# **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.Wilson sum | 4.Sasawat Yimleang |
| 2.Lebna Noori | 5. Mostafa Hasanalipourshahrabadi |
| 3.Samin Sorayya | 6.Radmehr Behzadfar |

## 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** |
| **Wilson Sum** | **Added 13 Tests, Filled out 3 Questions on Test Plan(Q1,Q2,Q3), Updated Jira, Committed files to github, Attended all meetings** | **MS2 MeetUp To fill forms** |
| **Lebna Noori** | **Filled out 3 questions on plan test (Q15,16,17), Reflection Q2. Updated Jira, Committed files to github** | **N/A** |
| **Samin Sorayya** | * **Completed three questions on the Test Plan (Q4, Q5, Q6)** * **Updated Jira with necessary information and progress updates** * **Uploaded files to GitHub for version control and teamwork** * **Attended all meetings related to testing** * **Reviewed code shared by Mostafa and Wilson** | **N/A** |
| **Sasawat Yimleang** | **Filled out 3 questions on Test Plan (Q10, Q11, Q12), Updates Jira, committed files to GitHub, attended all meeting, Review all source code files shared by Mostafa and Wilson** | **N/A** |
| **Mostafa H.** | **Developed validation header file and new structures, Worked on source files, Completed and did the final editing of scrum report (developing and editing initial reflection questions, completed meeting tables), Completing sections 7, 8, and 9 of test plan, Updated Jira, Attended all meetings, committed changes to Github, Did the final editing of deliverables** | **N/A** |
| **Radmehr Behzazdfar** | **Attended all the meeting on the planned time.**  **Updated Jira with the latest tastings and information.**  **Filled out the questions 13 and 14 which was assigned by the group.**  **Committed and push the files related to the part on the github.** | **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** | **MS2 MeetUp To fill forms** |
| **Reason for delay or block** | **Misunderstandings and schedule differences** |
| **Impact on Project** | **1 day late start** |
| **Solution or work-around** | **Had a new meeting.** |
|  |  |
| **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 |
| GitHub | **Gave a small intro to learn Github, shared links to learn Github and assigned tasks to learn Github to members who weren’t completely sure how to use it.** | So far about 3 of us know how to use github and there is a due date on when all members are required to know the basics of it. |
| Assigning test plan parts | **As for the test plan, the work was decided to be assigned to all members.** | **Each member complete do 2-3 sections.** |
| Explanation of MS2 Testing | **Briefly explained the testing process for testing at current stage.** | **Group members now have a general idea on how to test the program** |
| Project deliverables | **Discussed about the need of developing new structures required by the project** | **Mostafa proposed to work on the new header file related to validation process** |
| Project specifications | **Analyzed the project thoroughly in the meeting to help grasp the idea Helped of what needs to be done through the milestone** | **We made sure everyone is on the right track at this stage of the project** |
| Ms3 details | **Went through ms3 in the meeting and read the details to be prepared ahead and developing our current work based on that** | **Mostafa got clue to write new structures through the reading details of ms3** |
|  |  |  |

**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 |
| All members are to learn how to use GitHub | This will make doing the project as a team much easier since we will be able to all work on one project. Also, this is a requirement. |
| Splitting Test Plan Questions | The test plan is quite long and detailed therefore it’s best if we split the test plan amongst the members. |
| Identify and prioritize critical tasks | In this phase, we first identified being able to work with Jira and Git as the most important part of the project and made sure everyone can work smoothly with these platforms. Prioritizing critical tasks will ensure that time-sensitive items are addressed promptly. |
| Meeting time problem encountered | Encourage Active meeting participation. During the last two weeks, we held in person meetings right after the class, ignoring the need to give group members enough time to fully grasp the idea of the milestone and analyze the project. This led to problems that caused the change in our approach to the meetings. |
| Working in Jira | Being able to work and contribute in Jira was identified as a necessity in organizing our works, so it was addressed in first two days after the lab. |
|  |  |
|  |  |

