# **Milestone 1 Scrum Report**

All students are expected to attend the scrum meetings and to participate. Failure to do so will result in greatly reduced grades.

**GROUP**: **2**

**Members Present**:

|  |  |
| --- | --- |
| 1. Eric Yakimoff | 4. Philip Ayomide Tijani |
| 2. Karishma Singh Mahender | 5. |
| 3. Luca Novello | 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 | 25% |
| Git repository | 25% |
| Jira project | 25% |
| Scrum report & reflections | 25% |
| **Deadline** | 20% deduction for each day you are late |  |
| **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** |
| Eric Yakimoff | Group Contract, Scrum Report, Github setup, & Jira setup | n/a |
| Karishma Singh Mahender | Group Contract, Scrum Report, Github setup, & Jira setup | n/a |
| Luca Novello | Group Contract, Scrum Report, Github setup, & Jira setup | n/a |
| Philip Ayomide Tijani | Group Contract, Scrum Report, Github setup, & Jira setup | n/a |

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 |
| Github setup | Setup and processes | Setup complete |
| Jira setup | Setup and processes | Setup complete |
| Group Contract | Terms and conditions of contract | Contract complete |
| Scrum Report | Format and reflections | Report complete |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Online meeting location | We chose to communicate via Microsoft Teams |
| Prioritization of tasks | Setting up how to equally split the workload |
| Contract conditions and dates | Decided when and where to meet as well as additional conditions |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Eric Yakimoff | Group Contract, Scrum Report, Github setup, & Jira setup | 1-2 hrs | Yes |
| Karishma Singh Mahender | Group Contract, Scrum Report, Github setup, & Jira setup | 1-2 hrs | Yes |
| Luca Novello | Group Contract, Scrum Report, Github setup, & Jira setup | 1-2 hrs | Yes |
| Philip Ayomide Tijani | Group Contract, Scrum Report, Github setup, & Jira setup | 1-2 hrs | 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 |
| Eric Yakimoff | Meeting, analysis of the problem, Jira entries, scrum report, reflections |
| Karishma Singh Mahender | Meeting, analysis of the problem, Jira entries, scrum report, reflections |
| Luca Novello | Meeting, analysis of the problem, Jira entries, scrum report, reflections |
| Philip Ayomide Tijani | Meeting, analysis of the problem, Jira entries, scrum report, reflections |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Contract completed | Dates and conditions established |
| Github repository setup | Access to the project files are available to everyone |
| Jira project setup | Project planning can begin |
| Scrum report completed | Week 1 report completed |
| Online communication established | Members can communicate at anytime |

**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 |
| Contract completed | Group effort |
| Github repository setup | Group effort |
| Jira project setup | Group effort |
| Scrum report completed | Group effort |
| Online communication established | Group effort |

**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)**:

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.***

A version control system (VCS) like Git offers numerous benefits for software development and project management:

* **Collaboration**: A VCS allows multiple developers to work on the same project simultaneously. Each developer can work on their own branch without interfering with others' work. This makes it easier to manage changes and ensures that everyone is working with the most up-to-date code.
* **History and Tracking**: Every change made to the codebase is recorded, along with who made the change and why. This history record is useful for debugging, understanding the evolution of the project, and rolling back to previous versions if a new change introduces bugs.
* **Backup and Recovery**: A VCS ensures that your code is safely stored and can be recovered in case of accidental deletion or hardware failure. Since the code is stored in multiple locations, the risk of losing the entire codebase is minimized.

1. ***What is a version control system? Why does GitHub qualify as a version control system?***

A version control system (VCS) is a tool that helps manage changes made to source code over time. It tracks revisions made by multiple programmers, enabling them to collaborate on a project without overwriting each other's work. It also provides a way to revert to previous versions of the code if needed. GitHub qualifies as a version control system because it hosts Git repositories, allowing users to manage their code with Git's powerful version control features. GitHub adds additional functionalities such as pull requests, issue tracking, and project management tools, making it a comprehensive platform for collaboration in software development.

1. ***What is Jira? How are we going to use Jira for this project?***

Jira is a project management and issue-tracking tool developed by Atlassian. It helps teams plan, track, and manage software development projects through features like customizable workflows, issue tracking, sprint planning, and reporting. For our project, we will use Jira to organize and prioritize our tasks, track the progress of each team member, and ensure that we meet our deadlines. By creating user stories, tasks, and subtasks in Jira, we can break down the project into manageable pieces and assign them to team members. This will help us maintain clear communication and ensure that everyone is aligned on project goals and timelines.

1. ***Why is a Kanban board useful in software development. What are the advantages of using Kanban board?***

A Kanban board is a visual tool used in software development to manage workflow and track progress. It is divided into columns that represent different stages of the development process.

The advantages of using a Kanban board include:

* A Kanban board provides a clear visual representation of the work in progress, making it easy to see the status of tasks briefly. This helps team members and stakeholders quickly understand what is being worked on, what is completed, and what is pending.
* Unlike other project management methodologies, Kanban is highly flexible and can be adapted to fit the specific needs of the team. Tasks can be added, removed, or moved between columns as needed.
* By limiting the number of tasks in progress, a Kanban board helps ensure that the team is focused on completing work rather than starting new tasks. This promotes a continuous flow of work and helps prevent overloading team members.
* The transparency and visibility provided by a Kanban board encourage regular review and improvement of the workflow. Teams can easily identify inefficiencies and improve productivity and quality.