# **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**: 5

**Members Present**:

|  |  |
| --- | --- |
| 1. Gabriel Yeung | 4.Sheng Chieh Lin |
| 2.Ali Riza Sevgili | 5.Sheida Hashem Dabbaghian |
| 3.Rong Gang Xu | 6. Hon Kit Mok |

## 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** |
| **Gabriel Yeung** | **Setup github repository** | **0 ( No delay)** |
| **Gabriel Yeung** | **Setup jira board** | **0** |
| **Gabriel Yeung** | **Share git experience** | **0** |
| **Gabriel Yeung** | **Discuss tasks distribution** | **0** |
| **Gabriel Yeung** | **Add teammate to github repo** | **0** |
| **Gabriel Yeung** | **Add teammate to jira board** | **0** |
| **Hon Kit Mok** | **Register in the project git repository and jira board** | **0** |
| **Hon Kit Mok** | **Sign the project contract and contribute in the scrum report** | **0** |
| **Rong Gang Xu** | **Join the github repository** | **0** |
| **Rong Gang Xu** | **Join the jira board project** | **0** |
| **Rong Gang Xu** | **Learn how to use git** | **0** |
| **Rong Gang Xu** | **Learn how to use jira** | **0** |
| **Rong Gang Xu** | **Discuss tasks distribution** | **0** |
| **Ali Riza Sevgili** | **Share git experience** | **0** |
| **Ali Riza Sevgili** | **Discuss tasks distribution** | **0** |
| **Sheng Chieh Lin** | **Join the Github repository and the Jira board project** | **0** |
| **Sheng Chieh Lin** | **Discuss tasks distribution** | **0** |
| **Sheida Hashem Dabbaghian** | **Discuss tasks distribution** | **0** |
| **Sheida Hashem Dabbaghian** | **Join Jira board project and the GitHub repository** | **0** |
| **Sheida Hashem Dabbaghian** | **Learning how to use GitHub** | **0** |

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 discussed in the meeting and the outcomes of the discussions.

|  |  |  |
| --- | --- | --- |
| **Topic** | **Discussion Summary** | **Outcome** |
| Testing plan | **Analysis the program requirements and functionality, documentary the findings** | **Summarized the details in test plan** |
| Header file of data structure | **Discuss the missing data structure required for the program** | **Summarized the data structure in header file** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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** |
| **Everyone should study the source code** | Team members would take turns writing codes in it. |
| **Separate duties** | To complete the project efficiently, team members would take care of different duties. |
| **Task on Jira Board** |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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?** |
| Gabriel Yeung | **Push required source code and documents to git repository.** | **5 min** | **Yes** |
| Gabriel Yeung | **Update Scrum report for week 2** | **30 min** | **No** |
| Gabriel Yeung | **Study week 2 requirement** | **30 min** | **Yes** |
| Gabriel Yeung | **Study source code** | **5 min** | **No** |
| Gabriel Yeung | **Add data structure** | **5 min** | **No** |
| Rong Gang Xu | **Discussion about test plan** | **20 min** | **Yes** |
| Rong Gang Xu | **Update Scrum report for week 2** | **30 min** | **No** |
| Rong Gang Xu | **Study week 2 requirement** | **30 min** | **Yes** |
| Rong Gang Xu | **Study source code** | **5 min** | **No** |
| Hon Kit Mok | **Study week 2 requirement** | **30 min** | **Yes** |
| Hon Kit Mok | **Study the template source code** | **15 min** | **Yes** |
| Hon Kit Mok | **Discuss and prepare the draft of test plan** | **4 hours** | **Yes** |
| Hon Kit Mok | **Update Week 2 scrum report** | **30 min** | **Yes** |
| Ali Riza Sevgili | **Study week 2 requirement**  **Check and get feedback about draft, test plan and other related duties.** | **1 hour** | **Yes** |
| Ali Riza Sevgili | **Update Week 2 scrum report** | **35 min** | **Yes** |
| **Sheng Chieh Lin** | **Discussion about test plan** | **20 min** | **Yes** |
| **Sheng Chieh Lin** | **Update the draft of test plan.** | **40 min** | **Yes** |
| **Sheng Chieh Lin** | **Update Scrum report for week 2** | **20 min** | **Yes** |
| **Sheida Hashem Dabbaghian** | **Study week 2 requirements** | **15 min** | **Yes** |
| **Sheida Hashem Dabbaghian** | **Discuss about the parts of the milestone that we must do individually** | **15 min** | **Yes** |
| **Sheida Hashem Dabbaghian** | **Check the test plan to see if anything is left there** | **40 min** | **Yes** |
| **Sheida Hashem Dabbaghian** | **Update Week 2 scrum report** | **20 min** | **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** |
| **Gabriel Yeung** | Study the source code and share knowledge with teammates |
| **Gabriel Yeung** | Manage git repository |
| **Gabriel Yeung** | Provide better way for teammates to write the scrum report |
| **Gabriel Yeung** | Study test plan |
| **Gabriel Yeung** | Write Data Structure |
| **Rong Gang Xu** | Study the source code and discuss with teammates |
| **Rong Gang Xu** | Update git and jira board |
| **Rong Gang Xu** | Improve the way of writing the scrum report |
| **Rong Gang Xu** | Working on test plan |
| **Hon Kit Mok** | Study the provided source code and discuss with teammates |
| **Hon Kit Mok** | Finalized and touch up the test plan |
| **Hon Kit Mok** | Update the git and jira board |
| **Hon Kit Mok** | Update the scrum report to report the progress of project to teammates and professor |
| **Ali Riza Sevgili** | Contribution to figure the scrum report and study for source code and other modules |
| **Ali Riza Sevgili** | Update any future changes on Jıra and push related files to GIT |
| **Sheng Cheih Lin** | Continuously adjusting scrum reports and updating test plans |
| **Sheng Cheih Lin** | Update git and jira boards and share knowledge with team members. |
| **Sheida Hashem Dabbaghian** | Update jira and scrum report |
| **Sheida Hashem Dabbaghian** | Read all the test plan to see if anything’s left |
| **Sheida Hashem Dabbaghian** | Study more about Git, data structures and test plan |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| **Outcome** | **Impact on Project** |
| Work distribution | **Tasks was specific and to be delivered by team members** |
| Create a Header file | **Prepare the header file of missing data structure to be complete after meeting** |
| Testing Plan | **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 report | **All members contributed** |
| Jira board work distribution | **All members contributed** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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** |
| Data structure header file | **Further information to be confirmed before delivered** |
| Testing plan document | **Only board ideas were discussed during the meeting, further touch up of details to be conduct after the meeting** |
| Git update | **After test plan finalized** |
| Jira board update | **After test plan and scrum report finalized** |
| Work balance | **The workload was not completely balanced for MS2 but for the next one, we can prepare everything and assign the tasks to each member earlier.** |
|  |  |
|  |  |