**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? |
| Wilson Sum | * **Analyze the instruction, task, and the obstacle of the project.** * **Filling out Test Plan Questions.** * **Completed the SCRUM report by updating the task status, including progress and any obstacles encountered.** * **Create the test cases for the structure in mapping.c file.** * **Collaborated with Sasawat, Mostafa, and Samin to review the test cases and new data structures created in the new header file.** | **60 mins** | **Delayed because we needed to further understand our assignment.** |
| Lebna Noori | * **Filling out Test Plan Questions.** * **Completed the SCRUM report by updating the task status, including progress and any obstacles encountered.** | **1 hr** | **Delayed because we needed to further understand our assignment.** |
| Samin Sorayya | * **Analyze the instruction, task, and the obstacle of the project.** * **Filling out Test Plan Questions.** * **Completed the SCRUM report by updating the task status, including progress and any obstacles encountered.** * **Assisted Sasawat, Mostafa, and Wilson in reviewing the test cases and new data structures created in the new header file.** | **45 mins** | **Delayed because we needed to further understand our assignment.** |
| Sasawat Yimleang | * **Analyze the instruction, task, and the obstacle of the project.** * **Filling out Test Plan Questions.** * **Completed the SCRUM report by updating the task status, including progress and any obstacles encountered.** * **Assisted Wilson, Mostafa, and Samin in reviewing the test cases and new data structures created in the new header file.** | **45 mins** | **Delayed because we needed to further understand our assignment.** |
| Mostafa H. | * **Discussed the need for developing the new structure and a header file with other group members. Shared my code I had worked on it to other members and asked their opinion (finally was assigned to develop the header file and commit it in GitHub)** * **Analyzed the specification of MS2 and MS3 in the meeting.** * **Discussed different sections of test plan and developed our idea of approaching to different parts of the test plan.** * **Discussed some weaknesses in our teamwork.** * **Collaborated with Wilson to review and develop the test cases.** * **Completing scrum reports based on previous experience.** * **Worked on the different sections of test plan.** | **80 min** |  |
| Radmehr Behzazdfar | * **Filling out Test Plan Questions.** * **Completed the SCRUM report by updating the task status, including progress and any obstacles encountered.** | **1 hr** |  |
|  |  |  |  |

**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 |
| Samin Sorayya | Partial Coding, scrum report. |
| Lebna Noori | scrum report. |
| Wilson Sum | Partial Coding, scrum report. |
| Sasawat Yimleang | Partial Coding, scrum report. |
| Mostafa Hasanalipourshahrabadi | Partial Coding, scrum report. |
| Radmehr BehzadFar | scrum report. |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Assigned roles and responsibilities on test plan | **The test plan will be split amongst members to reduce the workload on each other.** |
| Further Analysis of problem. | **A better understanding of the assignment which will help us make our approach to the assignment.** |
| Agreed upon project deliverables | **We achieved clear expectations and goals for the project.** |
| Changes in in-person meetings settings | **Prioritize the analyzing the problem before trying to work would lead to achieve a better outcome.** |
|  |  |

**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 |
| Understanding of the project. | **Communication channel established on the first week between members allowed for immediate discussion between members, and making sure everyone is on the right track.** |
| Task allocation in test plan | **Clear assignment of tasks to team members.**  **• The meeting ensured that each task was allocated to a responsible team member.**  **• Clear roles and responsibilities were defined, avoiding confusion or duplication of efforts.** |
| Duration of meeting | **The second meeting that held online via Zoon application didn’t take long, as members were fully aware of what needs to be discussed about and what are the main obstacles.** |
| Meeting environment | **Having open and constructive discussions, enabled the team to explore different viewpoints** |
| Participation | **Everyone had an opportunity to express their opinions and perspectives** |

**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 |
| Meeting time setting | **Not everyone was able to attend.**   * **Vote for a time that suits everyone.** |
| Meeting Agenda | **The first meeting held right after the class, group members were confused and had no idea on what to talk about.**   * **Give group members enough time to analyze the problem individually, grasp it and them set the meeting, this way everyone knows what is necessary to discuss on** |
| Task Prioritization | **Lack of clarity and understanding of project deliverables.**   * **Request clear guidelines on milestones and ask as many questions as we could in class time.** |
|  |  |

