# **MILESTONE 1** -- 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. Yatin Bawa | 4. Rehat Preet Kaur |
| 2. Muhammetyar Yarov | 5. Aayush Bhogal |
| 3. Nehmat Ladhar | 6. |

**Milestone 1 Tasks**

In this phase of the project you will:

* Setup teams of about 3-5 developers (6 is too large)
* Write and sign a team contract
* Create a GIT account
* Create a Jira account
* Add your professor to the GIT and Jira accounts
* Update Jira with the work performed and planned

**Deliverables due 4 days after your lab day:**

* Completed team contract.
* Fully initialized Git repository. **Be sure to send your professor the link to your GitHub repository and a screenshot of the GitHub users.**
* Fully setup Jira project. **Be sure to send your professor the link to your Jira Project.**
* Completed scrum report including reflection questions answered.

**Rubric**

|  |  |  |
| --- | --- | --- |
| **Individual** | Group participation | 80% |
| Teamwork | 20% |
| **Group** | Contract | 15% |
| Git repository | 25% |
| Jira project | 25% |
| SCRUM report & reflections | 25% |
| Meets deadlines | 10% |
| **NOTE** | Both the individual and group marks are calculated separately. Each member of the group will have their mark calculated based on their contribution to the group work and their contributions to the team. The group participation is a percentage that your professor feels you contributed to the group work. This is multiplied by the weight of the group participation component to determine your grade. |  |

**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** |
| Nehmat Ladhar | -Set up Jira and Github Account  -Completed Scrum Report  -Completed Reflection Question  -Participated in Group meeting  -Helped in bringing the tasks to completion | **-** |
| Yatin Bawa | -Set up Jira and Github Account  **-**Set up Git Repository  -Created Git Directory Structure  -Set up Jira Project  -Participated in Group meetings  -Helped in bringing the tasks to completion | **-** |
| Muhammetyar Yarov | -Set up Jira and Github Account  -Participated in Group meeting  -Helped in bringing the tasks to completion | **-** |
| Rehat Preet Kaur | **-**Set up Jira and Github Account  -Participated in Group Meeting  -Helped in bringing tasks to completion | **-** |
| Aayush Bhogal | **-**Set up Jira and Github Account  -Participated in Group meeting  -Helped in bring tasks to completion | **-** |
|  |  |  |
|  |  |  |

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 discusses in the meeting and the outcomes of the discussions.

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| 1. Requirements of Milestone1 | **-**Setting up of Github accounts  -Setting up of Jira accounts  -Setting up of Git repository  -Setting up Jira Project  -Complete Scrum Report  -Complete Reflection Questions | Success |
| 1. Division of Tasks | Each member was assigned a task to complete and other members assisted him in the successful completion of the task | Success |
| 1. Set up Git Accounts | All the members setup their Github accounts during the meeting following the instructions in the document provided | Success |
| 1. Set up Jira Accounts | All the members setup Jira Accounts following the instructions provided in the document | Success |
| 1. Providing Details | All the members sent the details of their respective accounts so that the group leader could successfully create a git repository and add them in it. | Success |
|  |  |  |
|  |  |  |

**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 |
| 1. Assignment of Tasks | Each member had a particular task assigned and the completion of it was their responsibilty. |
| 1. Account Setups | Everyone successfully setup their accounts on both Github and Jira |
| 1. Reflection | Everyone participated in the discussion of the questions provided in the Reflection |
| 1. Details | Everyone provided their details so that Git repository could be setup by the group leader. |
|  |  |
|  |  |
|  |  |

**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? |
| Yatin Bawa | Set up Git Repository  Set up Jira Project | 1 hour | Yes |
| Nehmat Ladhar | Set up Git Account  Set up Jira Account  Completed Scrum Report  Completed Refection questions | 1 hour | Yes |
| Muhammetyar Yarov | Set up Git Account  Set up Jira Account  Discussed Reflection Questions | 1 hour | Yes |
| Rehat Preet Kaur | Set up Git Account  Set up Jira Account  Discussed Reflection Questions | 1 hour | Yes |
| Aayush Bhogal | Set up Git Account  Set up Jira Account  Discussed 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 |
| 1. Working on the code | Each member will assigned their part in the testing process of the code provided |
| 1. Reflection | Eac member will provide their insight in the refection part of the assignment |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| 1. Successful Setup | The work on the project can be started as everyone is added and can successfully add their part in the reppositories |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Setting Up Accounts | Group Participation |
| Discussion on Refelction | Group Participation |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 (to be answered by the group)**:

Answer the following questions using your own words. Make sure that each answer comprises a minimum of 100 words.

