



PES UNIVERSITY, BANGALORE
Department of Computer Science and Engineering
B. Tech (CSE) 5th Semester
Aug-Dec 2024

UE22CS341A - Software Engineering
PROJECT PLAN DOCUMENT
Sales Data Analytics

Team : 446_454

PES1UG22CS454	Priyanka Kumari
PES1UG22CS446	Prayashi Verma

1. Lifecycle Model Selection

The **Waterfall Model** has been chosen as the most appropriate lifecycle model for this project due to the clarity of the requirements and the linear nature of the development process. This approach aligns well with the goals and structure of the project, ensuring that each phase of development is completed before moving to the next stage.

Reasons:

1.1 High Degree of Certainty in Requirements

The project's goals, such as creating a sales analytics dashboard with functionalities like data extraction, transformation, visualization, and user management, are clearly defined from the start. This certainty in requirements reduces the need for iterative cycles and prototyping.

1.2 Minimal Scope for Changes

The project scope is clear, and big changes are unlikely during development. The Waterfall Model works well for projects with few expected revisions since it follows a step-by-step process.

1.3 Structured Phases:

The sequential nature of Waterfall allows for well-organized phases such as requirement gathering, system design, implementation, testing, and deployment.

The **high degree of certainty** is associated with this project. The minimal expected changes, familiarity with structured and predictable development cycle make Waterfall the optimal choice for this project.

2. Tools Used Throughout the Lifecycle

2.1 Planning Tools:

Canva: For task management and tracking progress.

2.2 Design Tools:

Canva: To create architectural diagrams such as data flow diagrams(DFD), ER diagram, Use Case diagram.

2.3 Version Control:

GitHub: For code versioning and collaboration.

2.4 Development Tools:

Flask: Backend web framework for building the application.

HTML, CSS, JS: For building the user interface of the dashboard.

MySQL: For handling the relational database.

VS Code: IDE for writing code.

2.5 Bug Tracking:

GitHub: Tracking bugs and change requests.

2.6 Testing Tools

3. Deliverables:

Reusable Components:

- Authentication Module: Can be reused in future applications requiring login and user role management.
- Data Visualization Module: Charts and graph components can be used for other analytical dashboards.
- ETL Pipeline Code: The ETL code for data transformation can be reused for other data processing applications.

Build Components:

- Dashboard Visualizations: These need to be built based on specific requirements.
- Data Warehouse Setup: Sales data storage will be specific to the project's needs.
- OLAP function

Justification: Reusable components are those that provide generalized functionality such as authentication, data pipeline etc, which can be adapted to other projects. Build components are specific to the unique needs of this project.

4. Work Breakdown Structure

4.1 Project Initiation

4.1.1 Define Project Scope

4.1.2 Create Project Charter

4.1.3 Create Project Team

4.2 Requirements Analysis

4.2.1 Gather User Requirements

4.2.2 Analyze Functional Requirements

4.2.3 Analyze Non-Functional Requirements

4.2.4 Software Requirements Specification (SRS) Document.

4.2.5 Review and Approve SRS Document.

4.3 System Design

4.3.1 Design System Architecture (4+1 Architecture)

4.3.2 Design Database Schema

4.3.3 Design User Interfaces

4.3.4 Create Use Case Diagram.

4.3.5 Review and Approve Design Documents.

4.4 Implementation

4.4.1 Set Up Development Environment

4.4.1.1 Install Flask

4.4.1.2 Dashboard Interface (visualization of sales data).

4.4.1.3 User Interface (for admin and analysts).

4.4.1.4 Setup MySQL database

4.4.1.5 Setup version control

4.4.2 Implement Database

4.4.3 Develop Backend

4.4.3.1 User Management (role-based access control - RBAC).

4.4.3.2 Sales Data Management

4.4.3.3 Analytics Engine

4.4.4 Integrate Frontend and Backend

4.5 Testing

4.5.1 Develop Test Plans.

4.5.2 Perform Unit Testing

4.5.3 Perform Integration Testing

4.5.4 Perform System Testing

4.5.5 Perform User Acceptance Testing

4.5.6 Bug Fixing and Retesting.

4.6 Deployment

4.6.1 Prepare Deployment Environment

4.6.2 Deploy the Sales Analytics Dashboard Application.

4.6.3 Perform Final Checks

4.7 Project Closure

4.7.1 Prepare Final Project Documentation

4.7.2 Conduct Project Review

5. Gantt Chart

	AUGUST				SEPTEMBER				OCTOBER				NOVEMBER			
Weeks	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Problem finalization																
SRS & Documentation																
Architecture & Design Planning																
Backend Framework Setup																
Database Design & Setup(ETL)																
User Authentication & RBAC																
OLAP																
UI/Frontend Design																
Data Visualization Components																
Data Integration (Backend to Frontend)																
Integration Testing																
Unit & Load Testing																
Final Testing & Documentation																

6. Effort Estimation

1 year = 260 working days

1 month = $260/12 = 21.66$ working days

7 full working days for one person = $7/21.66 = 0.323$ person-months

6.1 Initiation: 0.5

6.2 Requirement Analysis: 1

6.3 System Design: 1.5

6.4 Implementation: 2

6.5 Testing: 1

6.6 Deployment: 0.5

6.7 Closure: 0.5

Total Effort: 7.0 person-month

Effort per person = Total Effort / Number of person

= $7.0 / 2 = 3.5$ person-month

Time taken = 3.5 months

7. COCOMO Model Estimation

$$\text{Effort (E)} = a \times (\text{KLOC})^b$$

Sales Analytics Dashboard project is **Semi-Detached**

$a=3.0$ and $b=1.122$

Let us assume LOC ~ 4000

$$\text{KLOC} = 4000/1000 = 4 \text{ KLOC}$$

$$E = 3.0 * (4)^{1.122}$$

$$= 14.211 \text{ person-month}$$