# PROJECT PLAN for Version Control System

Prepared by:
Hridhay Kiran Shetty PES1201800068
Ashish Harish Shenoy PES1201801447
Manu M Bhat PES1201801452

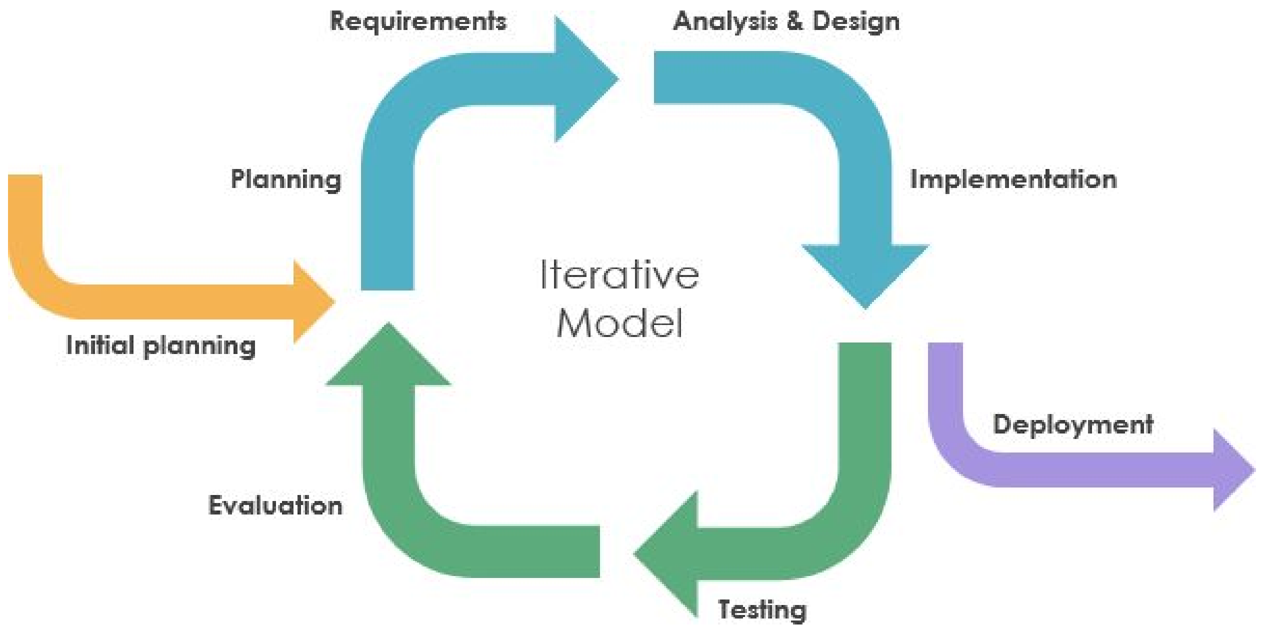
## Table of Contents

## 1. Project Lifecycle

The lifecycle we intend to follow for the execution of our project, of version control system,  is the **Iterative Model**. We intend to do so for the following reasons :

- The SRS strictly defines the requirements of the final product
- While the main software deliverable is predefined by SRS, it may advance over time and may need to be adapted to the modified requirements.
- Easier to track problems, defects and any other issues and solve them effectively.
- By getting user feedback and improving iteratively, we can assure the project is upto the expected standards and satisfactory to the users.
- This approach also reduces the documentation overhead and hence ensuring more time towards the design, development and testing.

**Lifecycle Diagram** :



## 2. Tools to be Used

### 2.1. Version Control:

- GitHub

### 2.2. Design Tools:

- StarUML
- draw.io
- LucidChart

### 2.3. Development Tools:

- Vscode
- Python3
- Postgresql

### 2.4. Planning Tools:

- Trello

### 2.5. Bug Tracking:

- JIRA

### 2.6. Testing Tools:

- Katalon Studio
- Postman for testing REST API's
- Unittest

## 3. Deliverables:

We intend to deliver a version control system, it would allow users to carry out software development with usage of source code management, collaborate and maintain their project in a more efficient manner. By also providing a web app,we allow users to also store their code in a remote repository, hence providing a back up if necessary and making collaboration easier for developers.

