# **MILESTONE 2** -- SFT221 SCRUM Report and Reflections

This report should be completed in the class and submitted at the end of class. Late submissions cannot be accepted without prior approval of the instructor.

**GROUP**: 4

**Members Present**:

|  |  |
| --- | --- |
| 1. Song Hwan Oh | 4.Sangjune Lee |
| 2. Shine Lee | 5. Ji Ho Nam |
| 3. Yoojin Lee | 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 at End of Lab**

* Completed SCRUM report & reflections

**Deliverables Due within 48 hours of lab**

* 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.

**Rubric**

|  |  |  |
| --- | --- | --- |
| Individual | Group Participation | 75% |
| Teamwork | 10% |
| SCRUM Report | 15% |
| Group | Data structures (complete, correct and well-designed) | 20% |
| Test Plan (complete, well-written) | 20% |
| Git Usage (used properly with good structure) | 10% |
| Jira Usage (creates issues, tracks progress) | 10% |
| Meets Deadlines | 15% |
| SCRUM Report and Reflections | 25% |

**SCRUM Report**

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

Here you can list all of 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 Hwang Oh** | **SCRUM, Git contribution, Contrat** | **N/A** |
| **Shine Lee** | **SCRUM, Git contribution, Contrat** | **N/A** |
| **YooJin Lee** | **SCRUM, Git contribution, Contrat** | **N/A** |
| **Sangjune Lee** | **SCRUM, Git contribution, Contrat** | **N/A** |
| **JI Ho Nam** | **SCRUM, Git contribution, Contrat** | **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 discusses in the meeting and the outcomes of the discussions.

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| SCRUM | **SCRUM done** | **SCURM Finished** |
| TEST PLAN | **TEST PLAN done** | **TEST PLAN Finished** |
| NEW H FILE | **NEW H FILE DONE(Finder.h)** | **H FILE Finished** |
| Jira | **Task Schedules setup 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 |
| Prioritization of tasks | Equal amount of works assigned to each member of team. |
| Testing Decision | Optimized algorithm for shortest path possible. |
| Testing Functions | Shipment Allocation Function, Shortest Path Calculation Function, Capacity Calculation Function, Output Message Generation Function |
|  |  |
|  |  |
|  |  |
|  |  |

**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 could not be completed, the student should indicate why this was not possible.

|  |  |  |  |
| --- | --- | --- | --- |
| Member | Task Attempted | Time Spent | Complete? |
| ALL | **Analysis discussion, Analysis of problem in program and discussed** | **30min** | **Yes** |
| ALL | **Writing SCRUM report together** | **1hr** | **Yes** |
| ALL | **H file(Finder.h)** | **1hr** | **Yes** |
| ALL | **Test plan** | **30min** | **Yes** |
| ALL | **Jira and Github Project page updated and assigned** |  |  |
|  |  |  |  |
|  |  |  |  |

**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 |
| Sangjune Lee | Jira control |
| ALL | Meeting on July 17th Monday 9pm to 11pm |
| ALL | SCRUM , Reflection |
| ALL | Function specification(name, purpose, description , return type and parameters) |
| ALL | A set of blackbox tests as test documents with test data for the functions. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| H File | **H file creation will help with testing the project and to identity potential bugs and errors(Finder.h)** |
| Test Planning | **Testing specification has been set through group members.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 | All contributed. |
| Git | **Useful for version control and keeping track of changes** |
| Meeting | All attended second meeting. |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 actually writing the software can be more difficult than just writing the software directly to complete the project.

In the absence of writing the software, there is a risk of missing certain important details. By not going through the implementation process, there is a possibility of overlooking certain complexities that may arise during development. Secondly, without implementing the software it is harder to understand the problem. It requires abstract and conceptual thinking of tackling the problem and potential solutions

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

i. By first analyzing the problem and designing the software functions, you can establish clear requirements and specifications for the software. This ensures team members/stakeholders have a shared understanding of what the software needs to accomplish.

ii. The process of analyzing and designing software allows for better planning and organization of the development effort. It helps identify potential challenges, dependencies, and risks early on, allowing for more effective resource allocation, scheduling, and prioritization of tasks.

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

Structured Approach: Test plans provide a clear and organized framework for testing. It helps testers follow a systematic approach and ensure that all necessary aspects of testing are covered.

Comprehensive Coverage: Test plans help in identifying and addressing all functional requirements, features, and scenarios that need to be tested. This ensures thorough testing and reduces the risk of missing critical areas.

Effective Communication: Test plans act as a communication tool between different stakeholders involved in testing. They facilitate better collaboration and understanding among team members, including developers, project managers, and clients.

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

Review: We examined the header file, and configuration files to understand the implementation details and identify potential areas for improvement or issues.

Support: We collected supporting related to the software. Analyzing reported issues provided valuable insights into the strengths, weaknesses, and improvement areas of the software.

Collaboration and Discussions: we engaged in discussions with the group to clarify any doubts and gather additional information.