# **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.Cristhian Rosero | 4. |
| 2. Ayush Parajuli | 5. |
| 3. Arian Kazemi | 6. |

Complete guys

**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 at End of Lab**

* Completed SCRUM report & reflections

**Deliverables Due 24 hours after lab**

* Completed team contract
* Fully initialized Git repository
* Fully setup Jira project

**Rubric**

|  |  |  |
| --- | --- | --- |
| **Individual** | Group Participation | 75% |
| Teamwork | 25% |
| **Group** | Contract | 15% |
| Git Repository | 25% |
| Jira Project | 25% |
| SCRUM Report & Reflections | 35% |
| **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** |
| **Cristhian Rosero** | * Completed team contract * Fully initialized Git repository * Fully setup Jira project | **-** |
| **Ayush Parajuli** | * Completed team contract * Fully initialized Git repository * Fully setup Jira project | **-** |
| **Arian Kazemi** | * Completed team contract * Fully initialized Git repository * Fully setup Jira project | **-** |
|  | * Completed team contract * Fully initialized Git repository * Fully setup Jira project | **-** |
|  |  |  |
|  |  |  |
|  |  |  |

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 |
| Fully initialized Git repository | **Step by step of how to create, duplicated/clone a repository and set up your private and public SSH identification** | **All the members have the repository** |
| Fully setup Jira project | **Explanation of how to create your Jira account** | **All the members are in the project group** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 the project on Github and clone it | Project starting |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Cristhian Rosero | **Created and add the member to Github and Jira** | **1 hour** | **YES** |
| Ayush  Parajuli | **Reviewed the project and contributed while setting up jira project** | **45min** | **YES** |
| Arian Kazemi | **Setting up the jira project and contribute with the team for the rest of the assignment** | **1 :15** | **YES** |
|  | **Completed guys** |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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 |
| Cristhian Rosero | Review Milestone #2, Create meetings for discussion, Keep in touch with the group members |
|  |  |
| Completed guys |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Repository created |  |
| Jira group done | **Project initiated** |
| Group Members on Git |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Group creation | **All the member group help on the project** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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)**:

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

1. Teamwork and Collaboration: Version control systems make it possible for team members working on the same project to collaborate effectively. The project can be worked on by multiple developers at the same time, and Git enables them to smoothly merge their modifications. By tracking each change made to the codebase and offering tools to address any conflicts, it helps prevent disputes. This encourages collaboration, boosts output, and guarantees a smooth development process.

2. Versioning and History Tracking: A version control system makes it simple to keep track of your project's history. Every alteration made to the repository is documented, along with the person who made it, the date it was made, and the changes that were made. This capability of tracking changes and keeping a complete history is useful for auditing, troubleshooting, and rolling back to earlier versions when necessary. It acts as a safety net, letting you experiment and test out new ideas without worrying that you'll lose important work.

3. Branching and experimentation are supported by Git and many other versions control systems. Using branches, developers can set up independent development lines so they can work on new features or try out various strategies without affecting the main source. Changes made in one branch do not instantly affect the others because of the isolation provided by the branches. This feature simplifies the management of simultaneous development initiatives and supports agile development approaches. A branch can be merged back into the main codebase after it is stable and prepared, merging the changes into the finished result.

**2. Jira is a modern, web-based tool for managing software projects. Describe 3 advantages of using a project management tool like Jira.**  
  
1. Centralized Project Tracking: Jira provides a centralized platform to manage and track all aspects of a software project. It allows you to create and organize tasks, issues, and user stories, providing visibility into the progress and status of each item.

2. Agile and Scrum Support: Jira is well-known for its support of agile methodologies, particularly Scrum. It provides robust features to plan, prioritize, and manage agile projects effectively. Jira allows you to create and manage product backlogs, sprints, and user stories. It provides customizable Scrum and Kanban boards that offer visual representations of work in progress, making it easier to track tasks, identify bottlenecks, and ensure smooth workflow.

3. Customization and Integration: Jira offers extensive customization options to tailor the tool to the specific needs and workflows of your software project. You can create custom fields, workflows, and issue types, allowing you to adapt Jira to match your team's preferred terminology and processes. Additionally, Jira integrates with various development and collaboration tools, such as version control systems (e.g., Git), continuous integration servers (e.g., Jenkins), and communication platforms (e.g., Slack).

**3. Write a brief history of the Kanban board. Describe why it is useful in a project like this one.**The Kanban board is a visual project management tool that originated from the Toyota Production System (TPS) in the 1940s. It was developed by Taiichi Ohno, an industrial engineer at Toyota, to improve manufacturing efficiency and reduce waste.

The term "Kanban" comes from the Japanese words "kan" (meaning visual) and "ban" (meaning card or board). Initially, the Kanban system used physical cards (kanban cards) to signal the need for production or movement of inventory items within a manufacturing process. These cards represented individual tasks or items and were moved across a board, representing the flow of work from one stage to another.

In this project, a Kanban board can help the team track the progress of tasks related to the development, testing, and deployment of the software. It allows the project manager and team members to have a clear overview of the project's current state, identify potential blockers, and make informed decisions to ensure timely delivery. The visual representation of tasks and their movement across the board facilitates effective communication and coordination, enhancing overall project management and productivity.