Project Planning

for

Code Management and Version Control System(Github)

Prepared by

A R Prashanth PES1201800410

R Sree Harsha Reddy PES1201800233

Nikhil N Gowda PES1201802023
Team G16

PES University

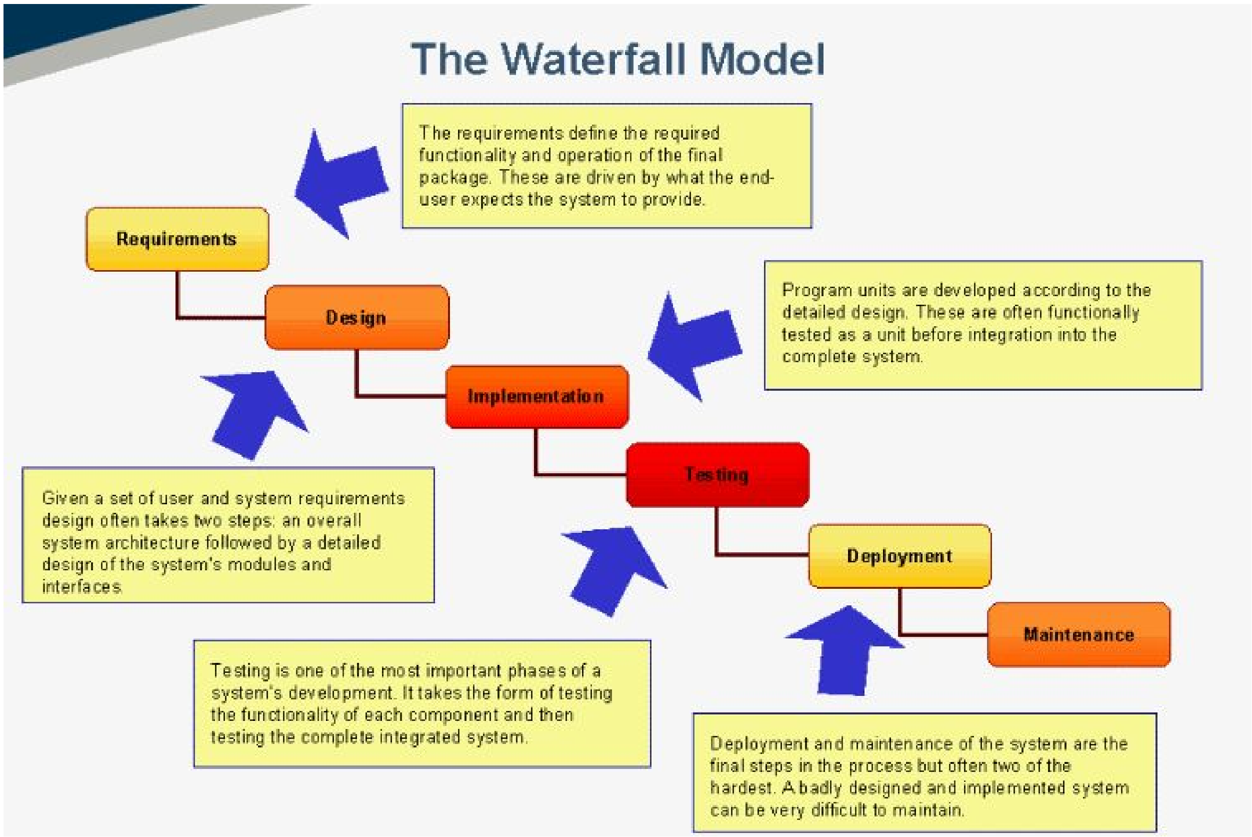
16-Feb-2021

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

Waterfall Model

We are selecting the waterfall model because all of the necessities are recognised earlier, clean and not speculated to alternate in future and the problem assertion is strong, technology is thought, no ambiguous requirements and also the project is brief (four months).The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.



The Waterfall Model

The requirements define the required functionality and operation of the final package. These are driven by what the end-user expects the system to provide.

Program units are developed according to the detailed design. These are often functionally tested as a unit before integration into the complete system.

Given a set of user and system requirements design often takes two steps: an overall system architecture followed by a detailed design of the system's modules and interfaces.

Testing is one of the most important phases of a system's development. It takes the form of testing the functionality of each component and then testing the complete integrated system.

Deployment and maintenance of the system are the final steps in the process but often two of the hardest. A badly designed and implemented system can be very difficult to maintain.

