# **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**: \_\_\_\_\_\_Group 4\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1. Jo, Eric - 137057188 | 4. Peralta, Joe - 114751209 |
| 2. Nguyen, Huu Linh - 118197227 | 5. Dominguez, Daniel - 110835188 |
| 3. Jang, Hyeri - 115328221 | 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** |
| **Peralta, Joe** | **Data Structures** |  |
| **Nguyen, Huu Linh** | **Test Plan** |  |
| **Jang, Hyeri** | **Test Plan** |  |
| **Jo, Eric** | **Test Plan** |  |
| **Dominguez Daniel** | **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 |
| Task distribution | **One member has been assigned to do the data structures, while others have been assigned to complete the test plan. Each member should answer questions in the test plan.** | **The truck and box structure has been created in its own header file. The test plan has been completed by the remaining members.** |
| Program Analysis | **Every member has been assigned to analyze and understand some of the core concepts and functionality of the program.** | **Each member has reviewed and understand some of the functionality and purpose of the program.** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Dividing parts of Test Plan | Test plan template has 17 articles to make a whole plan |
| Creating data structures is assigned to a member | He was confident of it and appealed his ability to complete it. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Eric | **Test Plan: Deliverables, Criteria, and Dependencies** | **1h** | **Yes** |
| Huu Linh | **Test Plan: Intro, Scope, Strategy, and Test Schedule** | **1h** | **Yes** |
| Hyeri | **Test Plan: Functions, Resources, and procedures** | **1h** | **Yes** |
| Joe | **Analysis and Data Structure creation** | **1h** | **Yes** |
| Daniel | **Test Plan: Risks, Tools, Documentation, Approvals** | **1h** | **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 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Created a basis for the test | **We got to have insights how to execute test.** |
| Good understanding about the given code | **As we communicate and share ideas, all members got great understanding for the given code.** |
| Fair workload assignment | **We all agreed on the fair workload for each member.** |
|  |  |
|  |  |
|  |  |
|  |  |

**Things That Went Well in This Meeting: Communication and interaction within the group was great**

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 |
| Code Analysis | **Each member didn’t hesitate to share his or her insight with others.** |
| Data Structure Creation | **The member assigned to this task did very well.** |
| Test Plan | **The other 4 members took their own parts and contributed to the task thoroughly.** |
|  |  |
|  |  |
|  |  |
|  |  |

**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 without actually writing the software can be more challenging than simply writing the software to complete the project.**  
      **When analyzing a problem, it requires a deep understanding of the problem domain, user requirements, and potential constraints. This involves gathering and analyzing data, identifying patterns, and determining the most effective approach to solving the problem. This analysis phase requires a higher level of critical thinking and problem-solving skills.**  
      **Problem analysis and software design often deal with ambiguous or incomplete information. It requires making assumptions, conducting research, and making informed design choices based on the available information. Dealing with ambiguity and uncertainty requires a flexible mindset and the ability to adapt the design as more information becomes available.**
   2. Describe two advantages of developing software in this manner rather than just moving on to writing the functions without writing specifications first.  
        
      **Improved Efficiency:**

**Developing software specifications provides a clear direction for the development team. It defines the desired functionality, behavior, and performance expectations of the software. With well-defined specifications, developers have a blueprint to follow, reducing ambiguity and the need for rework. This clarity of direction improves efficiency by minimizing misunderstandings and reducing the time spent on trial and error.  
Enhanced Accuracy:**

**Developing software specifications provides an opportunity to validate requirements and ensure they align with the stakeholders' expectations. By involving stakeholders in the specification design phase, any misunderstandings or gaps in requirements can be addressed early on. This leads to more accurate and precise specifications, reducing the likelihood of developing software that does not meet the intended objectives.**

1. Why is it a good idea to create a test plan? Describe at least 3 advantages of test plans.  
     
   **Organizing Testing Efforts: A test plan serves as a roadmap for the testing activities, providing a structured approach to the testing process and allows for prioritizing testing efforts based on factors such as business impact, risk assessment, and critical functionality. It helps allocate resources and time effectively, ensuring that high-priority tests are executed early and any critical defects are identified and resolved promptly.**

**Enhancing the Quality of Testing:**

**A test plan contributes to the overall quality of the testing process by providing a framework for systematic and thorough testing and establishes standardized testing methodologies, processes, and techniques to be followed across the testing team. It ensures consistency in testing practices, reducing variations and improving the reliability of test results.**

**Ensuring Accuracy and Objectivity:**

**A test plan promotes accuracy and objectivity in the testing process, contributing to reliable test results and conclusions and establishes traceability between requirements, test cases, and test results. It ensures that each requirement is adequately covered by test cases and that the test results can be traced back to the original requirements. This traceability enhances the accuracy and reliability of test coverage analysis and defect tracking.**

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

**1. Reviewing Software Documentation: We began by thoroughly reviewing the available software documentation, including requirement specifications, design documents, user manuals, and any other relevant documents. This helped us gain insights into the intended purpose of the software, its features, and the expected behavior.**

**2. Functional Testing: To validate the software's functionality, we performed functional testing. This involved executing different use cases and scenarios to observe how the software responded and whether it met the specified requirements. By testing various input combinations and comparing the actual results against the expected results, we gained insights into the software's behavior and identified any discrepancies or issues.**

**3. Performing Gap Analysis: We compared the observed functionality and behavior of the existing software with the documented requirements and user expectations. This gap analysis helped us identify any inconsistencies, missing features, or areas where the software deviated from the intended specifications.**

**4. Documentation: Throughout the analysis process, we documented our findings, observations, and recommendations. This documentation served as a reference and provided a consolidated view of our understanding of the existing software.**

**By following this comprehensive analysis process, we gained a deep understanding of the existing software's functionality, architecture, and requirements. This knowledge formed the foundation for further activities, such as designing improvements, identifying areas for optimization, or planning for future enhancements.**