Project Planning Report
Online Rental Car Management System

Prepared by Hrishikesh Vasant Pilare
PES2201800687
Chapters 1 and 2

Prepared by Ayushi Agarwal
PES2201800053
Chapters 5 and 6

Prepared by Simmi Alok
PES2201800446
Chapters 3 and 4

SECTION-F

PES University, EC Campus

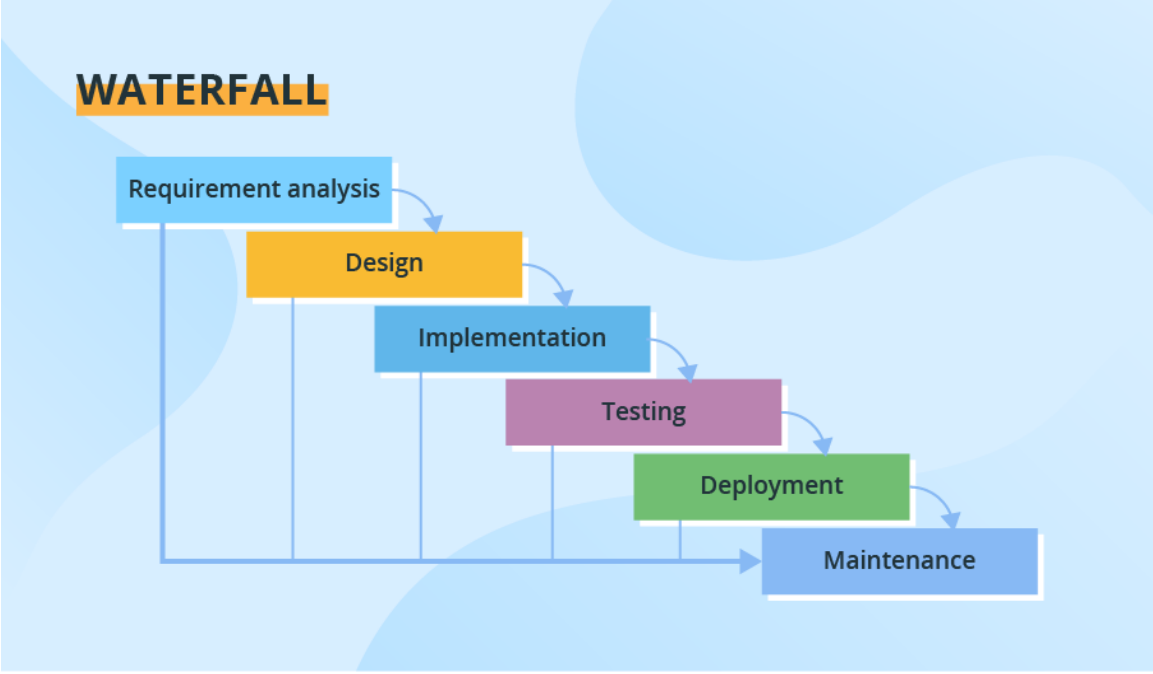
Table Of Contents

1. Software Development Methodology

1.1 MODEL

- After going through various models we have decided that the Waterfall Model is the Software development Lifecycle methodology we plan to follow for our project Online rental car management system.
- Through all development stages (analysis, design, coding, testing, deployment), the process moves in a cascade mode. Each stage has concrete deliverables and is strictly documented. The next stage cannot start before the previous one is fully completed.f the development process.
- Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

1.2 PHASES OF THE LIFE CYCLE



1.3 Advantages

- Simple small or mid-sized projects with clearly defined and unchanging requirements (small company website development).
- Projects that must adhere to multiple rules and regulations (healthcare projects).
- Projects where a well-known technology stack and tools are used.

1.4 Disadvantages

- No ability to see and try software until the last development stage is finished, which results in high project risks and unpredictable project results. Testing is often rushed, and errors are costly to fix.
- Software requirements cannot be re-evaluated further in development.

2. Tools and Techniques

2.1 Design -StarUML

StarUML is an open source software modeling tool that supports the UML (Unified Modeling Language) framework for system and software modeling

StarUML supports the following diagram types that we are going to be using.

- Use Case Diagram
- Class Diagram
- Activity Diagram

2.2 Version Control - GitHub

Version control allows us to keep track of your work and helps us to easily explore the changes that we have made, be it data, coding scripts, notes, etc.

Each file on GitHub has a history, making it easy to explore the changes that occurred to it at different time points. You can review other people's code, add comments to certain lines or the overall document, and suggest changes.

Having a GitHub repo makes it to keep track of collaborative and personal projects - all files necessary for certain analyses can be held together and people can add in their code, graphs, etc. as the projects develop. You can also ask certain users to review your code. For personal projects, version control allows you to keep track of your work and easily navigate among the many versions of the files you create, whilst also maintaining an online backup.

2.3 Development

We will make use of PHP and Apache XAMPP servers for the purpose of log in and sign up.

MySQL will be used to store data of customers,cars etc. in a database.

Java is the basic language that we are going to use to code the front end part of our project.

2.4 Planning - Microsoft Excel

Based on the Gantt chart model, this project plan template in Excel uses a simple visual representation to show how a project will be scheduled and managed over time from January to April.

3. Project Deliverables

3.1 Project Management

- Project Charter : *Reuse*

The project scope and structure has been discussed in our week 1 submission and will be used for the future development.

- Management Plan: *Build*

At its core, a management plan will define the approach and process the team will use to manage the project according to scope. This is to ensure that the long term trust of the software in terms of accessibility and reusability is projected well to the project stakeholders.

3.2 Initiation

- Project planning - *Build*

Given the timeline and resources, a project plan has been made.

