# **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**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1. Leo Ru | 4. Frank Fu |
| 2. WaiSun Lam | 5. |
| 3.Xinyang Ma | 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** |
| **XinYang Ma**  **Frank Fu** | **Analysis of the application and create new header file** |  |
| **Wai Sun Lam**  **Leo Ru** | **Test Plan** |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

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:**

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

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| Test plan discussion | Review and proofread | **Approved** |
| Header file discussion | Review and discussed | **Approved** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Agreed to have more meetings/week | Enhance communication and promotes discussion between group members |
| Prioritize the completion of function specifications | Once function specifications are done, group can then proceed with other tasks |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| XinYang Ma | **Reflection 1a.** | **20min** | **yes** |
| Frank Fu | **Reflection 1b.** | **20min** | **yes** |
| Wai Sun Lam | **Reflection 2** | **15min** | **yes** |
| Leo Ru | **Reflection 3** | **25min** | **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 |
| Frank Fu | A set of function specifications stored in the repository |
| Leo Ru  Wai Sun Lam  XinYang Ma | A set of blackbox tests as test documents with test data for the functions.- Start writing blackbox test code and store in repository. (at least 1 required)  A function-test matrix added to the repository. |
| Leo Ru  Wai Sun Lam  XinYang Ma  Frank Fu | Start implementing functions and store in repository. (optional) |
| XinYang Ma | Updated Jira project to show activities and progress. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Tasks assigned for next milestone/ Completion of scrum report 2 | **Everyone on the same page and knows what needs to be done** |
| Test Plan/Header File approval | **Can proceed to the next step** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| quickly decided on division of labor without conflict | current phase of assignment has clearly outlined deliverables that made it easy for group members to understand the necessary tasks as well as likely workloads involved |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

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

**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 to complete the solution without actually writing the software can be more challenging than just writing the software itself.

Analyzing a problem requires a comprehensive and thorough understanding of the conditions, constraints and objectives of the software. To achieve this, we need to gather as much information about the software requirements as we can, make assumptions and define the scope. After analyzing it, here comes the designing part, which is even more challenging. Designing the software means breaking a big problem into small, manageable parts. But to make the software work smoothly and to avoid potential problems, we first need to define the relationships between each part and also define the boundary or scope of them. Analyzing and designing are extremely important as the success of the project depends on these two phases. Once done, decisions can’t be easily changed. Therefore, we need to be very careful.

On the other hand, actually writing the software is just turning the design into actual code. As much as it requires strong technical knowledge, it largely relies on the analysis and design work that has been done before.

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

By doing this, developers gain a thorough understanding of the requirements and goals of the software. By analyzing the software and documenting specifications can identify potential ambiguities or missing requirements early on. This allows for effective communication between team members, stakeholders, and project managers, reducing the risk of misunderstanding. The second advantage is that it improves software design and structure. By having specifications in mind before actual coding of the software enables better design and structure and minimizes probability of modifying software design and structure later in the development process. This systematic approach ensures that the software is well-organized, modular, and maintainable.

1. Why is it a good idea to create a test plan? Describe at least 3 advantages of test plans.  
     
   Any software development method must include test planning. It acts as a template for software testing and aids in the testing process. Here are three major benefits of having a test plan. First, is improved quality assurance. For example, test plans give a systematic approach to testing, ensuring that all critical system components are tested. This aids in the detection of defects, mistakes, and inconsistencies that might otherwise go undiscovered. Because possible faults may be discovered and remedied early, the ultimate result is a higher quality product. In addition, efficiency. To explain, A test plan helps minimize redundancies and assures best resource usage by explicitly outlining what needs to be tested, how, and by whom. Finally, communication and consistency. Such as, all stakeholders, including developers, testers, and project managers, can refer to the test plan as a reference document. This ensures that everyone understands the testing process and expectations, allowing for improved collaboration and communication. It also provides uniformity in testing because all testers will adhere to the same strategy.

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

To analyze and understand the existing software, the first step we took was to read through the project instructions thoroughly, so we have a clear understanding of software requirements. Then we studied the provided documentation and files associated with this project which includes source files and various documentation templates. Next was to review the codes provided in the .c and .h files to gain a deeper understanding of how the software is implemented. Read through source code and pay attention to data structures. The group ran the code to see what the output of the current version of the code is. This allows for a better understanding of what the code currently offers and what needs to be added and implemented. The group documented their findings to share with each other.