# **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**: **2**

**Members Present**:

|  |  |
| --- | --- |
| 1. Jubril Olawale Akolade 167529213 | 4.Iraklis Tsanachtsidis 122226228 |
| 2. Frank Prerez 141647222 | 5.Aum Rasikbhai Parsana 112872221 |
| 3. Tarun Thomas 113605224 | 6.Rutarj Mrushad Shah 170870216 |
| 7. Faaz Sherwani 113026223 |  |

**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** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

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 |
| Milestone 1 – GitHub repo | We came together to setup the and version control (github) repository. | github repository all set up. Directory structure cloned to local machine, changes made and committed and pushed back to remote directory |
| Milestone 1 – Jira Project | Discussed and watched a demo tutorial on the sample working of a Jira project. Discussed how to go about it for our own project. | Jira project set up .All members and prof added , initial issues created and assigned. |
| Milestone 1 – Team Contract | Discussed and made the team contract. | Group Contract finished |
| Milestone 1 – scrum report | **Discussed how to fill up the scrum report.** | **done** |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Faaz [Group leader + Technical Lead] | Github repo finalize, make directory structure, add README files |
| Frank | Jira project finalized, create issues |
| Aum | Complete the team contract |
| Jubril | Scrum report [reflections] with inputs from all group members |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Project understanding | **We looked over the project requirements and made a plan** |
| Jira | **Learnt how to use the Jira platform** |
| Github | **Learnt basic pull, commit and push techniques.** |
|  |  |

**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 |
| Networking | **Everybody was eager to learn and get involved with the project.** |
| New Concepts | **Got to learn new concepts and software** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| Time management | **Managing the time and maintaining the efficiency was a problem and we had to make an agenda to stick to.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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.  
   i. Collaboration and Teamwork: Using version control systems, team members working on the same project can collaborate efficiently. Developers can work simultaneously on different parts of the codebase without clashing. They can create branches for specific features or bug fixes and then merge their changes back into the main codebase after they have made the changes. As well as resolving conflicts, version control systems also provide tools for resolving conflicts when multiple people make changes to the same file at the same time.

ii. Tracking and Reverting Changes: In version control systems, you can keep track of changes you have made to your project. Git records a snapshot of the entire project every time you commit changes. This history provides a detailed log of who made the changes, when they were made, and what changes were made. Using this functionality, you can track the project's evolution, review past modifications, and revert to previous states if necessary. It is particularly useful when dealing with bugs, regressions, or unintended changes.

iii. Branching and Experimentation: It enables developers to work on new features or experiments without affecting the main codebase. Version control systems provide branching mechanisms that enable developers to create independent lines of development. As a result, developers can work in isolation, experiment with new ideas, and iterate on new features without disrupting the stability of the main project. The branch can be merged back into the main codebase once the feature or experiment has been successfully completed.

1. Jira is a modern, web-based tool for managing software projects. Describe 3 advantages of using a project management tool like Jira.

i. Streamlined Project Planning and Tracking: It allows you to assign tasks to team members, set due dates, create dependencies between tasks, and create a centralized platform for planning and tracking all aspects of a project. Jira's user-friendly interface makes it easy to create and manage project backlogs, sprints, and Kanban boards, giving you a visual representation of the project's progress.

ii. Collaboration and Communication: To foster effective communication, Jira provides features that help team members communicate by allowing them to add comments, attach files, and mention other team members within tasks.

iii.Customization and Integration Capabilities: Jira offers a high level of customization and integration options, making it adaptable to a wide range of project management workflows and enabling seamless integration with other tools. Custom fields, issue types, and workflows can be created in Jira based on the needs of your project. A marketplace of plugins and extensions can also be integrated with Jira to enhance its functionality, such as agile planning tools, time tracking apps, and integration with other development tools like Git and Confluence.

1. Write a brief history of the Kanban board. Describe why it is useful in a project like this one.  
     
   Kanban boards are visual project management tools that were developed in the 1940s as part of Toyota Production System (TPS). Kanban was initially developed by Taiichi Ohno, an industrial engineer at Toyota, in order to optimize manufacturing processes and increase efficiency. In Japanese, Kanban means visual signal or card.

Kanban was initially used in Toyota factories to manage inventory levels and control materials flow. Physical cards or boards were used to represent work items and track their progress through various stages of production. During the course of the work, the cards represented specific tasks, and as the work progressed, they would move from one column to another, visually indicating the flow and status of the work.

Over time, Kanban was adopted and adapted by various industries and project management practices, including software development. In addition to visualizing work, limiting work in progress (WIP), and continuously improving processes, Kanban's principles proved valuable outside of the manufacturing industry.

In a project such as this one, the Kanban board is useful for promoting transparency, collaboration, and adaptability since it promotes transparency, collaboration, and adaptability. Through the Kanban board's visual nature, all members of the team are able to have a shared understanding of the project's progress and tasks at any time. By providing a clear view of work items, their status, and potential bottlenecks, it facilitates transparency. As a result of this visibility, team members can easily identify tasks that require assistance or discuss priorities based on the current status of the board.

Additionally, the Kanban board facilitates adaptability because it allows the team to change their workflow according to the changing requirements and processes of the project. It allows the team to easily add, remove, or rearrange columns to accommodate these changes. Team members are able to maintain organization and focus on the most important work if new tasks arise or priorities change, as they can easily be incorporated into the board. As a result of this adaptability, the project is capable of responding quickly to changing priorities, customer feedback, or emerging opportunities.