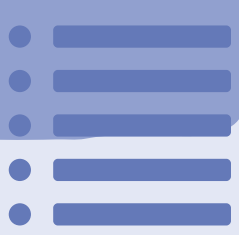
TC Reviews

Software Process Model

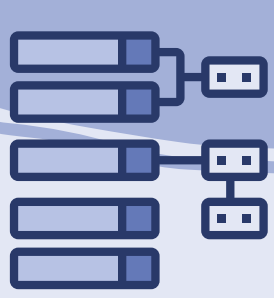
Research similar software products

Generating User Stories

Identifying features and creating product backlog



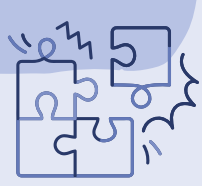
Selecting features from backlog, breaking down features into tasks



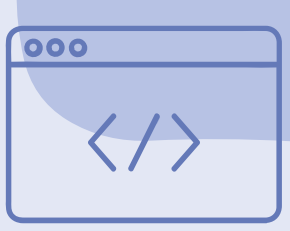
Assigning tasks/Timeline

- **Tracking through kanban board**

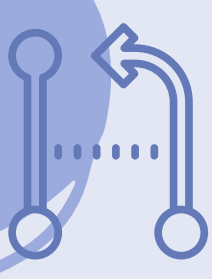
Identifying knowlege and skills of group members, helps to assign tasks better.



Coding/Implementation



Pull request and Merging



Code Review:

- *Generating new ideas*
- *Identifyng Issues*

Testing and Correction

Performence Review:

- *Reflections*

Working sofware for the sprint

Notes:

Improvements made since sprint 1:

1. Selecting features from backlog, breaking down features into tasks

Initially, we lacked a clear process for identifying necessary activities and subdividing them into manageable tasks. This oversight led to uneven workload distribution among team members. In the current sprint, we have enhanced our task breakdown process, assigning tasks based on individual team members' skills while still collaborating as required.

2. Tracking through kanban board

We have adopted the use of Kanban and Notion for more effective task tracking. Our Notion board, which is linked in the README, provides a comprehensive overview of pending tasks, thereby improving our project management and workflow visibility.

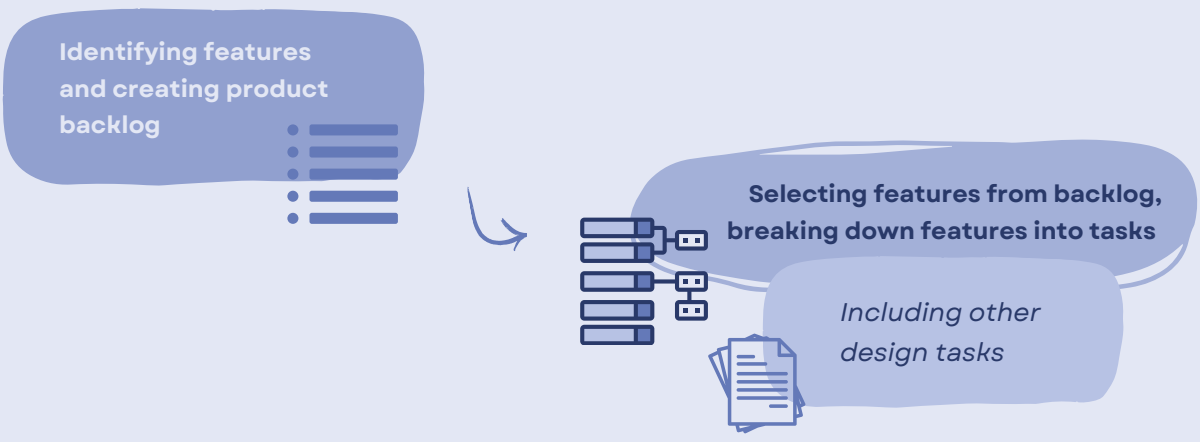
3. Code review

In Sprint 1, our code review process was not adequately defined. For Sprint 2, after consulting with our professor, we implemented a systematic review process during pull requests. This change has significantly improved our understanding of the methods and features we are developing. It has also sparked new ideas and highlighted potential improvements for subsequent sprints.

How might we improve our process in the future:

1. Prioritizing Design Documents and Documentation:

Currently, we tend to postpone our documentation and design tasks, starting sprints primarily with coding. This approach needs to change, as we often struggle to fully complete documentation tasks. A more balanced approach, where design and documentation are given equal importance from the start of the sprint, will likely yield better results.



2. Unit testing

Although our task breakdown for features includes unit testing, we have been more focused on implementing the features and conducting server run tests to verify method functionality. However, we have not been diligent in writing unit tests beforehand. To enhance our development process, we need to prioritize writing and passing unit tests before merging any pull request. This will ensure a more robust and reliable codebase.