# **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:** [**FALL24-SFT221-NAA-4**](https://github.com/Prashant-Khadka-Magar/FALL24-SFT221-NAA-4)

**Members Present:**

|  |  |
| --- | --- |
| **1. Prashant Khadka Magar** | **4. Nirajan Bhandari** |
| **2. Lucky Maharjan** | **5. Pradip Dhami** |
| **3. Rohit Siva** | **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** |
| **Rohit Siva** | **Reflection** |  |
| **Prashant Khadka Magar** | **Setup GitHub repo and Jira project** |  |
| **Lucky Maharjan** | **Scrum Report** |  |
| **Nirajan Bhandari** | **Added tasks and details to Jira** |  |
| **Pradip Bahadur Dhami** | **Finalized team contract** |  |
|  |  |  |
|  |  |  |

**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 |
| Team Contract | **Discussed roles, responsibilities, and project goals** | **Drafted and signed team contract** |
| GitHub Setup | **Set up repository and permissions** | **Repository initialized and accessible** |
| Jira Setup | **Established Kanban board, assigned initial tasks** | **Project board ready for use** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Weekly Scrum meeting schedule | **Ensures regular progress updates and task coordination** |
| Use GitHub for code versioning | **Enables collaborative work and history tracking** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Rohit Siva | **Reflection** | **30 min** | **Yes** |
| Prashant Khadka Magar | **Setup GitHub repo and Jira project** | **45 min** | **Yes** |
| Lucky Maharjan | **Scrum Report** | **1.5 hrs** | **Yes** |
| Nirajan Bhandari | **Added tasks and details to Jira** | **10 min** | **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 |
| Prashant Khadka Magar | **Analyze the problem requirements and review any existing code** |
| Lucky Maharjan | **Begin creating header files for new data structures and document them in GitHub** |
| Rohit Siva | **Draft the test plan template and outline initial test cases based on project requirements.** |
| Pradip Dhami | **Create and assign detailed Jira issues for Milestone 2 tasks** |
| Nirajan Bhandari | **Ensure the repository is well-organized, with all files from Milestone 1** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Team contract signed | **Clarified roles and responsibilities** |
| GitHub and Jira setup completed | **Enabled project tracking and version control** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| GitHub Setup | **Smooth setup with all members able to access the repository** |
| Team Contract | **Clear discussion of roles and responsibilities** |
| Jira Task Assignment | **Efficient organization of tasks and assignments** |
|  |  |
|  |  |
|  |  |
|  |  |

**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.  
     
   Benefits of using a version control system, is being able to effectively manage code and collaborate with other software developers, track the history of the software changes, and ensure proper backup exists for the code, in the case of an error occurring. First, when collaborating using a version control system, it allows many developers to work on code without overwriting someone else's work, which allows developers to create different features and test their code, with the working version before they push the modified code. Next, the history and versions of a program can be easily reviewed and compared with working versions of code, when creating new features, which helps debug and also allows you to revert back to the debugged code if a error in the code occurs, and you can view when and what was modified in code. Finally, version control provides backup, such as when an error occurs in the code, you can easily revert the code via the backup version of the code, and ensure that the code is always working.**
2. **Jira is a modern, web-based tool for managing software projects. Describe 3 advantages of using a project management tool like Jira.  
     
   Project management tools like Jira provide a group to effectively divide tasks for easy task management, it enhances the collaboration of the group work for the program, and finally allows an agile workflow assisting in visualizing work that needs to be done. Jira effectively provides enhanced task management by using its tasks and feature tracking, which teams working on projects are provided with deadlines and have their progress tracked, and it ensures every group member has their sights set on a goal making project task management much more effective. Next, it enhances the collaboration for group work by providing features like comments, notifications, and team tagging, creating smooth communication for team members. A group also can see all the necessary info required for working on the task in one place rather than needing to check notes from meetings or emails, a groups work is all held in a single place, improving the teamwork and efficiency of a team. Finally, it allows an agile workflow by making it easy to implement scrum and Kanban workflows, because teams can make sprints and track backlogs, in order to visualize work that is currently in progress by the group.**
3. **Write a brief history of the Kanban board. Describe why it is useful in a project like this one.  
     
     
   The Kanban board was first created in the 1940's by Toyota's manufacturing system to improve the production efficiency of work done, inspired by supermarket stocking procedures. The Toyota engineers named it Kanban meaning billboard in Japanese, so they could visually see the progress of production tasks, and make sure that workers were not overloaded with work, and work was moving as intended. The Kanban board is useful in a software development project, because it provides a clear view of the progress of a project, as tasks are displayed on the board and move through columns of the board as the task is being worked on such as To Do, In progress, Complete. The visualization of the workflow of the tasks makes it easy for group members to identify the work needed for a task, how much they should prioritize a task, and how soon they should complete a task, making it ideal for collaborative and software development work.**