The Sequential phases in Waterfall model are:

- **Requirement gathering and analysis-**All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** − The requirement specifications from the first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

Some of the major advantages of the Waterfall Model are as follows −

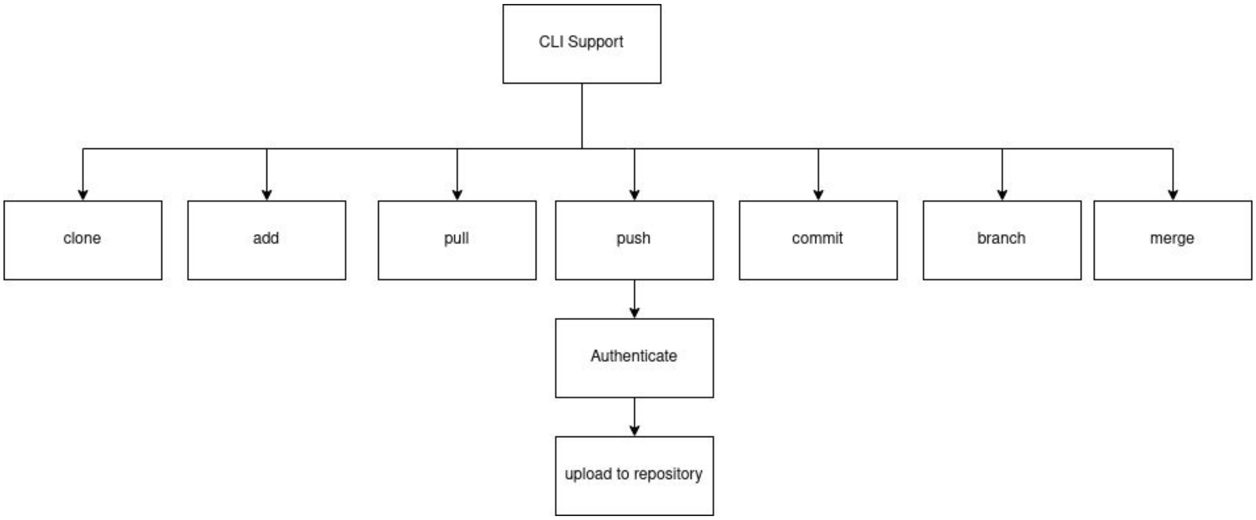
- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

2: Identify the tools which you want to use at different phases of SDLC like planning tool,design tool, version control, development tool, bug tracking and testing tool.

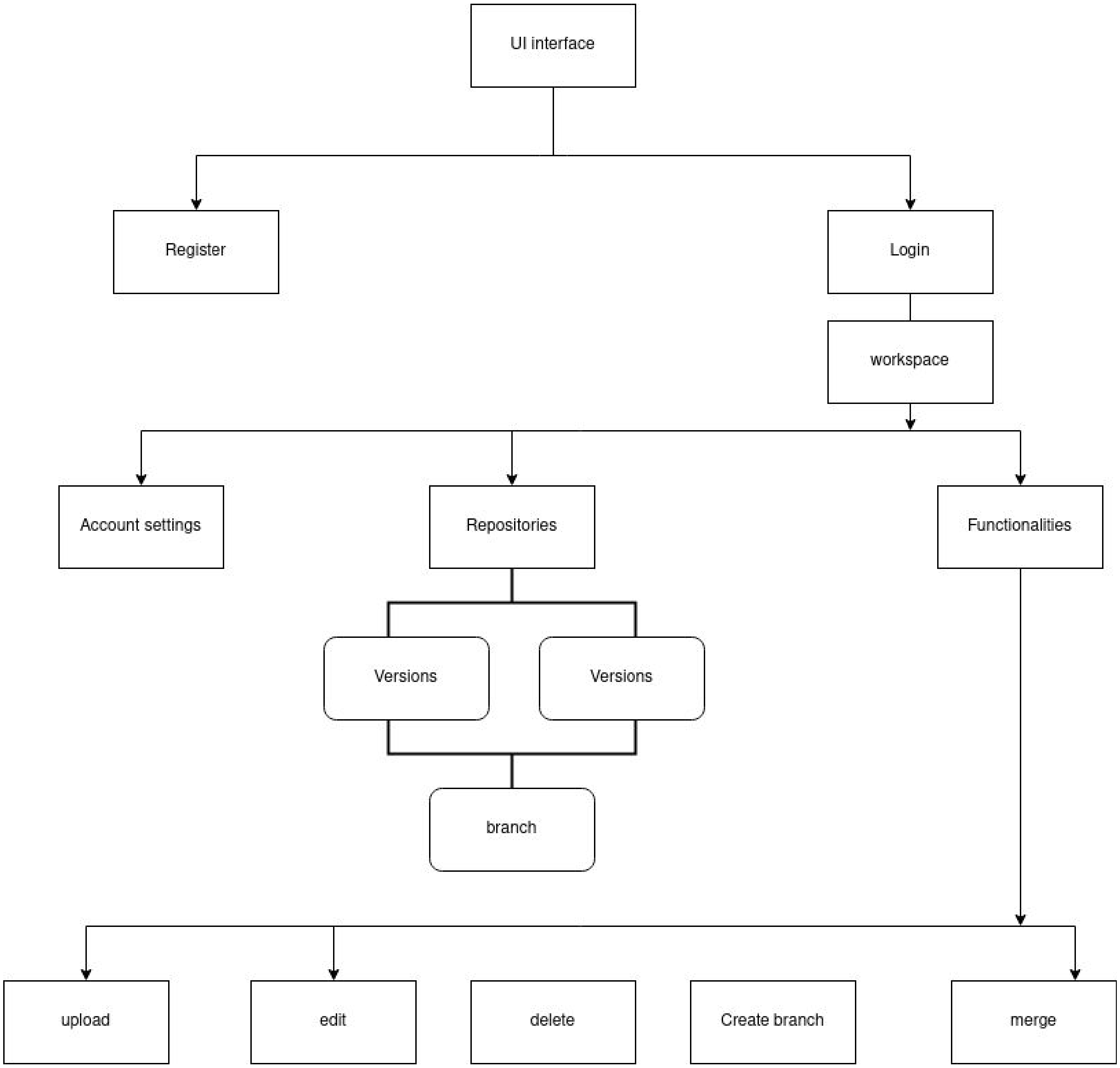
1. **Planning tool -** Discord, WhatsApp and Google meet are regular web platforms used for discussions and meetings.
2. **Development tools -** VSCode and sublime text are robust IDE used for writing codes effectively and also have inbuilt support for all the used languages.
3. **Version Control and lifecycle planning -** GitHub Projects will be used to organize the work flow between developers and to track feature implementations
4. **Use case diagram visualisation tool -** StarUML is the preferred choice as it supports most of the diagrams specified in UML 2.0
5. **Gantt chart preparation tool –https://www.canva.com** is used for gantt chart preparation
6. **WBS chart preparation tool –** Flow chart maker, as it allows users to create a task list quickly and then rearrange their priority, add details, divide further into sub-tasks and assign to team members

