures.** |
| SCRUM report has been discussed | **Good communication amongst the team was practiced where everybody’s work is mentioned.** |
| Test plan was discussed | **The test plan discussion allowed the team to not leave any loopholes while testing the program.** |
|  |  |
|  |  |
|  |  |

**Things That Went Well in This Meeting:**

|  |  |
| --- | --- |
| Topic/Work Item | Reason for Success |
| Data Structures have been made | **Enthusiasm and willingness to understand the problem statement to solve it.** |
| Reflect is finished | **Team’s communication and discussion made thorough reflect answers.** |
| Test Plan is ready | **Test plan was thoroughly discussed by the team to make sure there are no loopholes while testing the program.** |
| SCRUM Report is ready | **Made possible thanks to clear communication amongst the team.** |
|  |  |
|  |  |
|  |  |

**Things That Did NOT go Well in This Meeting:**

|  |  |
| --- | --- |
| Topic/Work Item | Reason for Problem and How to do Better |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Reflections**:

1. In this milestone you have been asked to analyze a problem and design software(functions) to complete the solution without actually 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?  
        
      Analyzing a problem and designing software functions without writing the software can be more challenging than simply writing the software to complete the project. This is because the process of analysis and design requires a deep understanding of the problem and careful consideration of various factors such as requirements, constraints, and user needs. It involves breaking down the problem into smaller components, identifying the necessary functionalities, and determining how they should interact with each other.

In contrast, writing the software itself involves translating the design into code, which can be more straightforward once the design is well-defined. It requires technical knowledge and programming skills to implement the functions according to the design specifications.

However, the analysis and design phase is crucial as it sets the foundation for the software development process. It helps ensure that the software meets the desired objectives and requirements and reduces the risk of errors or inefficiencies during implementation. Therefore, while it may be more challenging, the process of analyzing and designing software functions is essential for creating a successful software solution.

* 1. Describe two advantages of developing software in this manner rather than just moving on to writing the functions without writing specifications first.  
       
     Developing software by analyzing and designing functions before writing the code has a couple of advantages over immediately jumping into writing functions without specifications:

1. Better understanding and fewer mistakes: Taking the time to analyze and design software functions helps us understand the problem more clearly. It reduces confusion and ensures that the software meets the requirements accurately without making mistakes.
2. Efficient use of resources: By investing time in analyzing and designing functions, we can use our resources more efficiently. It helps us identify potential issues early on, so we can plan better and avoid wasting time and effort on unnecessary changes later.

1. Why is it a good idea to create a test plan? Describe at least 3 advantages of test plans.  
     
   Creating a test plan is a good idea for several reasons. Here are three advantages of having a test plan:
   1. Ensures comprehensive testing: Having a test plan helps ensure that the software undergoes thorough testing. It provides a structured approach to testing, outlining the specific test cases and scenarios that need to be covered. This helps identify potential defects and vulnerabilities, ensuring that all critical functionalities and edge cases are tested before the software is released.
   2. Improves software quality: A well-defined test plan contributes to the overall quality of the software. It helps in identifying and resolving bugs, errors, and usability issues early in the development process. By executing a range of test cases, the test plan helps in verifying the correctness, reliability, and performance of the software. This improves user satisfaction and reduces the risk of costly post-release issues.
   3. Facilitates collaboration and communication: A test plan serves as a reference document for the testing team, development team, and stakeholders involved in the software project. It outlines the testing scope, objectives, and resources required. This promotes effective collaboration and communication among team members, ensuring that everyone is aligned on the testing approach and goals. It also helps in coordinating efforts, tracking progress, and managing expectations throughout the testing phase.

1. Describe the process you used to analyze and understand the existing software.
   * To analyze and understand existing software, a systematic process can be followed:
   * Gather information: Collect available documentation and resources related to the software.
   * Explore the user interface: Interact with the software's interface to understand its features and functionality.
   * Review the source code: If accessible, examine the code to understand its structure and functions.
   * Identify dependencies: Determine external components that the software relies on.
   * Trace data flow: Follow how inputs are processed and transformed within the software.
   * Identify key components: Recognize the main modules or components and their interactions.
   * Understand business rules: Grasp the workflows, algorithms, or decision-making processes implemented.
   * Note limitations and issues: Document any known limitations or problems with the software.

By following this process, developers can gain insights into the software's structure, functionality, and code, which helps inform decision-making and potential modifications.