# **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**: 1

**Members Present**:

|  |  |
| --- | --- |
| 1. Taehwa Hong | 4.Hyunjoo Han |
| 2. Jenna Moon | 5. Natalya Pak |
| 3. Farouk Alhassan | 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, & project updated) | 20% |
| Test plan (complete, well-written) | 20% |
| Git usage (used properly with good structure) | 10% |
| Jira usage (creates issues, tracks progress) | 15% |
| Scrum report & reflections | 25% |
| Meets deadlines | 10% |

**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** |
| **Taehwa Hong** | **Group Contract, Scrum report & reflection and setting up github & jira** | **None** |
| **Jenna Moon** | **Group Contract, Scrum report & reflection** | **None** |
| **Farouk Alhassan** | **Group Contract, Scrum report & reflection** | **None** |
| **Hyunjoo Han** | **Group Contract, Scrum report & reflection** | **None** |
| **Natalya Pak** | **Group Contract, Scrum report & reflection** | **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** | **None** |
| **Reason for delay or block** | **None** |
| **Impact on Project** | **None** |
| **Solution or work-around** | **None** |
|  |  |
| **Delayed or Blocked Task** | **None** |
| **Reason for delay or block** | **None** |
| **Impact on Project** | **None** |
| **Solution or work-around** | **None** |

**Summary of Meeting:**

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

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| Jira | **Confirm tasks** | **Task updated and assigned on Jira** |
| Header File | **Analyze the code provided and create new header file** | **New header file updated on Git** |
| Git | **Keep updating the file and communicate with other group members** | **Files and Source code updated on Git to each branch for version control** |
| Test Plan | **Analyze the code and plan** | **Test plan completed** |
| SCRUM | **SCRUM done** | **SCURM Finished** |
|  |  |  |
|  |  |  |

**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 |
| Task Distribution | Assigned fair amount of work to each member |
| Testing Decision | Algorithm optimized for the shortest possible path. |
| Testing Function | Send assignment function, shortest path calculation function, capacity calculation function, output message generation |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| All | **Analysis the problem of program** | **3hours** | **Y** |
| All | **Finished Scrum Report** | **2hours** | **Y** |
| All | **Header File** | **2hrs** | **Y** |
| All | **Test Plan** | **3hours** | **Y** |
| All | **Updated Jira** | **20mins** | **Y** |
| All | **Updated Github** | **20mins** | **Y** |
|  |  |  |  |

**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 | Jira management |
| All | Holding meetings |
| All | Scrum report & reflection |
| All | Specify functions |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Task distribution | **Distribute a fair amount of work to all members** |
| Test decision | **Decide what tests to run** |
| Creating new header file | **Build plan to complete the program** |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Git | **Useful for tracking changes and version control** |
| Jira | **Useful for managing To-do list** |
| Scrum Report | **All member participated** |
| Meeting | **All member attended** |
|  |  |
|  |  |
|  |  |

**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 |
| None | **None** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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?

As a software tester, the process of analyzing a problem and designing software functions before diving into actual software writing can make the overall project significantly easier. While it might seem difficult and more complicated at first, this preparatory phase has many advantages over immediately jumping into software development. Comprehensive analysis of a problem provides a deeper understanding of the issue of the software, building a solid foundation for the entire project. By carefully examining the problem, potential issues down the road can be found and identified from the start. It also helps with understanding clear requirements for the software. This extra step might seem to be taking longer, but it will lead to more efficient solutions to the problem. If there are challenges during software development, a planned-out strategy from the analysis facilitates a smoother process. This method will significantly minimize the costly rework and strengthen the overall efficiency of the development process. On the other hand, leaping straight into writing software without prior analysis might seem faster initially, but it will possibly lead to more complexities, unexpected roadblocks, and more expensive rework, making the whole process more difficult and less efficient.

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

There are two distinct advantages to developing software through thorough analysis and design. Firstly, this phase will help understand the requirements clearly as a team. The team can figure out how they are going to make a software map, making sure everyone understands the requirements and any implications. Any missing and misleading details will be cleared up through analysis of the problem via team communication. Secondly, this helps us to define the problem and potential solutions clearly. We can break it down into smaller, more manageable steps, making the process easier. Through in-depth analysis, we can discover hidden aspects or issues that might not have been obvious initially. This phase often leads to generating new ideas and designing great solutions that strengthen the overall quality and efficiency of the software. The comprehensive analysis phase acts as a strong foundation for the development, a more successful and efficient software creation process, helping the team work together.

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

Structured Approach: One of the main advantages of having a test plan is to provide a structured approach to testing. This outlines the scope, objectives, resources, timelines and activities performed in the testing process. A structured approach allows the test team to systematically identify and resolve problems, comprehensively understand the scope of testing, and minimize the risk of missing critical features or scenarios.  
  
Risk Management: Test plans help identify potential risks and challenges early in the project lifecycle. Analyzing requirements and project restrictions, the test team can anticipate possible problems and plan response strategies accordingly. This aggressive risk management approach helps teams effectively address unexpected challenges and minimizes the likelihood of project delays or failures. The test plan also includes a readiness plan to respond to unexpected situations.

Effective Communication: A test plan is a central document for clear understanding and collaboration among different stakeholders, which describes test objectives and strategies to share a vision and enable seamless interaction among team members, developers, project managers, and customers.

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

Review: All team members checked header files and configuration files together to identify functions that needed implementation for the program to operate normally. They also pinpointed areas with potential for improvement or problems that needed addressing.

Analysis: By collecting data supported by the software and analyzing issues, we were able to identify components of the software that function normally, those that operate abnormally, and areas that require improvement.

Collaboration: During the meeting, each member actively participated in discussing the project problems, sharing their insights, and proposing potential solutions. This collaborative effort not only facilitated problem-solving but also fostered a sense of teamwork and shared responsibility among all team members.