3.3 Requirement Analysis

- Requirement Gathering - *Reuse*

Customer requirements taken into consideration during the project build of similar apps in the market like Zoomcar. This approach is inherited by this project in order to understand our clients.

- Requirements Analysis - *Reuse*

The gathered requirements are analyzed for feasibility as done by Zoomcar.

- Security Planning - *Reuse*

Basic security in terms of data integrity is maintained. Any future security threats can be handled by further modelling.

3.4 Design

- High Level Design - *Build*

A good software is one that can be easy to use by the clients (user) and which has self explanatory features.

- Proof Of Concept: *Reuse*

Zoomcar serves as a proof of concept as they are implemented and are feasible.

- Detailed Design - *Build*

A detailed explanation of the project features are documented including its action diagrams and use case diagrams which can be used by successive development and management teams to understand the project and its structure better.

3.5 Development

- Front end: *Reuse*

Front-end design can be inherited from the base design of similar apps in the present market.

- Backend: *Build*

The database is built from scratch based on the scope of the application.

- Integration Plan - *Build*

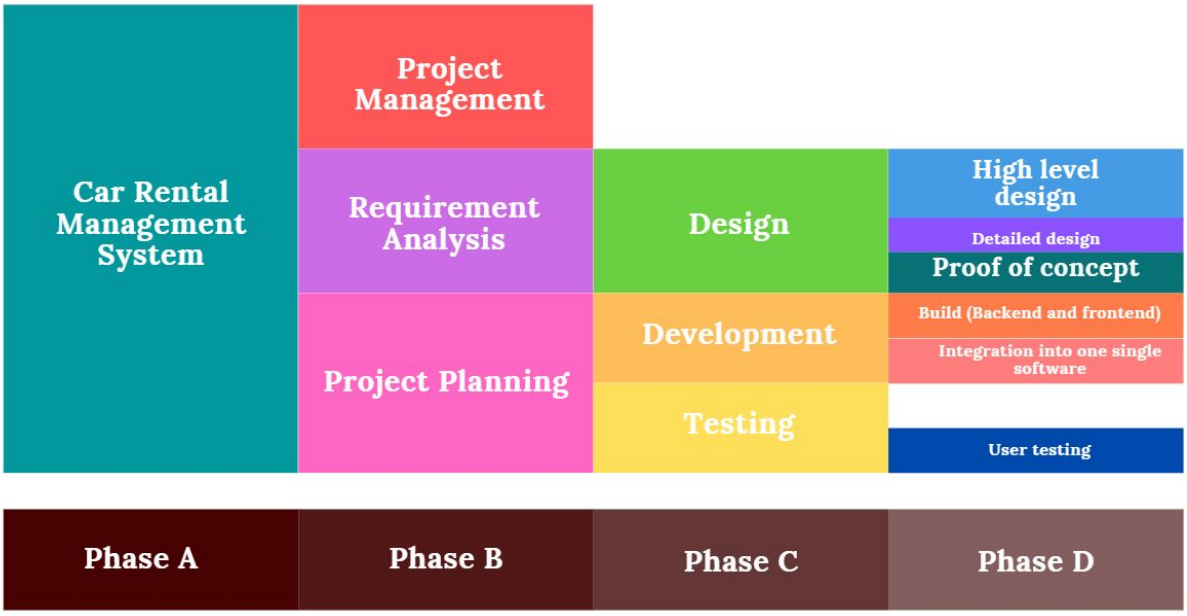
The frontend and backend would be combined using necessary Frameworks.

3.6 Testing

- User acceptance*: Build*

User acceptance testing would be done with a set of users in the proximity of the team and those who helped out with requirement gathering, and any changes to be made would be taken from their responses.

4.WBS -Work Breakdown Structure



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

This project uses the Constructive Cost Model (CoCoMo model) specifically the organic method because the team size is small(3 members in a team) and the problem is well understood.

The effort required to accomplish the task is given by the following formula:

E = a(S)^b Person months

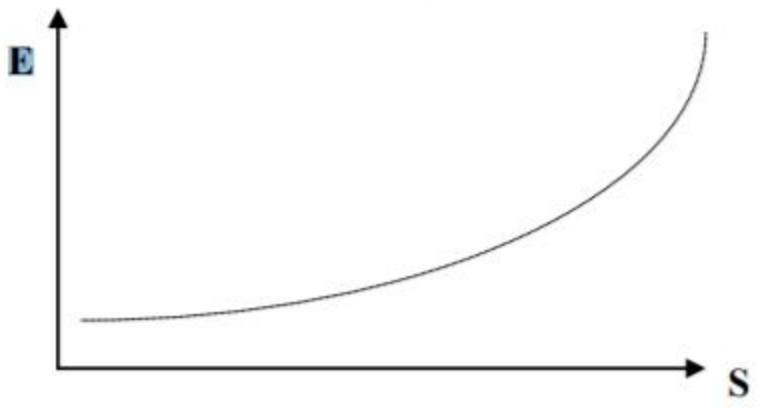
Where:

E represents effort in person-months,

S is the size of the software development in KLOC and,

a and **b** are values dependent on the development mode,

development mode: organic a = 2.4, b = 1.05



This graph demonstrates that the amount of effort(E) accelerates as size(S) increases.

The expected lines of code required for the project:

Effort required for the different phases are as follows:

1) **Design**: Developing the login and register page using HTML and PHP.

   Effort = 2.4 * (95) ^ 1.05 = 287 PM

2) **Coding**: Using Java Programming language to build classes and functions.

   Effort = 2.4 * (500) ^ 1.05 = 1638 PM

3) **Testing**:

   Effort = 2.4 * (60) ^ 1.05 = 176 PM

6: Create the Gantt Chart for scheduling.