**PROJECT PLAN**

**STOCK PRICE PREDICTOR**

**Team Details:**

**MIHIR JAYAPRKASH PES2UG20CS196**

**MITUL JOBY PES2UG20CS199**

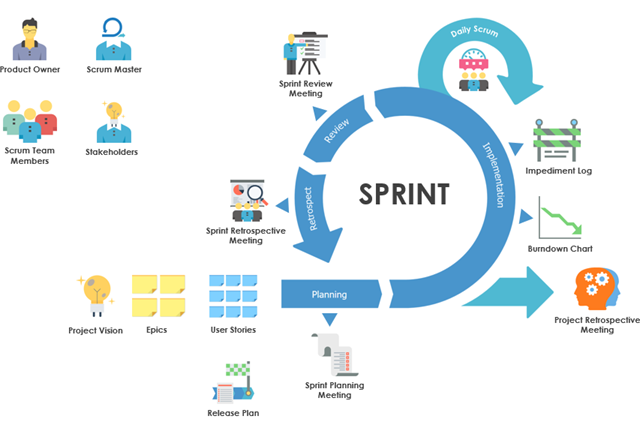
**NAVYAE GOYAL PES2UG20CS219**

**NOEL JACOB ABRAHAM PES2UG20CS234**

**1. Identify the lifecycle to be followed for the execution of your project and justify why you have chosen the model.**

The life cycle followed for the execution of our project is the Agile Scrum. This methodology involves multiple iterations of certain steps and a daily scrum to update each other on the progress.

These iterations are the sprints and are pre decided before starting the project. We follow a continuous deliverable format which verifies tasks for the day and if the sprint completion is up to date.



**2. Identify the tools which you want to use throughout the life cycle like planning tool, design tool, version control, development tool, bug tracking and testing.**

Planning tool: JIRA

Design Tools: PowerPoint, Canva

Version Control: git

Development Tool: VS Code, Python

Bug Tracking: JIRA

Testing: Selenium

**3. Determine all the deliverables and categorize them as reuse/build components and justify the same**

* Login Page (Client, Admin): Reuse

There are many existing login page components that we can use and change to our requirements.

* Update the databases with companies’ stock: Build
* Try to add as many companies’ stock prices to the database: Build

A simple process of adding data to an existing database and connecting them

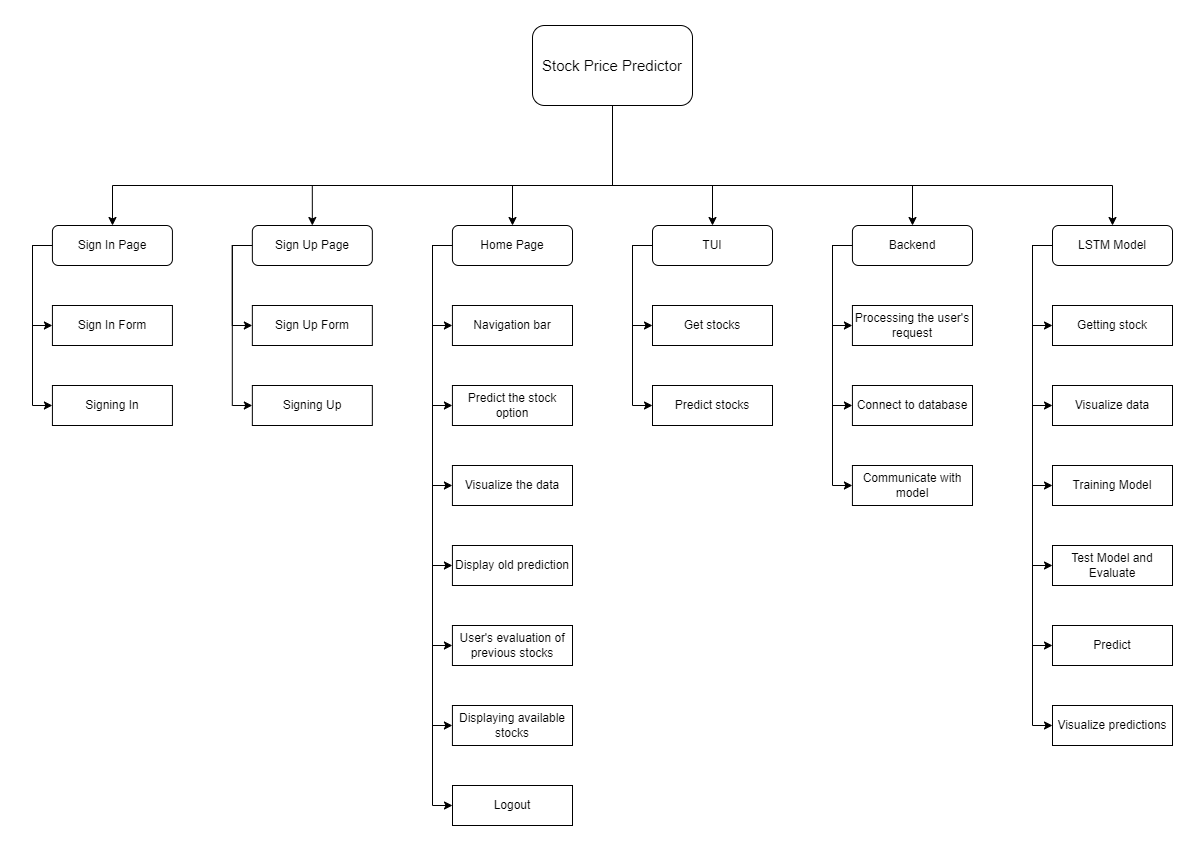
* Generate plots for historical stock prices: Build

