



Simple Store

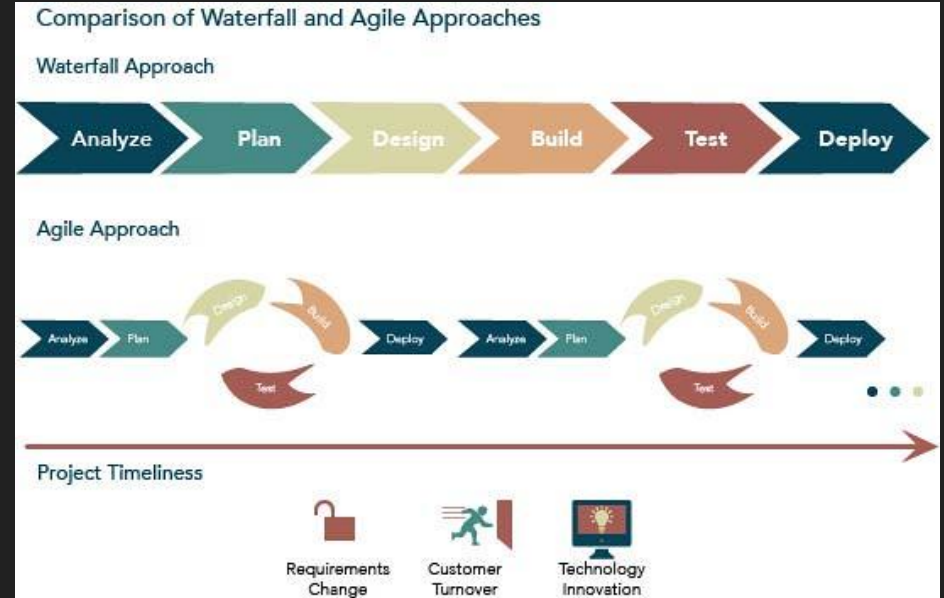
A cloud based storage system deployed
on Localhost.

Joe Rickard, Luke Ingalls, Aiden Prohaska, Nick
Julander, Matthew Spallas

Tools Used

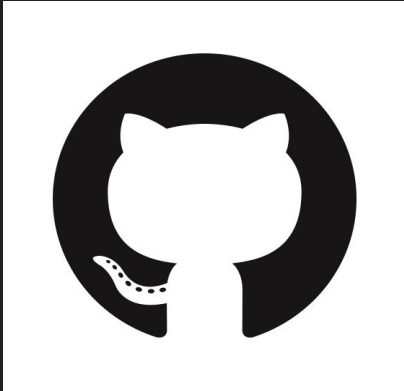
Methodology

We developed using agile. We rated this as a 3 because it helped us stay organized for the most part but at times we found it distracting and not helpful.

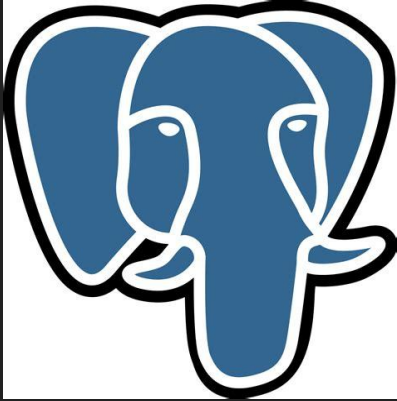




Bootstrap: CSS framework directed at responsive, mobile-first front-end web development. We used this to format our HTML pages and would rate it 5 stars as it helped us create appealing webpages much faster and easier than if we did all the CSS ourselves.



GitHub - Version control platform and project tracker. We used this to collaborate on our project. We gave git/github 4 stars. We had some issues with merging that were caused by files that were changed by the webserver to cater to the user's local machine. These couldn't be resolved with .gitignore and became an issue. All other aspects were good -- even the kanban board.



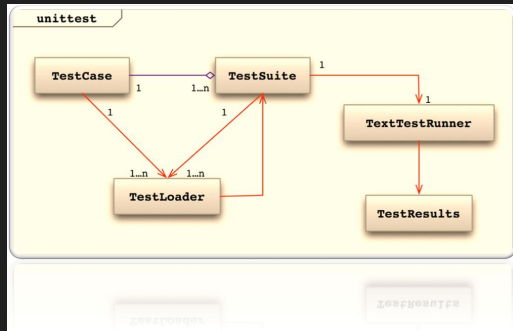
PostgreSQL: Open-source relational database management system emphasizing extensibility and technical standards compliance. We used this to store user and file data. We gave postgres 5 stars as we encountered no issues and it worked exactly as expected.



Flask: A python wrapper for an Apache server. We used this to receive and process http requests from both the GUI and CLI. Flask received 5 stars from the group. It makes the webserver much more simple and has extensive documentation that make working with it very enjoyable.



SQLAlchemy: an open-source SQL toolkit and object-relational mapper for the Python programming language. This was used so the flask server could interact with our database. We rated this as a 3 because while it got the job done the interaction with sqlalchemy is not intuitive and at times just confusing.



Unit testing: For this project we did manual unit testing. Each function was tested based on criteria we put in milestone 5 document. This got us to our working product without issue (performing tests) so we gave it 5 stars.



Localhost: For deployment we used localhost because this was the easiest to accomplish given the time constraints of the project. This gets a 5 out of 5 since it was easy to use and had no problems.

Challenges

Challenges

1. We were going to initially deploy on AWS which turned out to be a difficult process and was not free.

Solution: We deployed on localhost so that we could deploy our code and see whether it worked without the headache of dealing with AWS.

Challenges

2. We had a hard time getting our code to work across all of our machines because of how postgres interacted with sqlalchemy

Solution: We tested and developed the product on macs so that we were working in a consistent environment. If we were to continue with the project we could then get into compatibility issues.

Challenges

3. We had problems merging our repositories.

Solution: We used the .gitignore to try to alleviate some issues that were avoidable. Beyond this we just tried to meet when major merges were happening so that it would be done correctly from the beginning.

Challenges

4. Sending a file and login info in the same http request. Also making http requests that were consistent for the CLI and GUI

Solution: For logins we started having the server store them so that a user could login and use the server without having to re authenticate each time. For consistency issues the biggest issue was how data was sent from the html page. Buttons had to be changed to p tags styled like buttons and some extra endpoints had to be added in the webserver to make certain actions interface specific.