**CPT330 Software Engineering Project Management**

# Sprint Retrospective Notes

**Sprint number: 1**

Meeting date: 07/10/2021

Attendees:

|  |  |  |
| --- | --- | --- |
| **Student name** | **Student number** | **Role (e.g., Scrum Master)** |
| Lars Werner | s3800032 | Scrum Master |
| James Haig | S3803084 | Developer |
| Miriam Saftlas | s3739725 | Developer |
| John Cunningham | s3775563 | Developer |

**Lessons learned**

1. What went well?
   * Team dynamics
     + Constant communications
   * Development
     + The team was motivated and devoted much time to coding and tackling the set features and tasks
2. What did not go well?
   * Organising schedules within the time constraints, working across different time zones presented its challenges.
   * RMIT desktop not working for testing and clarification needed constantly to know what we needed to do.
   * Had to organise message scrums due to everyone not being available for a group chat daily.

**Start / stop / continue**

1. What should we start doing in future sprints?
   * Dedicated area for us to keep daily scrums
   * Communicating where we are up to with our coding
   * Discuss which tasks rely on each other and must be completed first
2. What should we stop doing in future sprints?
   * Leaving branches open once completed
   * Delegating tasks to individuals without detailing that they are reliant on others completing their task first for it to be commenced effectively
3. What should we continue doing in future sprints?
   * Trying to make sure daily targets or tasks and effort is achieved
   * Offering help to others where needed

**Sprint number: 2**

Meeting date: 14/10/21

Attendees:

|  |  |  |
| --- | --- | --- |
| **Student name** | **Student number** | **Role (e.g., Scrum Master)** |
| James Haig | S3803084 | Scrum Master |
| Lars Werner | s3800032 | Developer |
| Miriam Saftlas | s3739725 | Developer |
| John Cunningham | s3775563 | Developer |

**Lessons learned**

1. What went well?
   * Communication was better than before; messages were less flooded and only focused on critical issues.
   * Features to be worked on were completed ahead of schedule.
   * Workload was spread out well amongst all members of the team.
2. What did not go well?
   * Certain task conflicts meant that some peoples tasks were staggered in starting rather than starting at the beginning of the week as they had to wait for completion of others task before starting their own.
   * Underestimated the amount of work that could have been completed in the timeframe.

**Start / stop / continue**

1. What should we start doing in future sprints?
   * Work out common factors, such as shared variables ahead of time, rather than fix code later.
   * Have any underlying questions about specifications answered ahead of sprint commencing so we don’t need to change anything midway.
2. What should we stop doing in future sprints?
   * Sandbagging the amount of work that we can complete.
   * Daily updates feel unnecessary, especially in a small team where tasks are completed efficiently.
3. What should we continue doing in future sprints?
   * Our communication methods through discord, our task breakdown framework. Time management and general coding practice, outside of the common variable issue mentioned previously.

**Sprint number: 3**

Meeting date: 21/10/2021

Attendees:

|  |  |  |
| --- | --- | --- |
| **Student name** | **Student number** | **Role (e.g., Scrum Master)** |
| Miriam Saftlas | s3739725 | Scrum Master |
| James Haig | s3803084 | Developer |
| Lars Werner | s3800032 | Developer |
| John Cunningham | s3775563 | Developer |

**Lessons learned**

1. What went well?
   * Receiving and implementing feedback
   * Team members all chipped in and helped each other out when coding tasks required help or clarification
   * Members communicated well regarding the times that they were available to work on the coding and other matters related to the assignment
   * Git branching and merging took place in an easy fashion without any obstacles
2. What did not go well?

* Troubleshooting was not as clear and concise as it could have been.
* At times, we had to wait on discussion board feedback before continuing work, holding the coding back.
* There was a certain task that was first labelled as one chunk. Once the coding of the task started, it was noted that it had to be divided into further smaller tasks so that it would not be too large of a chunk to be worked on at once.

**Start / stop / continue**

1. What should we start doing in future sprints?

* Breaking up methods into smaller methods for readability
* Create more than one class
* Add tickets to backlog and differentiate between features and bug fixes
* Incorporate a team board to keep track of where we were at
* Thinking about how a task will be broken up from the outset

1. What should we stop doing in future sprints?

* We were not granular enough with our tasks
* Having the owner responsible for subtasks as they did not always match up and could better be tackled by other team members

1. What should we continue doing in future sprints?

* Communicating with team with the same frequency.
* Clarifying tasks with client.
* Asking for help from fellow developers when we hit a roadblock or snag
* Letting our team-mates know where we are up to with our tasks