Deliverable that takes care of generating historical data such as closing price, opening price, high and low and then based on that prints the stock prices.

* Generate future prices of the stock: Build

Generating the future prices of the stock using machine learning.

**4. Create a WBS for the entire functionalities in detail**

****

# Rough estimate of effort required

Organic models are used when the problem statement is well understood and have been used in the past, and works well with small teams. The module selected for testing was the LSTM model.

This model was selected because it is the core module and is a dependency for other modules.

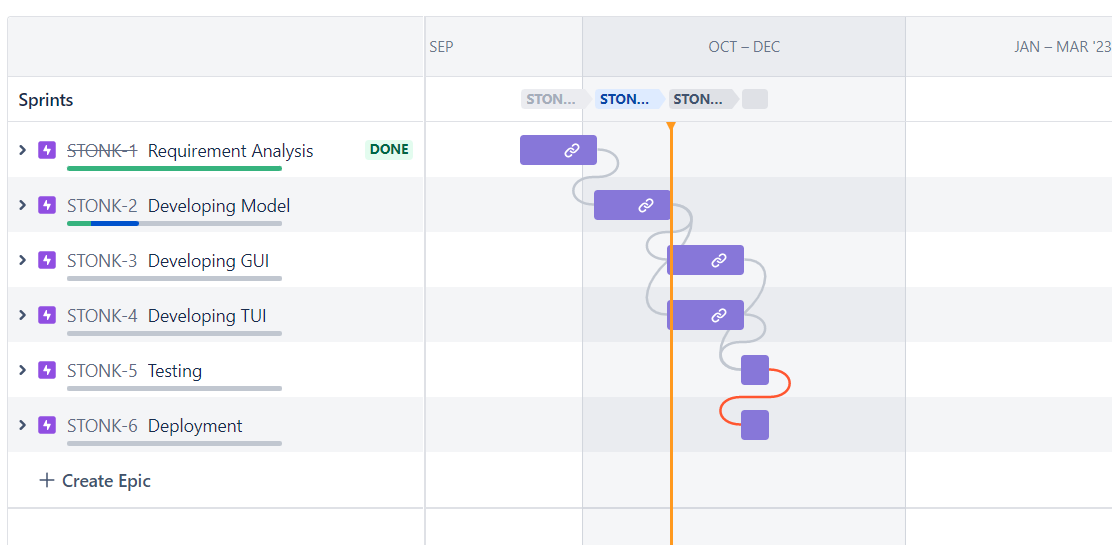
KLOC = 2