1. GIT is an example of a version control system. List and explain 3 benefits of using a version control system.

Ans) The practice of monitoring and maintaining changes in the software code is **version control**, also known as source control. Software repositories with the capacity to maintain several versions of the program are known as version control systems. This implies that a file can save both the old and new versions whenever it is edited and saved in a version management system. The benefit of this is that the version control system makes it simple to roll back to a previous version in the event of an error.

The benefits are as follows:

1. A comprehensive **historical record** of each file allows you to track all modifications made to that file throughout its history, along with the individuals responsible for these alterations. Furthermore, most version control systems mandate the inclusion of comments for each software modification. This practice serves a dual purpose: it not only facilitates the monitoring of alterations but also offers insights into the reasons behind those changes.
2. Version control systems help software teams perform faster and more effectively, since the development environment is accelerating**. Branches** enable the creation of separate paths from the project's main history, allowing independent sets of changes. These branch modifications don't impact the main history. It's akin to making a copy of the software for parallel changes, facilitating the introduction of new features. Successful ones can be merged into the main branch, while unsuccessful ones can be discarded. This fosters experimental development without risking the integrity of the main product.
3. Leveraging a version control system offers improved traceability by integrating it with **bug tracking tools** such as Jira, which automatically associates each change with the relevant bugs being resolved. This becomes invaluable when you need to ascertain whether all project issues have been tackled and, more importantly, how they were resolved. It provides a clear and efficient means of tracking the resolution process and maintaining a comprehensive record of bug fixes.
4. Jira is a modern, web-based tool for managing software projects. Describe 3 advantages of using a project management tool like Jira.

Ans) A well-liked platform for leading groups of people working on software projects is **Jira**. Jira is divided into projects, and every project has a large number of issues. An issue is something that requires attention or action in some way.

Advantages of using a project management tool like Jira include:

**1. Efficient Task Management** : Jira excels in helping teams manage tasks, issues, and work items effectively. It provides a structured environment where different types of work, including feature development, design, and testing, can be organized into distinct issues. This clarity in task management ensures that all project-related work is systematically recorded and monitored. It reduces the risk of tasks being forgotten or overlooked, contributing to more comprehensive project execution.

**2. Clear Communication and Collaboration** : Jira facilitates clear and efficient communication among team members. By creating issues and assigning them to specific team members, everyone involved in the project knows their responsibilities and what tasks they need to address. As issues progress through various stages on the Kanban board, automatic notifications keep team members informed about changes in the status of issues they are working on or following. This streamlined communication process enhances collaboration, reduces confusion, and fosters teamwork.

**3. Customization and Flexibility** : Jira's adaptability is a significant advantage. Each project is unique, with different requirements, workflows, and objectives. Jira allows users to customize their project's setup to match these specific needs. For instance, you can create custom issue types tailored to your project's distinct elements, define fields to capture specific information, and even introduce new field types if necessary. This customization empowers teams to structure Jira to reflect their unique project processes. As project requirements evolve, Jira can be easily modified to adapt to changing needs, making it suitable for various project types, from small, straightforward endeavors to complex, long-term initiatives.

1. Write a brief history of the Kanban board. Describe why it is useful in a project like this one.  
   Ans) The status of our project is shown as three or more columns on a **Kanban board**. In its most basic from, a Kanban board consists of three columns: requested issues are listed in the first, ongoing issues are listed in the middle, and finished issues are shown on the right.

In Japan, the first Kanban board was created for Toyota automobiles in the 1940s. At every level of the production process, work and inventory needed to be controlled and managed. Kanban is a system’s intrinsically visual way to show how work is progressing. The steps in the overall process are represented by the columns, and you can visually see where each item is in the process by looking at where it is placed inside the columns. You can customise the Kanban board to fit the software process you’re using by adding columns.

A Kanban board is like a to-do list with columns. The first column is for tasks waiting to start. When we begin working on a task, we move it to the middle column. Once a task is all done, it goes to the last column. If we need more steps for our project, like quality checks, we can add extra columns. For example, for quality checks, we can have a special column.

In our project, we can use a Kanban board to help us control how much work we do at once. This means we don't want to start too many tasks at the same time, or it might get confusing and slow us down. With the Kanban method, we can set a limit on how many tasks we're allowed to work on simultaneously. This keeps our team from getting overwhelmed by having too much stuff going on all at once, and it helps us focus better on each task.