# Product Development Life Cycle

The project is split into four teams – the Web Development (WEBDEV) team, the Data Science (DATASCI) team, the Design Team (DESIGN) and the Project Team (PROJECT). The WEBDEV team focusses on elements of the project related to the web application. The DATASCI team focuses elements of the project related to the exploration, analysis and development of use cases based on datasets found on Melbourne Open Data. The DESIGN team focusses on the functionality, design and usability of the site. The PROJECT team focusses on supporting the needs of each team and providing assistance when needed.

First, the project team meets to discuss the overall schema for the project. Second, the web development, data science and design teams work separately in developing their products. Once the initial draft of the product was ready, the teams communicate amongst themselves to complete shared tasks such as walkthroughs of the use-case, plans for integration and documentation.

We use Trello as a productive way of keeping issues transparent and communication flowing among the different teams to ensure a successful project.

Trello is being used to establish the goals to be met and tasks to be completed as well as visualise the progress of each team. GitHub is used to store our source code. Finally, Microsoft Teams is used to communicate between teams, between individuals and to run online meetings.

The teams are self-organising, so how the work is distributed and who takes charge of the work is decided and discussed within each team.

## New Tasks

The overall vision of the product is established in the first few team meetings to determine the goals and deliverables to be accomplished at the end of the trimester. Teams are then given the autonomy on how they go about creating new tasks.

For the DATASCI team, the datasets available on CoM’s website are reviewed and new use cases are developed based on the problem statement. The tasks are created on the Trello board and put in the “Product Backlog” stage. It is encouraged that each user story on the Trello board should follow the template outlined here <https://trello.com/c/Z3Gvf4Pz>.

We mark each user story with the relevant banner to distinguish WEBDEV, DATASCI, DESIGN AND PROJECT tasks.

Figure 9.1: Team Banners



 



As soon as tasks are created, a Trello card should be created and placed in the “Entire Backlog”. From there cards that are being worked on in the current sprint are added to the “Current Sprint-WIP”. Once the task on the card has been completed it gets moved to the “Sprint 1-Completed” stage. Any tasks which do not get completed within the sprint are moved into the “Current Sprint-Backlog” stage. Members who are involved in that specific task are added to the card. Each card has a checklist of tasks which need to be completed in order for the card to be considered complete. Cards may also have a priority rating: low, medium or high, depending on their importance. Each card can also include a due date if there is a specific timeframe in which the task must be completed by.

Figure 9.2: Trello board stages

Graphical user interface, text, application, chat or text message

Description automatically generated

## Task Review

Work review and testing are required before the task is declared to be complete. The individual(s) allocated to the pull request in GitHub are responsible of reviewing the work. Once the peer is satisfied with the work, they will approve the work in GitHub.

After implementing the changes, the individual would then ask squad-members to re-check their work. The responsible reviewer(s) allocated to the task verify that best practises have been exhibited, in terms of committing on GitHub, code quality and comments in Trello. When all the criteria are satisfied, this task is done and moved to the “Tasks Completed” stage.

## Testing

Product testing is completed by ensuring that different aspects of the product match the ‘acceptance criteria’ under the relevant Trello card. By comparing the product features to this acceptance criteria, the team can be certain that the product is of high quality. As such, we were able to use Trello effectively to enhance the product feature. An example of this has been shown below:

Text

Description automatically generated

Chart, bar chart

Description automatically generated

This process allows the team to be confident that the accomplished product meets the acceptance criteria and is ready to be implemented into the master.

User testing is performed on local machines before deploying to the website. When the feature is deployed, another test is performed on the live website to make sure the product is deployed as per the acceptance criteria.

## Branching Strategy

Team members are encouraged to create their own branch if they wish to contribute to any work instead of committing directly to master so that wrong commits that might break the web application can be prevented.

Each team creates a new branch via GitHub for each new feature to be developed upon starting a new user story. The created branch name will be decided within the team and updated in the description of the Trello card.

During this project, it was recommended that every project member commit their work to the branch after the successful completion and review of allocated tasks. Additionally, it was recommended that any relevant comments be made in GitHub. This allows other members of the project to understand the completion status of the work. The commit message should contain enough high-level information to inform the reader of the new features which have been added and what has changed. This also allows members to understand the progression of the work.

Once the feature has been tested locally and understood to work, the branch is peer reviewed using the pull request function on GitHub. After at least one approval, the branch is merged into the master, either by the reviewer or the requestor. All commits made to the branch or to master should be a working copy of the web application. Rebasing of commits is discouraged so that the git history is not unnecessarily re-written.