# **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**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_E\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1. Jay Vijaykumar Vakil | 4. Srujal Patel |
| 2. Mohit Sheth | 5. Viren Vaishnav |
| 3. Harsh Dugar | 6. Hetav Mamtora |

## 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** |
| **Jay Vakil** | **Made a private repository followed by creating jira account and lastly creating a project on kanban along with adding team members.** | **NULL** |
| **Mohit Sheth** | **Created my GitHub account using Seneca’s email ID and created my Jira account for project management. Along with that I researched how Jira is used as a software project management tool and what are the advantages of using Jira. This helped me in answering the 2nd question of the Reflection.** | **NULL** |
| **Harsh Dugar** | **Created a Jira account and set up my github account. Moreover, maintained a good communication in the group about the tasks.** | **NULL** |
| **Viren Vaishnav** | **By setting up GitHub and Jira accounts, extensively studying Jira's features, and fostering strong communication within the team, I facilitated smooth project management and effective collaboration.** | **NULL** |
| **Hetav Mamtora** | **Created GitHub and Jira accounts, conducted thorough research on Jira's functionality, and maintained excellent communication within the team to ensure efficient project management and coordination.** | **Null** |
| **Srujal Patel** | **Created a jira account and explored the git to look for it’s advantages. Also attended the group meetings for the better creation of our project.** | **NULL** |
|  |  |  |

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 discussed in the meeting and the outcomes of the discussions.

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| How the tasks will be divided |  | **The activity will be divided according to the extent of the milestone.** |
| Due to submit project files to the group leader | **1 Day Before The Group Assignment Is Due.** | **Everybody Agreed Upon this rule.** |
| How Will the upcoming milestones will be done? | **Meeting platforms, reading and understanding the tasks,etc** | **Will follow a procedure**  **Meeting -> Analysing -> implementing -> reviewing -> submitting.** |
| Project Setup | **How will the project setted up, how will the resources be divided** | **Setup of repository project** |
| Deadline Settlements | **When will team submit their parts, how will they sumbit** | **Project flow of work was arranged** |
|  |  |  |
|  |  |  |

**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 |
| Error Handling and Logging: | It was decided to develop an extensive error handling mechanism and logging system. |
| Deployment Strategy | The team deliberated on the best deployment strategy for the software and decided to adopt a containerization approach using Github |
| Collaboration Tools |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Jay Vakil | **As a team leader I helped with distributing the tasks and collaging the work done by other members of the group. Also, helped in understanding the usage of Jira and Git to the other members of the group.** | **1 hour** | **Yes** |
| Mohit Sheth | **Participated in the meeting to gather information about the project what my task is, understood some basics of Jira from the team leader and documented the discussion held in the meeting which help to create the SCRUM report easilly** |  |  |
| Harsh Dugar | **Researched about the topics all the members along with me need to revise and study in order to understand and complete the milestone.** | **40 mins** | **Yes** |
| Viren Vaishnav | **I took on the critical responsibility of conducting research to provide valuable insight into the team's decision-making process. Actively engaged in research. I diligently searched for relevant data and carefully evaluated different options. My main goal was to identify possible options and justify their consideration. By thoroughly evaluating the potential impacts of each option, my goal was to provide the team with the information they need to make informed decisions.** |  |  |
| Srujal Patel |  |  |  |
| Hetav Mamtora | **Researched software as well as made other colleagues updated with working of each and when used. My key role at the SCRUM meeting last week was to undertake extensive research on various software applications relevant to the project needs. I committed myself to learning about their features and functions and supporting other team members in understanding their purpose and application.** |  |  |
|  |  |  |  |

**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 |
| Jay Vakil | Test plan, Database |
| Mohit Sheth | Test Plan, Database |
| Viren Vaishnav | Database and extensive research on how to make this project work. |
| Harsh Dugar | Database and extensive research on how to make this project work. |
| Srujal Patel | Database and extensive research on how to make this project work. |
| Hetav Mamtora | Database and extensive research on how to make this project work. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Task Tracking | **To keep track of the development and status of each individual work, the team put in place a task tracking system. This choice was chosen to promote responsibility and openness among the team members, allowing for efficient coordination and on-time fulfilment of project deliverables.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Better Understanding | **As some of our team members were using github for the first time. By researching and forced implementation lead to them learning and understanding github and git better.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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?  
        