a = 2.4

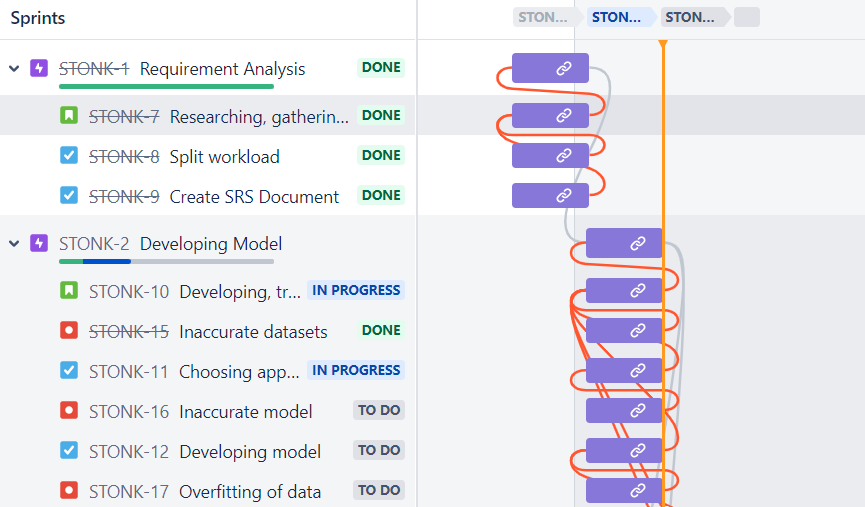
b = 1.05

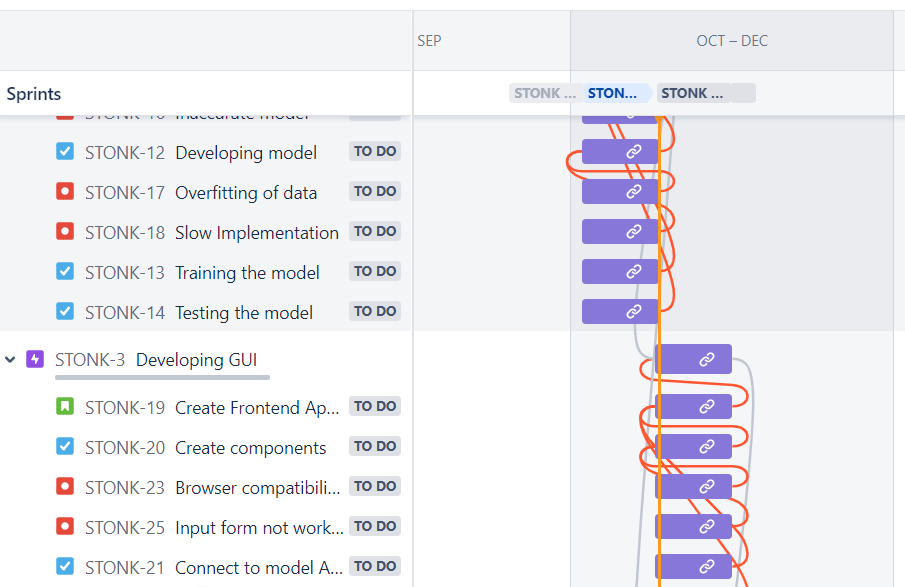
Effort = a\*(KLOC)^b = 4.96

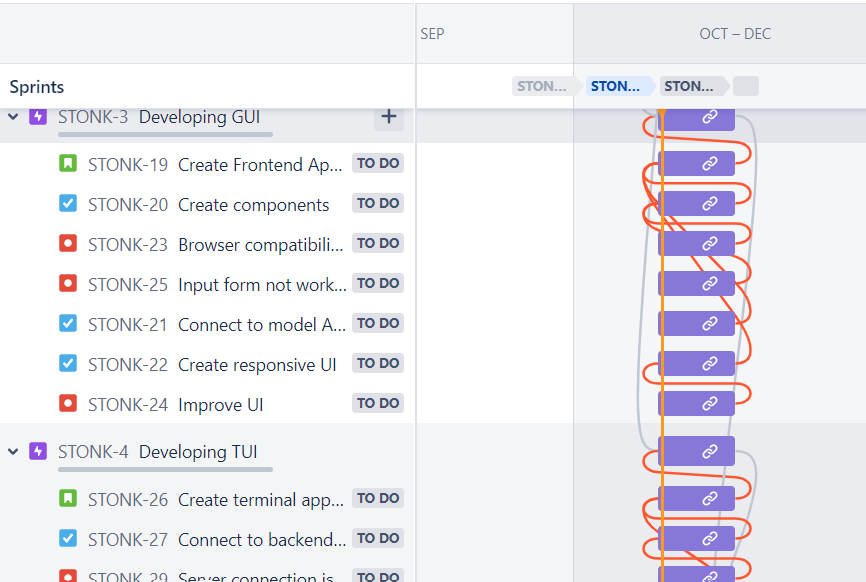
**Gantt chart**

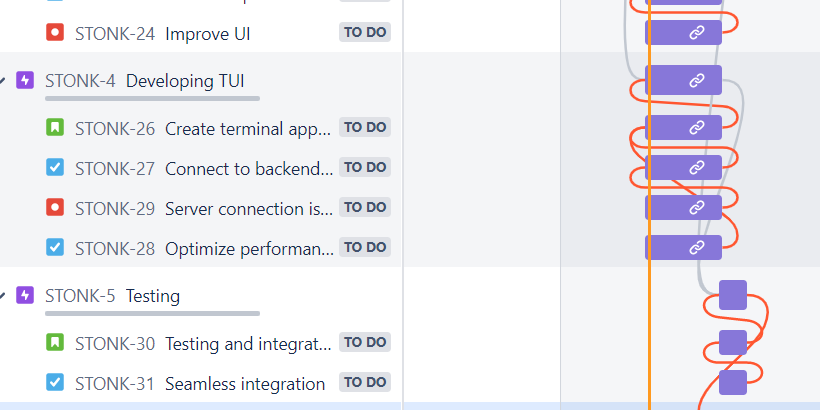