Following are the deliverables of out project :

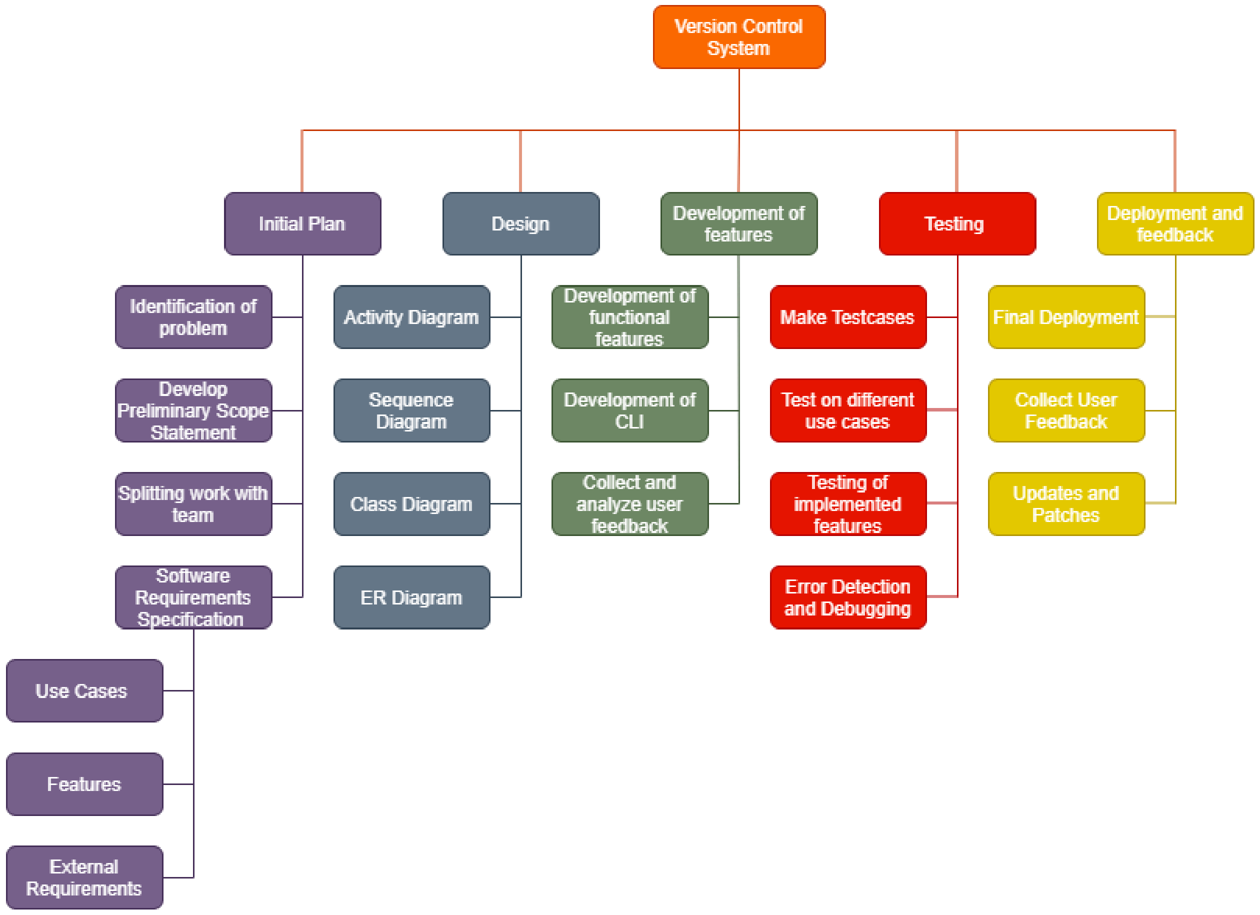
1. Version control :

    a. Creating repository
    b. Commiting changes to repository
    c. Push changes to remote repository
    d. Clone remote repository
    e. Pull changes made on remote repository

2. WebApp:

    a. Landing page
    b. Signup and login page
    c. Repository viewing page
    d. Source code reader
    e. User profile
    f. Commit history view

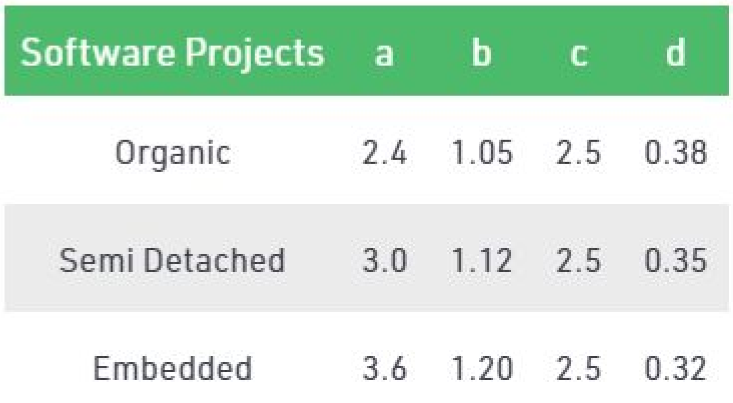
## 4. Work Breakdown Structure :



