# **MILESTONE 2** -- SFT221 SCRUM Report and Reflections

This report should be completed in class and submitted at the end of class. Late submissions cannot be accepted without prior approval of the instructor.

**GROUP**: 02

**Members Present**:

|  |  |
| --- | --- |
| 1.Preet Chakrani | 4.Jeny Prakashbhai Rangani |
| 2.Yagnik Dhankara | 5.Sakshi Sakshi |
| 3.Niyatiben Narendarbhai Patel | 6. Harjovan Singh |

## 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** |
| Harjovan Singh | Declared Structures in test.h after analyzing the project problem | **N/A** |
| Niyatiben Patel (Technical lead), Sakshi Sakshi (Project Manager) | Updating GitHub repository regularly by merging the branches and assigning tasks on jira to each team member. | **N/A** |
| Niyatiben Patel, Sakshi Sakshi, Jeny Rangani, Preet Chakrani, Yagnik Dankhara, Harjovan Singh | Make a scrum report including answering reflection questions | **N/A** |
| Niyatiben Patel, Sakshi Sakshi, Jeny Rangani, Preet Chakrani, Yagnik Dankhara, Harjovan Singh | Create a detailed test plan for the project | **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** |  |
| **Impact on Project** |  |
| **Solution or work-around** |  |
|  |  |
| **Delayed or Blocked Task** | **N/A** |
| **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 |
| Understanding the requirements of the milestone | **Examining each milestone requirement and discussing important points and tasks and learning about each other’s strengths and weaknesses in terms of technical knowledge.** | Increased familiarity among teammates and a better understanding of the group project. |
| Assigning tasks | **Delegating each member, a specific task based on their interests and abilities** | **Proper distribution of workload** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
|  |  |
| we will make a test plan first and then according to that proceed further to do the group work by thoroughly understanding the requirements | -proceed step by step and collaborate with other team members  -follow the plan and help the other team member if they face any kind of problem. |
| Ensuring the timely completion of scrum report – Preet Chakrani, Yagnik Dankhara, Jeny Rangani, Harjovan Singh | The decision was made collectively on the basis of mutual understanding and depending on the selected team member’s exceptional abilities |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Niyatiben Patel, Sakshi Sakshi, Jeny Rangani, Preet Chakrani, Yagnik Dankhara, Harjovan Singh | **Deep analysis of the problem and each team member suggested which data structures should be added.** | **40 minutes** | **Yes** |
| Niyatiben Patel, Sakshi Sakshi, Jeny Rangani, Preet Chakrani, Yagnik Dankhara, Harjovan Singh | Completion of all information required to fill in tables in the scrum report and answer reflection questions. | **1 hour** | **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 |
| Niyatiben Patel | Maintain Github repository |
| Sakshi Sakshi | Maintain jira account and assign tasks to team members |
| Jeny Rangani | Complete Scrum Report |
| Preet Chakrani | Write Test Code |
| Yagnik Dankhara | Answer Reflection questions |
| Harjovan Singh | Write Test documents |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Increased familiarity among teammates and better understanding of the group project. | **Better time management and increased productivity of team members.** |
| Proper distribution of workload | **Timely submission of proper and well-thought-out milestone 1.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Equal distribution and delegation of tasks required to be completed | Based on each team member’s abilities and interests and with mutual understanding. |
| Proper analysis of the milestone | **Active participation in group discussions and proper communication among team members.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 | **N/A** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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?

**ANS)** Because it necessitates a thorough comprehension of the issue, the user's requirements, and the limitations of the system, designing software without creating the code is more challenging than simply writing the software. Careful planning, modeling, and documentation of the system's architecture, parts, and interactions are essential components of effective software design. This procedure contributes to ensuring the software's effectiveness, maintainability, and scalability. It is also necessary for stakeholders, developers, and designers to work together on software design, which can be difficult if they have different perspectives and priorities. A well-designed software system can ultimately save time, money, and effort, making the design phase of software development crucial.

* 1. Describe two advantages of developing software in this manner rather than just moving on to writing the functions without writing specifications first.  
       
     **1) Understanding the Problem and User Needs:** Designing software before writing the code allows for a clear understanding of the problem that the software is intended to solve. This includes understanding the needs of the users who will be using the software. By understanding the problem and user needs, the software can be designed to effectively address these needs.

**2)Planning and Modeling:** Once the problem and user needs have been understood, the next step is to plan and model the software. This involves creating diagrams, flowcharts, and other visual aids to help understand how the software will work. This stage helps identify any potential issues with the software before any code has been written.  
  
**3) Saving Time, Money, and Effort:** Although designing software before writing the code may seem like an extra step, it actually saves time, money, and effort in the long run. By identifying any potential issues early on in the design process, the development team can avoid costly mistakes that would require significant time and effort to fix later on.

1. Why is it a good idea to create a test plan? Describe at least 3 advantages of test plans.  
     
   **Ensures Comprehensive Testing:** A test plan lays out a systematic approach to testing and ensures that all aspects of the software are thoroughly tested. It includes test cases, test scenarios, and test data, which helps in identifying and fixing defects early in the development cycle.

**Improves Communication:** A test plan clearly defines the testing objectives, scope, and timelines, which helps in communicating the testing requirements to all stakeholders. It also helps in identifying potential risks and mitigation strategies, which can be communicated to the development team.

**Saves Time and Resources:** A well-planned test plan can help in identifying defects early in the development cycle, which reduces the overall cost of development. It also saves time by streamlining the testing process and avoiding duplicate tests.

**Collaboration**:- is an essential aspect of software development, and test plans can help promote it. Test plans allow multiple team members to contribute to the testing process, share their findings, and work together to resolve issues. This collaborative approach can help ensure that all aspects of the software are thoroughly tested and that any defects are identified and addressed early on in the development process. By promoting collaboration, test plans can also help improve team communication and overall efficiency, leading to a better end product.

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

When it comes to promoting collaboration among team members, a test plan can act as a roadmap for everyone involved in the testing process. By outlining the scope of the testing effort, team members can better understand their roles and responsibilities, and work together more effectively to achieve the desired outcomes. Additionally, a test plan can provide a framework for reporting and tracking defects, ensuring that issues are addressed in a timely and consistent manner.

As for analyzing existing software, the process typically involves a combination of manual and automated techniques. Reviewing documentation can help provide context and identify potential areas of concern, while a code review can help identify issues related to coding standards, security, and performance. Using tools for analysis, such as static code analysis or dependency analysis, can help identify potential issues that may not be immediately apparent. Finally, running the software through various test cases can help identify areas where the software may be vulnerable to defects or may not be meeting functional or performance requirements. By taking a comprehensive approach to software analysis, teams can gain a deeper understanding of the software and identify areas for improvement.