      It might be difficult to design software without writing any code. Start with completely comprehending the issue and outlining the requirements in order to approach this assignment. Divide the issue into manageable jobs, defining each function's input, output, and behaviour. Choose the most effective solutions after determining the necessary data structures. Create high-level logic and algorithms to implement the needed functionality while optimising performance. Components, modules, and interfaces should all be included in the system architectural design. Consider situations including error handling and exceptions as well as security and privacy issues. Think about the performance requirements, then suggest improvements. Make sure to provide illustrations and explanations with your design selections. Obtain input from key players, then iterate as required. This strategy, albeit demanding, enables meticulous planning, abstract thought, and thorough design prior to execution.
   2. Describe two advantages of developing software in this manner rather than just moving on to writing the functions without writing specifications first.  
        
        
      There are two important benefits to creating specifications before developing software. First off, it offers accuracy and clarity. A greater grasp of the needs may be achieved by analysing the issue and creating specifications, which improves communication with stakeholders. A clear set of specs acts as a point of reference, preventing misunderstandings and guaranteeing the functionality required. Second, it improves efficacy and cost-efficiency. Early design defect detection allows for resolution at a simpler and less expensive stage. With this method, development is streamlined, rework is avoided, and time and money are conserved. Spending effort up front creating specs helps software development become more focused and in line with the intended goal, making the process more successful and efficient as a whole.
2. Why is it a good idea to create a test plan? Describe at least 3 advantages of test plans.  
     
   A test plan provides a well-organized framework for performing testing operations. In order to ensure that every component of the programme is completely tested, it helps in defining the test objectives, test techniques, and test scope. Random and uncontrolled testing techniques can be minimised by following a predefined plan that organises and speeds the testing activities.

By specifically outlining what needs to be tested, test plans help to improve test coverage. The strategy specifies the requirements that must be validated, ensuring that all essential parts of the software are examined. It plays a role in the identification of the various sorts of tests to be done, such as functional, performance, security, and usability testing, ensuring a thorough examination of the software's behaviour.

Test plans helps in the identification of risks involved with software testing. The test plan enables early risk identification by recording potential risks and uncertainties such as dependencies, limits, and resource limitations. It enables teams to properly allocate resources, detect possible bottlenecks, and establish risk-mitigation plans. When hazards are addressed early on, the overall testing process becomes more efficient and effective.

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

First, we read the software documentation. This gave us a basic understanding of the features and functionality of the software. I have seen some tutorials and demos. This helped me see how the software was used. I created a test project in the software. This gave us the opportunity to test the software's features firsthand. I have spoken with group members and retirees who use this software. It gave us an idea of how they see the software and what their experiences are with the software. A list of some functions. A code review tool to help identify potential problems in your code. A free analyzer that can be used to identify potential vulnerabilities in your code. A powerful profiling tool that identifies bottlenecks in your code. Using these methods and tools, I gained a deep understanding of existing software. This review identified areas where the software could be improved and helped develop a software improvement plan. After using GitHub and Jira, I concluded that GitHub was easier to use as a developer. The user interface is simple, and the code editor is efficient and easy to use. GitHub also has a large community of users who can help with troubleshooting and code reviews.

Jira is a simple and powerful project management application for project managers. Updated user interface and added project management features. Jira also integrates with several other apps, making it easy to track progress and interact with other apps. GitHub is a developer code hosting service designed to be simple and easy to use. It is not a project management tool, but project managers can use it to record and track code. Your individual needs will determine the right tool for you. Jira is a great solution for project managers looking for a more efficient project management system. Looking for developers