# **MILESTONE 2** -- SFT221 Scrum Report and Reflection

All students are expected to attend the SCRUM meetings and to participate. Failure to do so will result in greatly reduced grades.

**GROUP**: **WINTER24-SFT221-NAA-1**

**Members Present**:

|  |  |
| --- | --- |
| 1.Duc Phu Nguyen | 4.harmanpreet singh |
| 2. Anthony Korepanov | 5. karanbeer chanana |
| 3. Jashandeep Singh | 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 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) | 25% |
| Test plan (complete, well-written) | 25% |
| Git usage (used properly with good structure) | 10% |
| Jira usage (creates issues, tracks progress) | 20% |
| Scrum report & reflections | 20% |
| **Deadline** | 20% deduction for each day you are late |  |

**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** |
| Duc Phu Nguyen | -create test plan in Jira and assign member to pick up tasks  -Complete test plan with 3 parts Introduction, scope, test strategy |  |
| Anthony Korepanov | -Programming header file for data structure  -Complete resumption criteria, dependencies, tools, control procedures for test-plan. |  |
| Karanbeer chanana | -complete 3 parts execution strategy, environment requirements and deliverables.  - helped doing the reflection questions  -communicated with other team members to help and get their help |  |
| Harmanpreet singh | Completed the reflection 1 part b  In test file done with test schedule, resource & responsibility and function to be tested. |  |
| **Jashandeep Singh** | **-completed 4 patrys of test plan** Suspension/Exit Criteria, **Risks,** Documentation, Approvals  -helped in question 2 in reflection |  |
|  |  |  |
|  |  |  |

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** | **Delayed tasked** |
| **Reason for delay or block** | Nobody attended in 2 meeting. |
| **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 |
| Git Version Control | How to pull project. How to add and commit changes to the project. How to push changes to repository. | Source code files, data structure header file pushed to repository |
| Define Data Structures | Analyze and discuss project, determine what data structures would fit into the project | Two new data structures added to prototype |
| Use of Jira | Discussed the appropriate use of jira and right use of tiles in it | Everybody used jira efficiently |
| Test plan | The group collaborated to develop a thorough test strategy for the undertaking. They examined a test plan template that was offered and modified it to fit the unique needs and features of the product. | Each participant did their best to complete the test plan and comprehended the code. |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Create Package info data struct**u**re | Analyzing the output prompt for user input in the project. Determining what data is required to be input by the user. |
| Analyzed the code and parts in test plan carefully | - Verify the problems and the requirements by going over the problem statement.  -Looked for areas where code may be changed or improved for better functioning.  -Collaborate with the group to determine the main aspects to concentrate on and the testing scope. And all the parts |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Anthony Korepanov | Reflection question #3. Meetings for data structure.  Filled meeting tables | **1h** | **Yes** |
| Karanbeer chanana | Reflection ques 2. helped filling meeting tables and test plan template | **45 min** | **yes** |
| harman | Reflection 1 part b | 15 min | **yes** |
| Jashandeep | **Did reflection ques2 with a fellow team member, updated the progress of assigned task on jira** | **33 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 |
| Anthony Korepanov | C programmer. Design functions. |
| All | Git control |
| All | Scrum report and reflection |
| Jashandeep | Will write test to check if the function work as expected |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Implemented data header file | Organize project data for later implementations of functions |
| Analyzed test plan carefully | Better understanding of test plan |
| Project successfully coordinated | Initial stages completed and ready for further development |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| All group members present and complete assigned roles | Frequent communication and monitoring of project goals. Properly documenting tasks on Jira and assigning a member to each available task. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 on Sunday at 9am | **Nobody attended. Unacceptance. disrespected people. I am waiting 26 mins on WhatsApp and design to leave.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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?  
      To analyze a problem and design software (functions) without writing the code can be more challenge because there are some reasons
      1. Abstraction – what we need to do and what we don't need to focus on to ensure Time management to deliver project on time
      2. Communication is a biggest challenge because we need to have clear ideas to team member and making sure everyone on same page
      3. Testing at this point is more difficult because we are not writing the code. We need to consider how to test function
   2. Describe two advantages of developing software in this manner rather than just moving on to writing the functions without writing specifications first.  
      (i) First of all, it makes software development easier and systematic. The development process becomes more methodical and deliberate when the problem is carefully examined and the software is thoughtfully designed. This can lessen the possibility of mistakes and problems later on because the software will have been created with a clear grasp of the issue it is meant to address.

(ii) developing software in such a way helps in collaboration among teammates. Team members can understand their roles and duties more clearly if the functionalities and needs of the software are defined in detail. Moreover, Teams can spot any issues early in the process by designing software functionalities before creating code. By using preventive methods, this proactive risk assessment reduces the likelihood of mistakes and project setbacks.

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

The first step in ensuring the dependability, use, and efficiency of your software product is to create a test plan. The following are the three key benefits of having a test strategy:

Quality Assurance: To make sure that modules and functionalities operate as intended, a test plan describes all the testing procedures to be followed, along with the tests for each one.

Risk reduction: Test procedures help detect and neutralize such hazards early on. These could have to do with the functionality, security, performance, or user interface of the product.

Efficiency and resource management: Test strategies can help to streamline and accelerate the testing process. They assist in establishing the exam's criteria, locating the required materials, and projecting how long the examination will take.

Effective Communication: Test plans act as an effective means of communication among various stakeholders involved in the testing phase, such as developers, project managers and clients. By fostering collaboration and mutual understanding, within the team they foster effective testing practices.

Overall, the implementation of test plans is essential to ensure the quality, reliability, and functionality of software products.

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

The process for analyzing existing software involves breaking down the project into separate components and understanding the flow and execution of the software. To start, we examine the data structures of the project to understand the information involved in the software. Afterwards, we analyze the functions of the software and how they interact with the data structure. Examining the functions allows us to see the behavior of the software and outcome of different manipulations on data. Next, we analyze the input stream of the software, which involves the information it prompts the user to input. Using this we can see what information the software requires from the user to execute its functionality. Finally, we examine the output of the software and the outcome of the execution. This way we can see how the data was manipulated, and the product the software has generated because of its execution. This can be compared to the objectives of the project to see if the software has met those expectations. After determining this, the software can be evaluated in terms of its process efficiency.