# **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. Chia-Yu Chien | 4. Fang Lin |
| 2. Hansol Nam | 5. |
| 3. Wai Bong Yung | 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** |
| Chia-Yu Chien | **Setup GitHub, Jira** | **None** |
| Hansol Nam | **None** | **None** |
| Wai Bong Yung | **None** | **None** |
| Fang Lin | **None** | **None** |
|  |  |  |
|  |  |  |
|  |  |  |

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 |
| Group Contract | **Have a meeting discuss about group contact** | **Finished group contract** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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.

Version control systems like GIT provide several significant benefits.

First, we can use them to track and record files changes over time. It is helpful for identifying issues and understand when it was introduced and knowing when a feature is added.

Second, it supports multiple people working on a same codebase. It helps for managing conflicts between changes made on files. Team members can work independently on branches and merge their updates to the project.

Finally, we can reduce the risk of data loss with version control systems helps. As every commit is made, it is very likely to lose files. Version control systems can record every change and preserve data. Even if a developer loses some files due to some tiny errors, we can find them back very easily.

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

A version control system is a tool that helps manage changes to source code or other documents. VCS helps us work on the same files simultaneously, track individual changes over time and enabling uses to revert to previous versions. GitHub basically is a platform for storing repositories. It is qualified as a VCS, because it incorporates GIT, which is a distributed version control system. On GitHub, we can track issues and manage changes to our repositories. Developers can create branches, commits, and pull requests on it or with tools provided by GitHub. We also can create issues to discuss features with other developers on it.

1. What is Jira? How are we going to use Jira for this project?  
   Jira is a project management tool provided by Atlassian. It is primarily used for issue tracking, project management. We can separate our development works into many issues or tasks. Then we use it to track the progress. It is highly integrated with the treading concept of agile software development. Now, it is one of the most popular solutions for project management.  
   We are going to use Jira to plan our features and create tasks to track the accomplishment progress. We use it to collaborate with each different role and facilitate communication with team members. People can provide advice, documents, and updates with it, and we can understand everyone’s progress of their work. Also, we will use it as a tool to track each milestone and its workload. We will understand every team member’s performance and ensure the project remain on track and improve our delays.
2. Why is a Kanban board useful in software development. What are the advantages of using Kanban board?  
   A kanban board is a tool to help us visualize software development works. We can use it to have a transparent board to coordinate and plan future features. It displays all tasks visually. It helps a team understand every member’s progress. Also, it helps us to limit the number of tasks in every specific stage. That can reduce overloads for team members and find out the development bottlenecks. It is very flexible tool. We can arrange our Kanban board to match a team to use. It allows teams to make quick changes and continuously deliver work without waiting for review. Also, software development emphasizes on continuous delivery. We can use it to focus on completing tasking and moving them to other stage. Developers can have a consistent context for development. Kanban encourages a healthy cycle of development and delivery, and it can lead to faster feedback too.