# **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. Chang Cui | 4. Xiaofei Xu |
| 2. Lok Yin Tai | 5. Xinyang Wu |
| 3. Siu Man Cheng | 6. Ye Tian |

## 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** |
| Chang Cui | SCRUM Report & Reflections | N/A |
| Lok Yin Tai | Jira Project | N/A |
| Siu Man Cheng | Contract | N/A |
| Xiaofei Xu | SCRUM Report & Reflections | N/A |
| Xinyang Wu | SCRUM Report & Reflections | N/A |
| Ye Tian | Git Repository | 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 |
| Summary of last week | We first summarized the situation of MS1 last week. Each person gave a summary of their individual part. Then, we discussed and provided suggestions for improving other people's parts in future milestones. | We unanimously agreed that each of us put in our maximum effort. |
| Summary of MS1 score | Our group achieved a score of 100/100 in the last milestone. We shared our joy and excitement. | We unanimously agreed that each person should continue to work hard and strive for 100/100 in every milestone. |
| Summary of lecture content | Each of us listened attentively to the professor's lecture. We shared the knowledge we gained from the lecture and the requirements for MS2 to ensure nothing was missed. | After the discussion, we unsurprisingly found that each of us listened to the lecture very seriously and understood the requirements explained by Professor Diab. We exchanged our understanding of MS2 with each other. |
| Discussion on MS2 task allocation | We needed to divide the tasks of MS2 to ensure that the workload and difficulty were roughly evenly distributed among team members, ensuring fairness and sufficient contribution from everyone. | First, we carefully studied each part of the complete tasks required for MS2. We divided the entire milestone into six parts based on workload and difficulty (achieving absolute fairness was not possible, but we aimed for relative fairness). Then, to ensure fairness, we used an online random task allocation software to randomly assign tasks to each person. |
| Individual research on assigned parts | After determining each person's responsibilities, we individually studied our assigned tasks. | We had questions about our own parts, and everyone quickly raised questions and discussed ideas as a group. For any unresolved questions within our group, our team members reached out to Professor Diab for clarification and ensured that before the end of the lecture, every person in our group had no doubts about the overall content of MS2 and their individual parts. |
| Deadline discussion | We determined that the deadline for MS2 is next Monday. | Everyone unanimously agreed that each member should complete their part as early as possible, ensuring enough time for mutual inspection before the deadline to ensure high-quality completion of MS2. |

**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 |
| Divided the tasks of MS2 into six parts based on workload and difficulty | Ensures a balanced distribution of work among team members, taking into account individual capabilities and expertise |
| Assigned tasks using an online random task allocation software | Ensures fairness and impartiality in task distribution, minimizing bias or favoritism |
| Established a goal of achieving a score of 100/100 in every milestone | Promotes a high standard of performance, motivation, and quality throughout the project |
| Clarify the deadline for MS2 as next Monday | Provides a clear timeline for completion and allows sufficient time for reviewing, revising, and finalizing the milestone |
| Emphasized the importance of early completion of individual parts | Provides ample time for collaborative review, feedback, and addressing any potential issues or improvements |
| Decided to consult Professor Diab for clarification on unresolved questions | Ensures a comprehensive understanding of the overall requirements and objectives of MS2, reducing ambiguity and potential errors |
| Created issues in Jira for each task and added them to the "To Do" column | Provides a visual representation of the tasks to be completed and helps in tracking progress |
| Maintained regular updates on work progress in Jira | Enables real-time tracking of individual and overall project progress, promoting transparency and accountability |
| Uploaded completed parts to GitHub and updated progress in Jira | Ensures version control and centralized documentation of completed work, facilitating collaboration and enabling accurate reporting of progress |

**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? |
| Chang Cui | SCRUM Report | 30 mins | Yes |
|  | Reflections | 30 mins | Yes |
|  | Jira Board Management | 10 mins | Yes |
|  | GitHub Repository Management | 10 mins | Yes |
| Lok Yin Tai | Test Plan Control Procedures | 15 mins | Yes |
|  | Test Plan Functions To Be Tested | 15 mins | Yes |
|  | Test Plan Resources and Responsibilities | 15 mins | Yes |
|  | Test Plan Deliverables Session | 15 mins | Yes |
|  | Jira Board Management | 10 mins | Yes |
|  | GitHub Repository Management | 10 mins | Yes |
| Siu Man Cheng | Data structures | 15 mins | Yes |
|  | Test Plan Tools | 15 mins | Yes |
|  | Test Plan Documentation | 15 mins | Yes |
|  | Test Plan Approvals | 15 mins | Yes |
|  | Jira Board Management | 10 mins | Yes |
|  | GitHub Repository Management | 10 mins | Yes |
| Xiaofei Xu | Test Plan Introduction | 20 mins | Yes |
|  | Test Plan Scope | 20 mins | Yes |
|  | Test Plan Test Strategy | 20 mins | Yes |
|  | Jira Board Management | 10 mins | Yes |
|  | GitHub Repository Management | 10 mins | Yes |
| Xinyang Wu | Test Plan Environment Requirements | 20 mins | Yes |
|  | Test Plan Execution Strategy | 20 mins | Yes |
|  | Test Plan Test Schedule | 20 mins | Yes |
|  | Jira Board Management | 10 mins | Yes |
|  | GitHub Repository Management | 10 mins | Yes |
| Ye Tian | Test Plan Suspension / Exit Criteria | 15 mins | Yes |
|  | Test Plan Resumption Criteria | 15 mins | Yes |
|  | Test Plan Dependencies | 15 mins | Yes |
|  | Test Plan Risks | 15 mins | Yes |
|  | Jira Board Management | 10 mins | Yes |
|  | GitHub Repository Management | 10 mins | 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 |
| Chang Cui | * SCRUM Report * Reflections * Jira Board Management * GitHub Repository Management |
| Lok Yin Tai | * Test Plan Control Procedures * Test Plan Functions To Be Tested * Test Plan Resources and Responsibilities * Test Plan Deliverables Session * Jira Board Management * GitHub Repository Management |
| Siu Man Cheng | * Data structures * Test Plan Tools * Test Plan Documentation * Test Plan Approvals * Jira Board Management * GitHub Repository Management |
| Xiaofei Xu | * Test Plan Introduction * Test Plan Scope * Test Plan Test Strategy * Jira Board Management * GitHub Repository Management |
| Xinyang Wu | * Test Plan Environment Requirements * Test Plan Execution Strategy * Test Plan Test Schedule * Jira Board Management * GitHub Repository Management |
| Ye Tian | * Test Plan Suspension / Exit Criteria * Test Plan Resumption Criteria * Test Plan Dependencies * Test Plan Risks * Jira Board Management * GitHub Repository Management |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Thorough understanding of lecture content and MS2 requirements | Ensures alignment and clarity among team members, minimizing potential misunderstandings |
| Tasks for MS2 divided into six parts based on workload and difficulty | Promotes a fair distribution of work, optimizing productivity and individual contributions |
| Utilized an online random task allocation software for task assignment | Ensures an unbiased and impartial task distribution process |
| Promptly addressed questions and concerns regarding individual parts | Fosters a collaborative environment, promotes knowledge sharing, and ensures clarity for successful task execution |
| Decision to strive for a score of 100/100 in every milestone | Establishes a high-performance standard and motivates the team to continuously improve |