**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?  
        
      This process of analyzing a problem and designing software functions without actually writing the software can be both more difficult and easier compared to just writing the software itself. Let me explain why. On one hand, it can be more difficult because it requires a deep understanding of the problem and the system involved. You need to carefully analyze the problem, think about different scenarios and possible special cases, and design the software functions to address those requirements. It takes a lot of thought and planning to make sure the functions do what they're supposed to and meet the project's goals. It's like solving a puzzle where we have to consider many factors and potential trade-offs. On the other hand, it can be easier because it allows for a more focused and deliberate approach. By analyzing the problem and designing the software functions upfront, without the pressure of writing the actual code, we have the opportunity to explore different solutions and consider potential issues before diving into implementation. It helps us gain a clearer understanding of the problem and the things we need to achieve, which can make the actual coding process more efficient. It also allows you to get feedback from others involved in the project, which can help improve the design and catch any problems early on. In summary, this process of analyzing and designing software functions without writing the code can be challenging because it requires a deep understanding and careful decision-making. However, it also offers the advantage of thorough planning and consideration, which can lead to better software development outcomes.

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

When we take the time to plan and design software functions before diving into writing code, it brings us two important advantages. First, it gives us the opportunity to prepare ourselves and think through the problem we're trying to solve. By analyzing the problem carefully and considering different solutions, we can come up with a clear plan of action. This helps us avoid getting overwhelmed and ensures that we take things one step at a time. Instead of rushing into coding and making a mess, we can approach the problem in a more organized and manageable way. Second, going through the planning and design phase helps us understand the problem better. By studying the requirements and thinking about potential challenges or tricky situations, we gain a deeper understanding of what we're trying to achieve. This understanding is valuable because it helps us avoid mistakes and prevents us from having to go back and redo our work. When we have a clear picture of what needs to be done, we can focus on implementing the right solutions from the start. In a nutshell, taking the time to analyze and design software functions before writing code gives us the advantages of being prepared and having a solid understanding of the problem. It makes the development process smoother, reduces the chances of errors, and helps us create effective solutions. By approaching the problem in a thoughtful and well-planned manner, we set ourselves up for success.

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

As it helps in guaranteeing the quality and stability of a software system, creating a test plan is a very important stage in the software creation process. The three important advantages of having a thorough test plan are:

* + 1. - A test plan describes the test cases that need to be run as well as its goals and scope. It helps in locating and documenting the exact features, functionalities, and scenarios that need to be tested. A test plan guarantees that every aspect of the software is carefully checked, increasing the overall test coverage by providing an in-depth roadmap or idea for testing. It also helps in finding more bugs and potential problems, that could result in a product of high quality.
    2. A test plan helps in allocation of resources efficiency by detailing the testing activities, deadlines, and relationships. Based on how important the features or areas under the test are, it allows teams or organizations to prioritize their testing efforts. This guarantees that testing efforts are concentrated on the software's most important features, maximizing its effectiveness and efficiency.
    3. A test plan helps in the identification of possible bugs and challenges related to the software system, such as functional errors, performance difficulties, or weaknesses in security. Teams or the organization could consistently address these risks and create relevant test cases and methods to catch them by having a well-defined plan in hand. This guarantees that the program is stable and trustworthy while decreasing the possibility of serious problems that could go unnoticed.

In general, developing a test plan gives the testing process a stable structure and direction that can be trusted, while providing smooth communication, and collaboration between those who are involved. It increases test coverage, improves the use of resources, and helps in reducing risks, eventually resulting in a software application of higher quality.

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

To analyze and understand this project, I carefully read and studied the problem statement provided. This allowed me to gain a clear understanding of the problem the software is intended to solve and the requirements it needs to fulfill. It provided the context for my analysis and helped me establish the goals of the software. Then, I dove into the codebase and meticulously examined the code. Line by line, I analyzed the code to understand its structure, logic, and implementation. By doing so, I familiarized myself with the code's organization, the variables used, and the functions implemented. As I went through the code, I identified the key components of the software. This included structures, data types, and functions that played crucial roles in the program's execution. By recognizing these components, I gained insights into the overall architecture and the relationships between different parts of the code. I focused on understanding the data structures employed in the software. This involved examining structures such as Map, Point, Box, and Truck. I studied their attributes, how they were utilized, and their interdependencies. Understanding the data structures helped me grasp how information was stored, manipulated, and shared within the software. By following this process of reviewing the problem statement, examining the code, and analyzing data structures, I was able to gain a comprehensive understanding of the existing project. This understanding formed the basis for identifying potential improvements, designing new structures, and planning the software development process.