3: Create a Work Breakdown Structure for the entire functionalities of your project in detail.

CLI Support



UI interface



4: Determine all the deliverables and categorise them as reuse/build components and justify the same.

Design for the software

- High Level Design - Build
  A basic understanding of the features provided by the application is Documented.

- Detailed Design
  A detailed explanation of the project features are documented including its action diagrams and use case diagrams.

Development

- Front end-login and registration page: reuse
  Front-end login and registration design can be inherited from the templates available online.

- Frontend home page - Build
  The homepage has to be built from the scratch based on the requirement .

- Backend - Build
  The database and other scripts are built from scratch based on the scope of the application.

- Integration Plan - Build
  The frontend and backend would be combined using necessary frameworks.

- CLI Support - Build
  command line support has to be built and compiled with respect to OS.

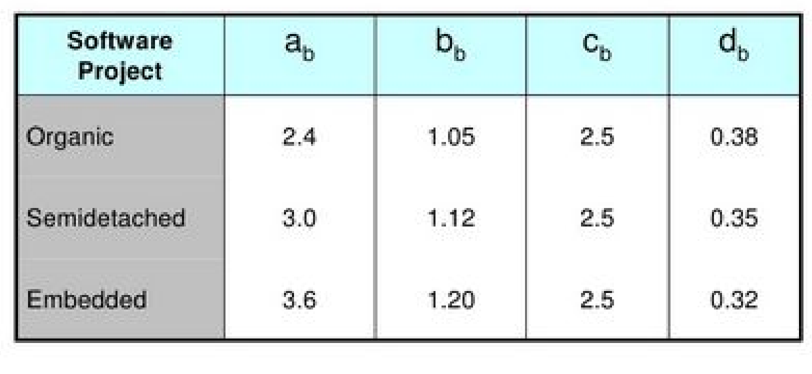
5 : Do a rough estimate of effort required to accomplish each task in terms of person months.

COCOMO(Constructive Cost Model) is a regression model based on lines of code.This model predicts the various parameters involved in a project such as effort,total project cost and scheduled time.

In COCOMO,projects are categorised into three types:

1. **Organic** : A development project can be treated of the organic type, if the project deals with developing a well-understood application program, the size of the development team is reasonably small.
2. **Embedded** : A development project is treated to be of an embedded type, if the software being developed requires the highest level of complexity, creativity, and experience.
3. **Semi-Detached** : A software project is said to be a Semi-detached type if the vital characteristics such as team-size, experience, knowledge of the various programming environments lie in between that of organic and Embedded.