****

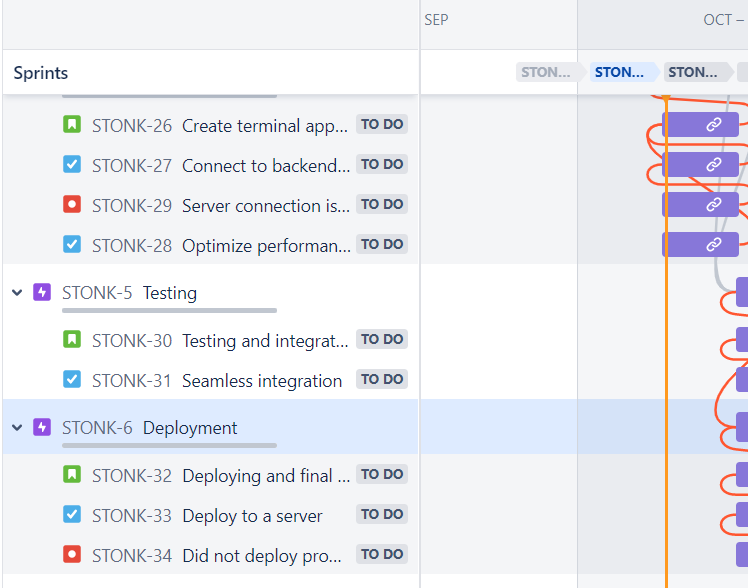
**Sprints**

****

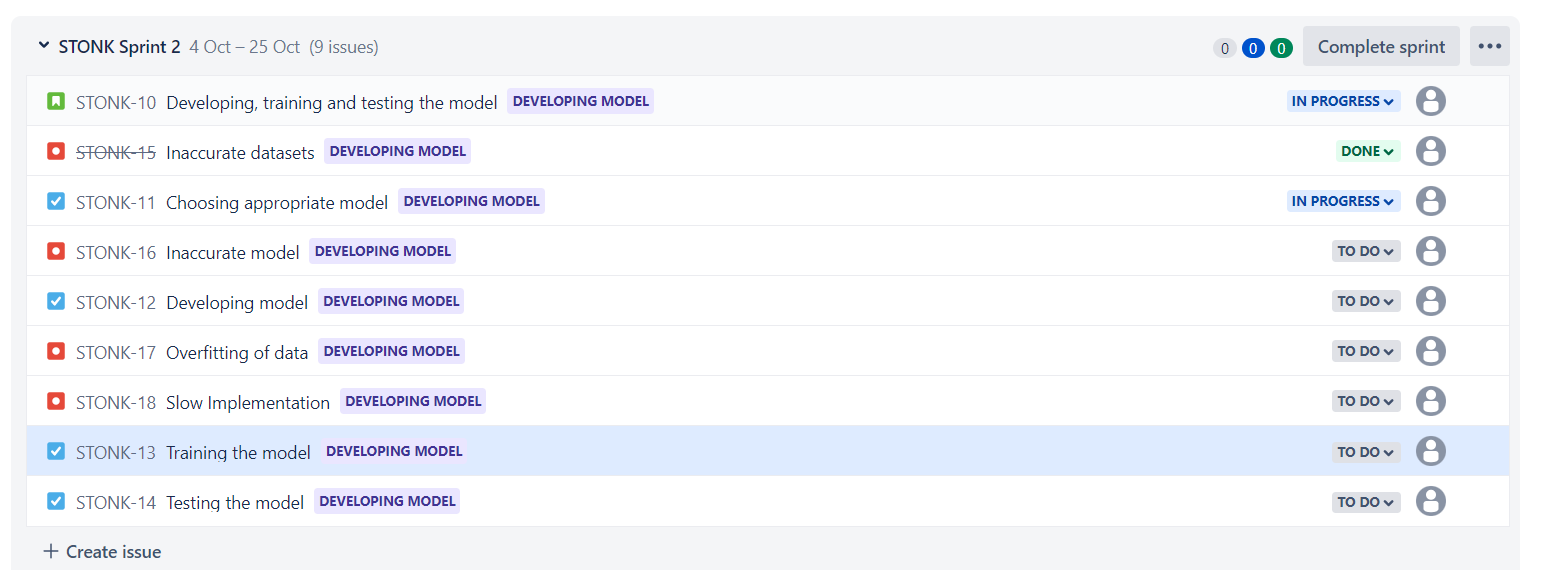
****

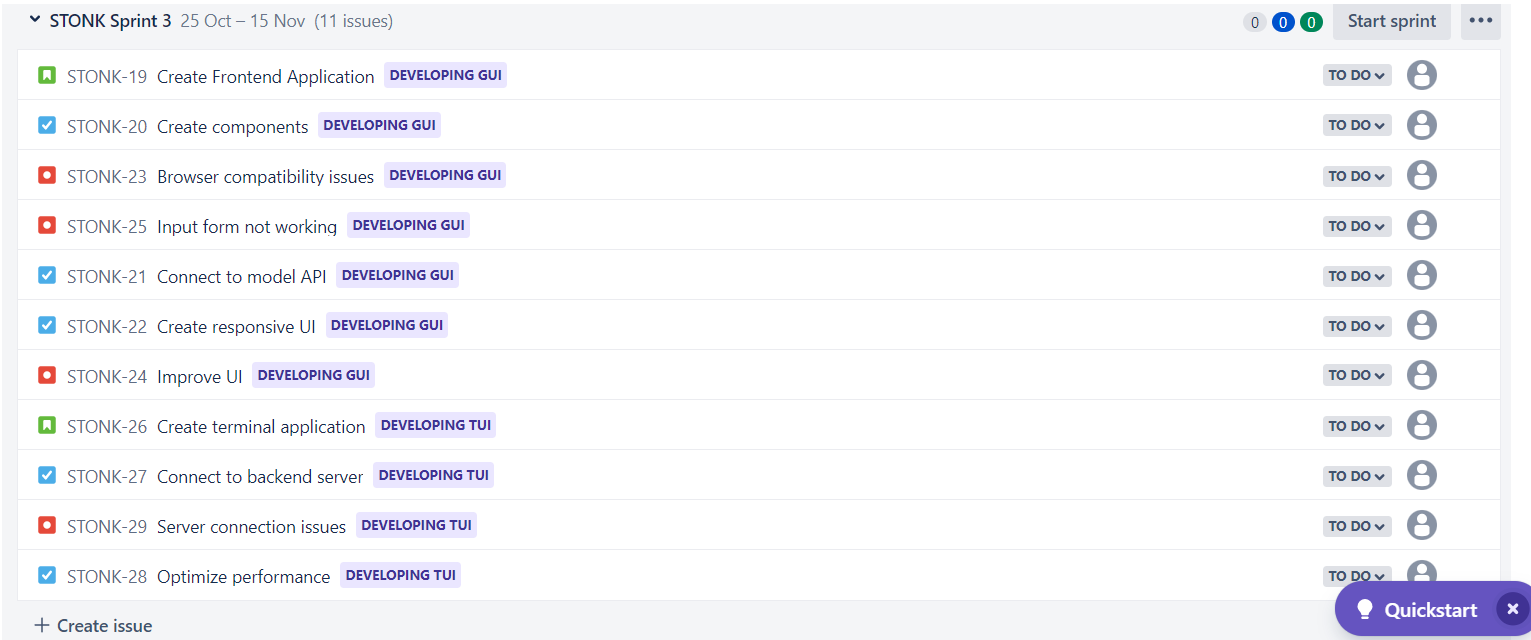
****

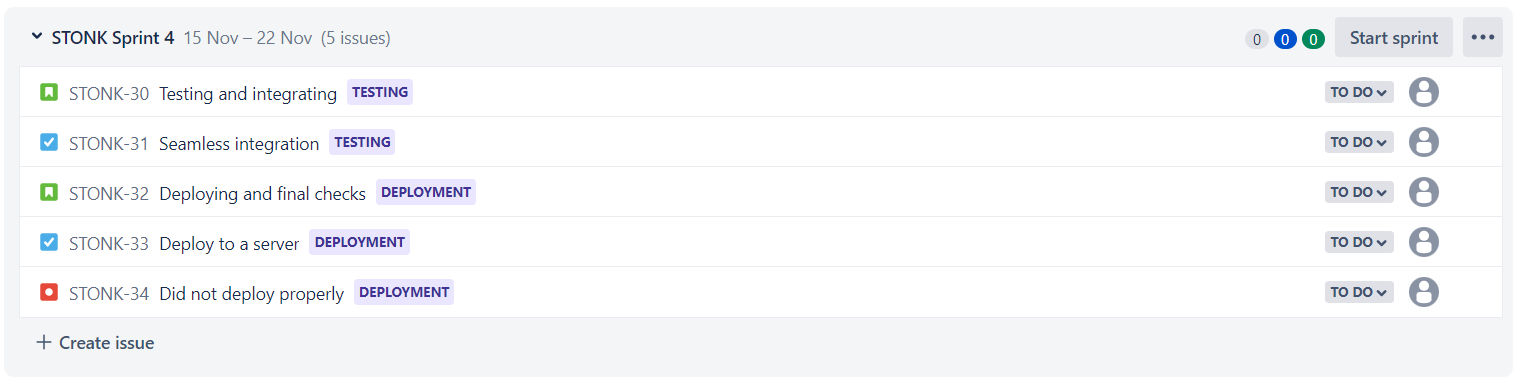




**Product Backlog**

****

****

****