# **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**: \_4\_

**Members Present**:

|  |  |
| --- | --- |
| 1. Anna Francesca Dela Cruz | 4. Irish Banga |
| 2. Gulpreet Kaur | 5. |
| 3. In Tae Chung | 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(s)** | **Tasks Completed** | **Tasks Delayed/Blocked** |
| 1,2,3,4 | Git Hub Repository set-up | **N/A** |
| 1,2,3,4 | Jira Project set-up |  |
| 1,2,3,4 | Group Contract signed |  |
| 1,2,3,4 | SCRUM Report contributions |  |
|  |  |  |
|  |  |  |
|  |  |  |

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 |
| SRUM Report | The discussion revolved around the contributions towards the SCRUM report and reflection questions. | **Equal Responsibility** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| SRUM report contributions | To avoid conflict, all group members will share equal responsibility with regard to SRUM reports and submit a final version that all members agree to. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| N/A | N/A | N/A | N/A |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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 |
| N/A | N/A |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| N/A | N/A |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| N/A | N/A |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| N/A | N/A |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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.**

Listed below are some of the benefits of using the GIT version control system:

* 1. The first and foremost benefit, as the name suggests, is the ability to keep a record of changes we’ve made to our files right from the time they were checked into the version control system. This is also useful to help us track the progress made on a certain project and help us backtrace our steps in case something goes wrong. We can easily revert to a working/bug-free copy of our files using this system in case an issue occurs at any point in time.
  2. The second benefit of GIT is the cloning feature and branching support. We can easily create a copy(clone) of the entire source code in the main repository allowing us to work independently This coupled with branching support proves even more useful as multiple people can work on the code with the same goal. This will allow multiple solutions to a problem and eventually, the most efficient or bug-free code can be incorporated into the main branch.
  3. The GIT version control system can work either remotely or locally without needing any external support. In short, we can keep our source code private and simultaneously keep a record of different versions on our machine itself. This is another boost for privacy while providing full functionality. It must also be mentioned that the details generated by this system are entirely unique to each machine; hence data manipulation is impossible.
  4. Another very useful benefit is it helps us create a “stash” of our work on top of a version. This is like creating a checkpoint in the git repository, saving all the modified files and their details but without committing the changes to our repository. This allows us to work on separate tasks at a given time without losing any progress or productivity.

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

Three advantages of using a project management tool like Jira:

* 1. Effectively allocating responsibilities: Jira lets us allocate tasks and responsibilities to team members. Thanks to this functionality, everyone is aware of their roles and duties within the project. We may evenly spread the effort and establish accountability for each team member by assigning tasks in Jira. It aids in eliminating overlapping tasks and confusion and encourages a more structured and effective work environment.
  2. Easy access to Task statuses and information: Team members may readily access project-related information and monitor the status of tasks with Jira's user-friendly interface. It provides real-time information on project progress, task fulfilment, and any potential barriers through dashboards and customized reports. Jira allows team members to check task statuses quickly, spot bottlenecks, and make choices based on the most recent information.
  3. Extensions and Integration of Code: Jira offers a range of extensions and integrations, specifically related to code repositories and development tools. By connecting Jira with version control systems like Git or Bitbucket, it enables smooth integration between code modifications and associated tasks. This integration enhances collaboration among developers, testers, and project managers, allowing them to easily access code changes, monitor issues, and establish connections between code commits and specific tasks in Jira.

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

As per information gathered from online sources and from the class notes, the “kanban” method was developed in the 1940s by Taiichi Ohno, an engineer at Toyota. After WWII, Toyota could not afford to follow the usual method of production and needed a more efficient way of managing inventory. The usual method was to order a large number of car parts and use them when needed, but this led to huge upfront costs and a waste of supplies. Ohno developed a method to order car parts only when necessary. A specific part was written on a card - a “kanban” in Japanese - and each time the part was used, the kanban was sent out to suppliers. Eventually, the “kanban” method was applied to project management, and the Kanban board was born.

The Kanban board is a tool that helps teams manage projects and is incorporated into Jira. Two major components of Kanban boards are cards and columns. Team members write one task or work item per card, which helps them visualize everyone’s responsibilities. Columns represent the different stages of the team’s workflow. Cards move through them. For example, the default columns in Jira are “To Do”, “In Progress” and “Done.” Utilizing columns allows teams to see the status of their teammates' responsibilities.

Kanban boards are useful for projects like this one for a few reasons:

1. Through cards, Kanban boards help break down large projects into smaller, more manageable tasks very similar to incorporating a new feature in our program with the help of several smaller functions.
2. This division of tasks makes assigning them to team members easier and encourages a fair distribution of work. As we move further along the project, the team will be split into two groups (i.e., developers and testers) and using the Kanban board shall ensure strict boundaries around the tasks assigned to each group.
3. Because Kanban boards keep a log of the tasks assigned to each team member, they ensure accountability. Also, we can set up dependencies (such as adding child issues) among different tasks to ensure proper workflow.
4. Kanban boards lessen the number of meetings and email updates; At just a glance, Kanban boards allow you to gauge the status of the project and see everyone’s progress at all times.
5. Through the comment feature on cards, Kanban Boards facilitate communication and feedback from team members about specific tasks.