# **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. Divya Devendrasinh Rana | 4. Prabhjot Singh Longia |
| 2. Harsh Pahurkar | 5. |
| 3. Mukul Sharma | 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 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** |
| **Divya** | **Followed all the instructions** |  |
| **Harsh** | **Worked on Scrum report** |  |
| **Mukul** | **Majorly handled reflection** |  |
| **Prabhjot** | **Set up Github and Jira** |  |
|  |  |  |
|  |  |  |
|  |  |  |

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 |
| Distribution of Duties | **Project Manager – Harsh,**  **Tech Lead – Prabhjot,**  **Members and Testers – Mukul and Divya. All the roles are flexible.** | **Everyone agreed on their Duties and committed to them to the best of their Ability** |
| Rules and Regulations for each member | **Everyone agreed on the consequences and course of action to take for each scenario.** | **Everyone agreed on and talked about the rules for the group project as well as the consequences of not performing their duties well** |
| Where and when to meet | **Discussed everyone’s schedules and fixed a day when we can all do the project, which is Friday.** | **Friday either in person on college or Online through Discord** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Repository Structure | To follow the default structure of repository in Github. |
| Using Jira for Testing Discussions | Resolving issues based on order of posting or interrelatedness. |
| Flexibility on tasks | Everyone can contribute to the roles of PM or TL based on Project Requirements. |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Harsh | **Helped create SCRUM report and group contract while everyone discussed the same.** | **30 mins** | **YES** |
| Mukul | **Helped work on Reflections and wrote content for it while discussing within the group.** | **45 mins** | **YES** |
| Prabhjot | **Creating and Managing Git Repository and Jira Project** | **60 mins** | **YES** |
| Divya | **Contributed to Reflection and SCRUM report.** | **15 mins** | **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 |
| Prabhjot | Managing Repositories and committing code. |
| Harsh | Keeping Track of group progress as well as meeting deadlines. |
| Mukul | Understanding the problem well and brainstorming solutions. |
| Divya | Meeting deadlines and contributing to the overall project. |
| All Members | Creating SCRUM Reports and discussing solutions for Reflection. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Discussed everyone’s contributions for the project | **Helped understand everyone’s strengths and weaknesses for the project and what they can provide.** |
| Understood everyone’s schedules and communication styles | **This helped us understand when and how everyone can contribute to the project, allowing everyone to manage their time well.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| SCRUM Report | **Good Communication amongst members** |
| Reflection | **Good Communication amongst members** |
| Duties | **Understanding each other’s strengths and weaknesses** |
|  |  |
|  |  |
|  |  |
|  |  |

**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.

Version control systems like GIT can be powerful in the following ways:

* Teamwork + Collaboration: Git allows for multiple instances of the original repository using Git Clone, every member can have their own clone of the repository and eventually merge the changes back to the main branch. Git provides the feature to see what changed were committed by who and when.
* Version Backup/History: Every time a commit is made to the repository, a new version is created. These series of versions can be viewed as History, compare older versions to newer ones and most importantly go back to any older version at any point of time. It is almost like a Backup Time Machine allowing to go back to any saved (committed) versions.
* Branching: Team members have the ability to create branches of the main repository (branch). This allows them to work on specific new features or bug fixes without interrupting other working members.

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

Organization and Planning: Jira helps break down projects into tasks, assign them to team members, and prioritize and schedule them for efficient project planning. This can be helpful by dividing big chunks of workload into piecemeals and dedicated issues.

Collaboration and Communication: Jira facilitates seamless collaboration by allowing team members to discuss issues, ask questions, and provide feedback within the tool itself. It also integrates with communication tools like Microsoft Teams which can bring the collaboration experience to apps we are already familiar with.

Tracking and Reporting: Jira provides active tracking and reporting features, allowing us to monitor task status, project progress, and generate reports for that can help decide future decisions and trajectories of the projects.

1. Write a brief history of the Kanban board. Describe why it is useful in a project like this one.

The word Kanban has been derived from a Japanese word kan meaning “visual” and ban meaning board. It is a project management tool that helps visualize workflow in the form of a board. Thus, the name Kanban was originated. Overtime Kanban has found use in project management and software development.

A Kanban can be used to **visualize and depict delivery routes** whilst also mentioning the different stages of the delivery such as “Incoming Shipment”, “Out for Delivery” etc.  
Kanban encouraged **regular updates/meetings** which motivated the team to look for places where there is room for improvement while maintaining **efficient and quick communication.**

1. Overall, Kanban provides the ultimate solution by visualizing the problem at hand (in this case delivery routes) then encourage regular updates while labeling different stages of the process. All this along with keen communication can lead to one of the best management techniques, leading to optimum productivity.