Value Chart for COCOMO Model:



Software Project	a_b	b_b	c_b	d_b
Organic	2.4	1.05	2.5	0.38
Semidetached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Type of Project :

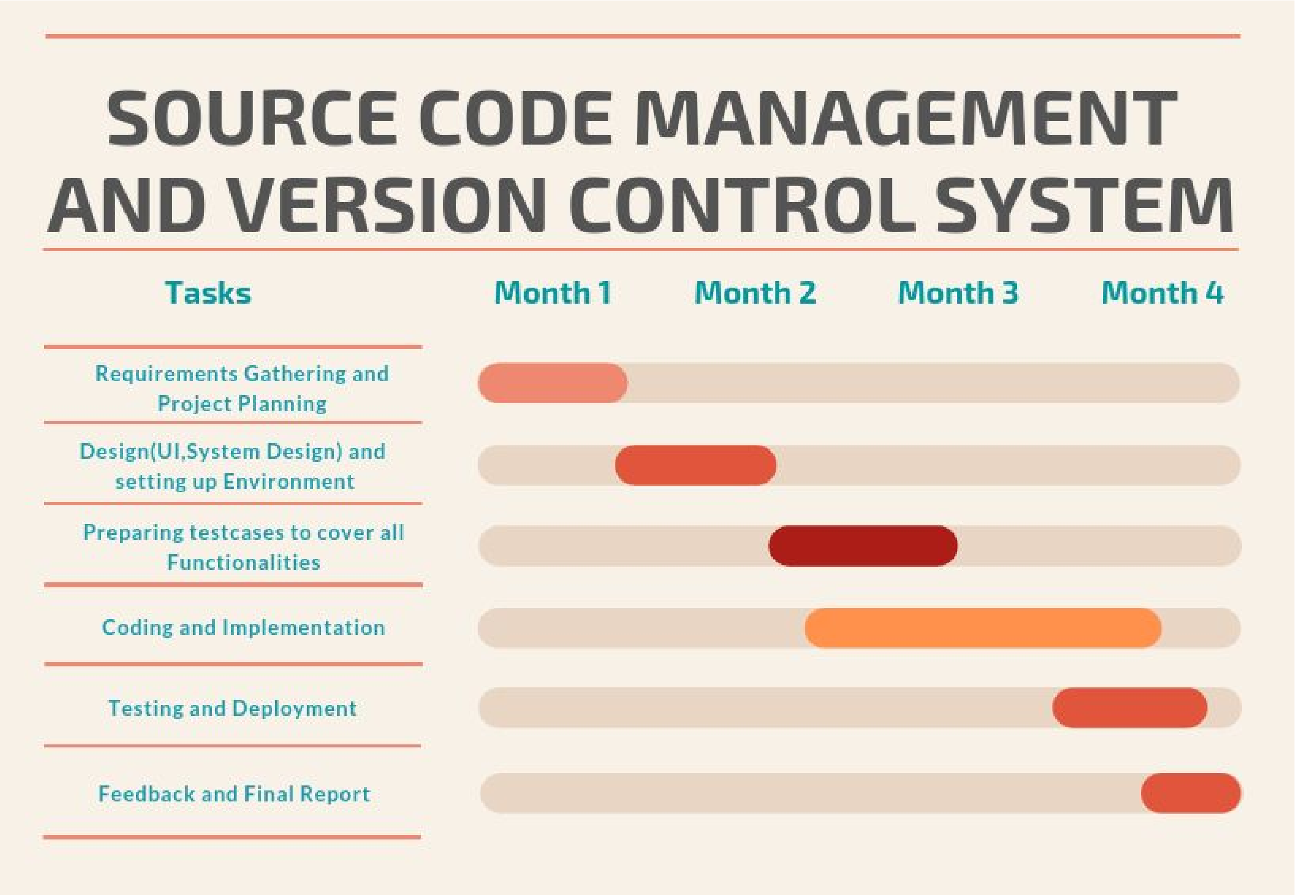
Our project falls under Organic as the team size is small.

Estimation of Effort:(Basic COCOMO)

- a,b,c,d are constants for each project type
- For Basic COCOMO a = 2.4 b = 1.05 c = 2.5 and d = 0.38
- Calculation:
  - Kilo lines of Code = 3k Approx

- Effort = a*(Kilo lines of code)^b
  = 2.4 * (3)$^{1.05}$
  = **7.6065 pm**
- Team Size = 3
- Time = c*(E)^d
  = 2.5 * (7.6065)$^{0.38}$
  = **5.4049 months**

6: Create the Gantt chart for scheduling the defined tasks.


SOURCE CODE MANAGEMENT AND VERSION CONTROL SYSTEM
Tasks
Month 1
Month 2
Month 3
Month 4
Requirements Gathering and Project Planning
Design(UI,System Design) and setting up Environment
Preparing testcases to cover all Functionalities
Coding and Implementation
Testing and Deployment
Feedback and Final Report