**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 |
| Individual part summaries and suggestions | Active participation and constructive feedback from each team member |
| Thoughtful task allocation for MS2 | Careful consideration of workload and difficulty, aiming for relative fairness |
| Proactive clarification of doubts and questions | Willingness to seek clarification and prompt communication with Professor Diab |
| Consensus on striving for 100/100 in every milestone | Shared commitment to excellence and continuous improvement |
| Clear deadline determination for MS2 | Setting a specific timeframe for task completion and milestone achievement |
| Active participation in discussions and collaborative problem-solving | Encouraging an inclusive and engaged team dynamic |

**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 | Everything went very well in the meeting;  we are a great team. |

**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 to complete the project. The main reason for this is that the analysis and design phase requires a deeper understanding of the problem and the desired solution. It involves carefully considering various aspects such as requirements, system architecture, user experience, and potential challenges. By focusing on analysis and design first, developers can ensure that they have a solid plan in place before diving into implementation. This approach helps identify potential issues and design flaws early on, reducing the risk of costly rework and improving the overall efficiency of the development process.

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

Improved clarity and alignment: By thoroughly analyzing the problem and designing software functions beforehand, the development team gains a clear understanding of the project's objectives, requirements, and constraints. This clarity enables better alignment between stakeholders, including developers, project managers, and clients. It reduces misunderstandings, promotes effective communication, and ensures that everyone is on the same page regarding the desired outcome.

Efficient resource utilization: Investing time in analysis and design upfront helps identify potential challenges and bottlenecks early on. It allows for better planning and allocation of resources, including time, budget, and personnel. By addressing potential issues at the design stage, developers can optimize the development process, minimize rework, and avoid wasting resources on implementing functionalities that may not align with the project's goals. This approach leads to greater efficiency and overall cost savings throughout the software development lifecycle.

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

Creating a test plan is highly beneficial for software testing for several reasons. Here are three advantages of having a test plan:

Provides Clarity and Structure: A test plan serves as an instruction manual for the testing process, offering a clear and structured approach. It outlines the objectives, scope, and schedule of the testing, ensuring that all stakeholders have a common understanding of what needs to be tested and how. This clarity helps avoid confusion and misunderstandings among the testing team and other project members, leading to more efficient and effective testing.

Risk Mitigation: Test plans include a list of risks associated with the project and propose strategies for mitigating them. By identifying potential risks early on, the test plan enables proactive measures to address them. It helps in identifying critical areas, dependencies, and vulnerabilities that may impact the software's quality and functionality. With this insight, the testing team can focus their efforts on critical areas and allocate resources appropriately, reducing the chances of major issues in the software.

Resource Planning and Management: A test plan includes a list of hardware and software resources required for testing, along with the roles and skills of the testing team members. This allows for effective resource planning and management, ensuring that the necessary tools, infrastructure, and expertise are available when needed. By clearly defining responsibilities and expectations, the test plan facilitates coordination and collaboration among team members, enabling them to work together towards successful testing.

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

In order to analyze and understand the existing software, our team followed a systematic process that involved thorough examination and investigation. The process can be summarized as follows:

Reviewing project documentation: We carefully studied the project documentation, including requirements specifications and any other relevant materials. This allowed us to gain an initial understanding of the software's purpose, functionality, and intended use.

Analyzing existing code: We conducted a detailed analysis of the software's codebase. This involved examining the code structure, modules, classes, and functions to understand the overall architecture and implementation details. By reviewing the code, we identified key components and their relationships, which helped us grasp the software's internal workings.

Identifying data structures: We paid close attention to the data structures used in the software. By analyzing how data was stored, manipulated, and accessed, we gained insights into the software's data management and processing capabilities. This analysis enabled us to understand the flow of information within the software.

By following this process, our team was able to gain a comprehensive understanding of the existing software, its functionality, and its alignment with the project requirements. This analysis formed the basis for further steps, such as designing test cases and executing the testing process effectively.