## 5. Estimation of Efforts :

Constructive Cost Model:

The Constructive Cost Model is a procedural software cost estimation model.The model parameters are derived from fitting a regression formula using data from historical projects. The various parameters used are the likes of size, effort, cost, time and quality.

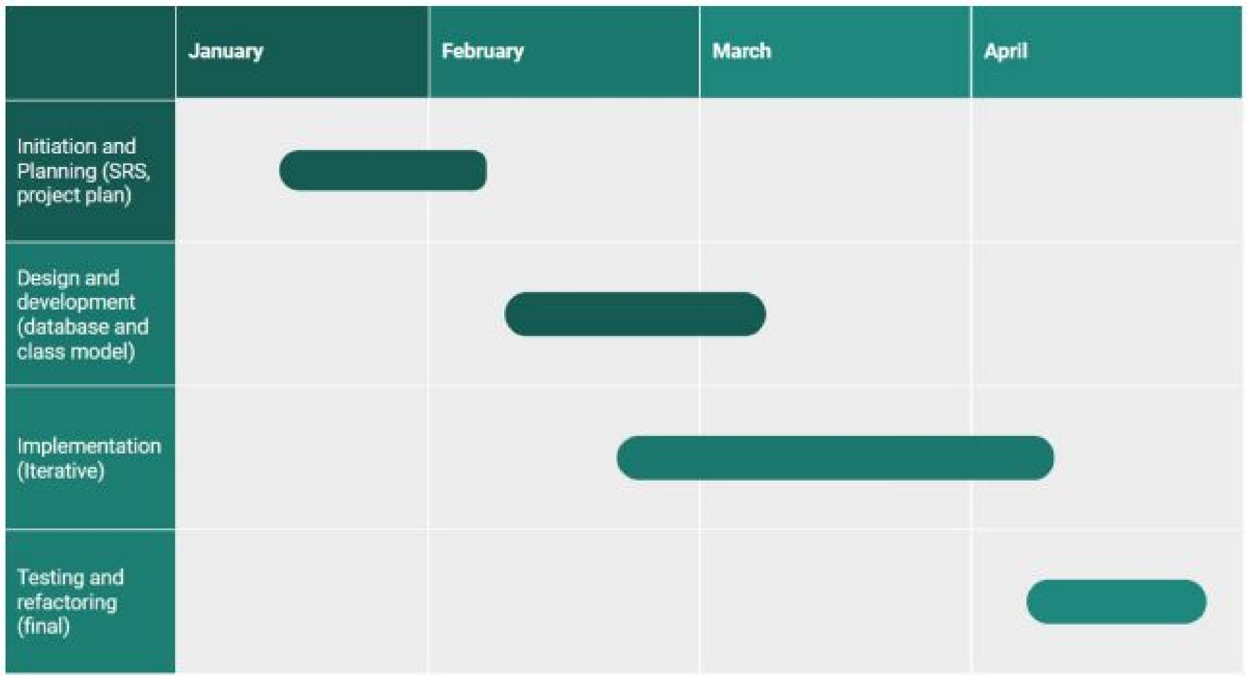


| Software Projects | a | b | c | d |
|---|---|---|---|---|
| Organic | 2.4 | 1.05 | 2.5 | 0.38 |
| Semi Detached | 3.0 | 1.12 | 2.5 | 0.35 |
| Embedded | 3.6 | 1.20 | 2.5 | 0.32 |

We can classify the project as Organic since the requirements are lucid and the team size is small.

- Kilo lines of Code: Approximately 5,000 lines = 5K lines
- Effort = a (Kilo lines of Code) $^{b}$
  Effort = 2.4 x $(5)^{)1.05}$
  Effort = 13.005 pm
- Duration = c $(E)^{d}$
  Duration = 2.5 x $(13.005)^{0.38}$ = 6.26 months
- Team Strength = 3

## 6. Scheduling


January
February
March
April
Initiation and Planning (SRS, project plan)
Design and development (database and class model)
Implementation (Iterative)
Testing and refactoring (final)