# **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 6**

**Members Present**:

1. Sahan Gallage
2. Ileperuma Gunarathna
3. Dhanuth Hennedige
4. Sean Li

**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** |
| All | Prepared and got together, entered on GitHub and Jira | 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** | N/A |
| **Reason for delay or block** |  |
| **Impact on Project** |  |
| **Solution or work-around** |  |
|  |  |
| **Delayed or Blocked Task** | N/A |
| **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.

|  |  |  |
| --- | --- | --- |
| **Topics** | **Discussion Summary** | **Outcome** |
| Introduction/onboarding | What this assignment program is about and what to expect, what is/using Jira, GitHub | Ready for milestone 1 and looking ahead at milestone 2 |

**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** |
| Choosing Sean as the team head | Has a relevant background in the context of this assignment |

**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?** |
| All | Understand the assignment and Jira, GitHub | 1-2 days | 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** |
| All | Study the assignment problem, study the code, prepare for milestone 2 |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| **Outcome** | **Impact on Project** |
| Completed milestone 1 | Ready for the next milestone and the ensuing completion of the assignment |

**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** |
| Milestone 1 | Members knew each other before, strongly adept at programming, know GitHub and Jira |

**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** |
| Milestone 1 | None |

**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. How did using GitHub simplify storing and organizing project files? Reflect on how having a centralized repository improved accessibility and efficiency for the team.

GitHub allows all kinds of files, whether program codes or word documents, to be stored in a collaboration-based synchronization system made for programmers, so all members can work independently while contributing together once each one has finished his tasks. You can easily push your commits and fetch changes with the GitHub desktop application. GitHub has the same exact file and folder arrangement as your desktop, so it works and behaves exactly like how you’re used too on your own computer. Having a centralized repository means that everyone can work on a cloud-type platform, easing communication and streamlining collaboration, with real-time commits seen instantly by all members.

1. How did GitHub help you manage file sharing and collaboration among team members?

GitHub allows all members to submit changes and fetch-synchronize changes from other members’ work in the blink of an eye, made easier by our use of the desktop application, or command-line instructions for those of us who prefer the traditional way. We can see pending changes in the form of pull requests, and easily approve the pull request so everyone will be able to absorb those changes into their code as well. Because all the activities are done in real time, we can see what others have accomplished right away, which is the key of online cloud collaboration.

1. How did the Kanban board in Jira assist in visualizing the progress of tasks and identifying bottlenecks? Reflect on how this feature helped prioritize tasks and manage the overall project timeline.

Jira Kanban boards allow us to see what assigned tasks, deadlines, and issues at a glance, it’s like having sticky-notes and agendas all together in one place in a virtual office. It’s like having a central project management dashboard where everyone can be updated on what’s expected of them, checking in on the status of tasks. These features allow us to see and communicate on what tasks are coming up and what’s still to be done, including time-sensitive due dates and critical paths that will take up extraordinarily amount time and effort, and keeps everyone in line in a synergized fashion, singularly focused on finishing the assignment jointly, so that nobody risks falling behind, slacks off, confused, or out of the loop.

1. How did using Jira to create and assign tasks improve your ability to manage the project's workflow? Reflect on how breaking down the project into smaller tasks helped you maintain clarity and focus.

Using Jira gives us the ability to effectively manage the project and other members. We can simply communicate to others what we expect of them and show everyone the planned schedule and overall workflow of the project right on the screen; everyone is on the same page about what to do and what’s done. By dividing up the tasks into manageable sizes that can be done in one or two settings, we break down the problem into smaller chunks that we can wrap our programmer’s mind around, without feeling unsure or bloated. This way, ambiguities and wasted work can be eliminated, we have a effective division of labour into assigned independent tasks, and we have a crystal clear idea of what we are supposed to be working on.