**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?

The process of analyzing a problem and conceptualizing software functions without directly engaging in the software writing phase can prove more challenging than directly writing the software to complete a project. The absence of actual software implementation introduces the risk of overlooking crucial details, potentially leading to the neglect of important complexities that may arise during development. Furthermore, abstaining from the implementation stage makes it more difficult to grasp the intricacies of the problem at hand.

This approach demands abstract and conceptual thinking to address the problem and envision potential solutions, emphasizing the necessity of a holistic understanding gained through the hands-on experience of software development. Therefore, while planning and design are integral, the direct engagement with the implementation process remains crucial for a comprehensive and effective software development approach.

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

1. Initial analysis and design of software functions lay the foundation for clear requirements and specifications, promoting a shared understanding among team members and stakeholders about the software's objectives. This collaborative clarity enhances communication and ensures everyone is aligned with the project goals.

2. The process of analyzing and designing software in the early stages allows for effective planning and organization of the development effort. It helps identify potential challenges, dependencies, and risks ahead of time, enabling better resource allocation, scheduling, and task prioritization. This proactive approach contributes to a more efficient and well-managed software development lifecycle.

1. Why is it a good idea to create a test plan? Describe at least 3 advantages of test plans.
2. Structured Framework: Testing plans establish a structured and organized framework for testing, guiding testers through a systematic approach. This ensures a methodical examination of all testing aspects, promoting efficiency and clarity in the testing process.
3. Thorough Examination: Test plans assist in identifying and addressing all functional requirements, features, and scenarios that necessitate testing. This comprehensive coverage minimizes the risk of overlooking critical areas, ensuring a thorough examination and validation of the software.
4. Enhanced Communication: Serving as a communication tool, test plans facilitate effective collaboration among diverse stakeholders involved in testing. This includes developers, project managers, and clients, fostering better communication and understanding throughout the testing phase.
5. Describe the process you used to analyze and understand the existing software.

Review requirement: Thorough review of header and configuration files was conducted to grasp implementation details, enabling the identification of potential areas for improvement or issues in the software.

Research: Relevant information related to the software, including insights from analyzed reported issues, was systematically collected. This process provided valuable perspectives on the software's strengths, weaknesses, and areas for enhancement.

Alignment and discussion: Engaging in discussions within the group proved instrumental in clarifying doubts and obtaining additional information. This collaborative effort ensured a holistic understanding of the software's intricacies, supporting informed decisions for refinement and optimization.