# **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: 3**

**Members Present**:

|  |  |
| --- | --- |
| 1. Chang Cui | 4. Xiaofei Xu |
| 2. Lok Yin Tai | 5. Xinyang Wu |
| 3. Siu Man Cheng | 6. Ye Tian |

**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** | **Tasks Completed** | **Tasks Delayed/Blocked** |
| Chang Cui | SCRUM Report & Reflections | N/A |
| Lok Yin Tai | Jira Project | N/A |
| Siu Man Cheng | Contract | N/A |
| Xiaofei Xu | SCRUM Report & Reflections | N/A |
| Xinyang Wu | SCRUM Report & Reflections | N/A |
| Ye Tian | Git Repository | 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** | N/A |
| **Impact on Project** | N/A |
| **Solution or work-around** | N/A |
|  |  |
| **Delayed or Blocked Task** | N/A |
| **Reason for delay or block** | N/A |
| **Impact on Project** | N/A |
| **Solution or work-around** | N/A |

**Summary of Meeting:**

A summary of the main points discusses in the meeting and the outcomes of the discussions.

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| Share contact with each other | Team members shared their contact information for better communication | Improved communication and collaboration among team members |
| Decide on communication method | The team agreed to create a group chat on Teams | Established a designated communication channel for efficient and timely information sharing |
| Task Assignments | Each member expressed their preferences and chose tasks to complete | Clearly defined individual responsibilities and task allocation for better task progress tracking |
| Create Group Contract rules | Discussed the importance of creating a group contract with defined rules | Created a group contract outlining team rules, expectations, and guidelines |

**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 |
| Share contact with each other | To improve communication and collaboration among team members |
| Decide on communication method | Establish a designated communication channel for efficient and timely information sharing |
| Task Assignments | Clearly define individual responsibilities and allocate tasks for better task progress tracking |
| Create Group Contract rules | Establish a common understanding of team rules, expectations, and guidelines for a productive work environment |

**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? | |
| Chang Cui | SCRUM Report & Reflections | 30 mins | | Yes |
| Lok Yin Tai | Jira Project | 40 mins | | Yes |
| Siu Man Cheng | Contract | 30 mins | | Yes |
| Xiaofei Xu | SCRUM Report & Reflections | 30 mins | | Yes |
| Xinyang Wu | SCRUM Report & Reflections | 30 mins | | Yes |
| Ye Tian | Git Repository | 40 mins | | 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 |
| Chang Cui | SCRUM Report & Reflections |
| Lok Yin Tai | Jira Project |
| Siu Man Cheng | Contract |
| Xiaofei Xu | SCRUM Report & Reflections |
| Xinyang Wu | SCRUM Report & Reflections |
| Ye Tian | Git Repository |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Improved communication and collaboration among team members | Enhanced coordination and efficiency in project tasks and decision-making |
| Established a designated communication channel on Teams | Streamlined communication process, ensuring timely information sharing |
| Clearly defined individual responsibilities and task allocation | Improved task management and progress tracking, reducing overlaps and confusion |
| Created a group contract outlining team rules and guidelines | Established a common understanding of expectations, promoting a productive work environment |

**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 |
| Active participation | All team members were engaged and contributed to the discussions. |
| Task completion | Several tasks, such as the SCRUM Report and Reflections, were successfully completed. |
| Task distribution | Each team member took responsibility for specific tasks, leading to progress in multiple areas. |
| Task prioritization | Our team prioritized tasks based on their importance and urgency, ensuring that critical items were addressed first. |
| Task coordination | Team members coordinated their efforts and collaborated smoothly to accomplish tasks in a timely manner. |
| Collaborative decision-making | Our team made decisions through consensus, considering different perspectives and reaching agreements that satisfied all members. |

**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 | Everything went very well in the meeting |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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.
2. Teamwork and collaboration: the version control system can allow multiple developers work on a project together. They can work on different parts of a codebase at the same time and merge changes they made easily, which is helpful to reduce conflicts, and everyone can know the latest changes of the code. It makes the teamwork more efficiently.
3. Version tracking: git allows developers to keep track and manage every change made to their codebase. Git can preserve complete history of changes and different versions, so developers can know who make changes and why they did that. If there is a mistake, developers can review, compare, and turn back to earlier version of the code.
4. Backup: Git also can work as a backup. Every team member has a complete version and modification history of the codebase. If the central server and backup drives break down. Developers can use team member’s local Git repository to recover all the code.
5. Jira is a modern, web-based tool for managing software projects. Describe 3 advantages of using a project management tool like Jira.  
     
   1) Project tracking: Developers can use Jira to create and manage tasks and issues. They enable to use Jira to customize their workflows and track status of each task of project. They also can use Jira to assign task to team members and monitor how they finish it.

2) Team collaboration and communication: it allows developers to use a shared platform to work on a same project, and developers can use this platform to comment and discuss on issues or problems. Team members also can use Jira to set up notification to let every team member know updates and participate in discussion, which is helpful to make team work more efficiently.

3) Reporting and Analytics: Jira has built-in reporting functionality to track progress, team performance. It provides different predefined reports and dashboards to let users know the information about project progress and problems. Users also can extract specific data to analysis and make decision by creating customized report and using advanced querying functionality.

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

Kanban began in the early 1940s. Taiichi Ohno (Industrial Engineer and Businessman) created the first Kanban system for Toyota Automotive in Japan. The term "Kanban" itself comes from Japanese, roughly translating to "visual signal" or "card." It was designed as a simple planning system with the goal of optimally controlling and managing work and inventory at each step of production.

While Taiichi Ohno invented kanban in the manufacturing industry, David J. Anderson was the first to apply the concept to IT, software development, and knowledge work in general in 2004. David based the Kanban Method on the writings of Taiichi Ohno, Eli Goldratt, Edward Demmings, Peter Drucker, and others, using principles such as pull systems, queuing theory, and flow.

A Kanban board is a tool for visualizing the flow of work and its status as it progresses toward completion. A simple board has three columns: To Do, In Progress, and Done. Tasks are represented by cards, which travel from left to right across the board to indicate their progress.

Kanban is useful in a project like this one as it can be:

(1) Visualize Work: Kanban boards make complex information more understandable by offering a visual depiction of the tasks at hand, allowing team members to monitor the status of every item of work at any moment.

(2) Limit Work in Progress: By keeping the number of tasks in the 'In Progress' stage to a minimum, we can ensure that jobs are performed more quickly and efficiently, as team members are not overburdened by taking on too much work at once.

(3) Flow Management: Kanban aids in understanding how work goes through the team's process by revealing bottlenecks and roadblocks, allowing for quicker issue resolution, and increased overall productivity.

(4) Flexible: Kanban can be adjustable according to real-life changes. If the priorities change, the tasks can be rearranged easily accordingly.

(5) Transparency into the workflow: it allows teams to identify opportunities to process continuous improvement.