Practical No. 02

Problem Definition – Estimation of Project Metrics

(Based on Movie Recommendation System Project)

A. Aim of the Experiment

To estimate the cost, time, and effort required to develop a Movie Recommendation System using standard software estimation models like COCOMO and Halstead's Metrics.

B. Problem Statement

Design and develop a Movie Recommendation System that allows users to view, rate, and receive movie recommendations based on their interests and previous history using machine learning algorithms. This system must be user-friendly, scalable, and efficient in generating real-time recommendations.

C. Introduction

Before developing any software, it is important to analyze the project’s requirements and estimate the resources required. This includes calculating the time to develop the project, manpower, cost, and complexity. Proper estimation allows for better planning and project management.

In this experiment, we explore how estimation models like COCOMO and Halstead's Metrics can help predict the size, effort, and time required for the project. Estimation is especially useful when working with limited resources and tight deadlines.

D. Objectives

After completing this experiment, you will be able to:

Understand the need for project estimation in software development.

Apply COCOMO and Halstead’s Metrics to estimate project efforts.

Analyze the complexity and time required to complete the system.

Plan the phases of the software development lifecycle based on metrics.

E. Theory

1. Requirements Gathering

Functional Requirements:

User Registration/Login

View & Browse Movies

Rate Movies

Generate Movie Recommendations

Admin Panel for Movie Management

Non-functional Requirements:

Quick Response Time

User-friendly Interface

Security for User Data

Scalability for large number of users

2. Estimation Techniques

a) COCOMO (Constructive Cost Model)

The Basic COCOMO model estimates the effort required based on the size of the project in Kilo Lines of Code (KLOC).

Let’s assume:

Project size = 10,000 LOC = 10 KLOC

Development type = Organic (simple and small project)

Formula:

Effort (E) = 2.4 × (KLOC)^1.05

Time (T) = 2.5 × (Effort)^0.38

Calculation:

Effort = 2.4 × (10)^1.05 ≈ 27 person-months

Time = 2.5 × (27)^0.38 ≈ 9.5 months

Team required = Effort / Time ≈ 27 / 9.5 ≈ 3 developers

b) Halstead’s Metrics

This is used to estimate program complexity based on the number of operators and operands in the code.

Key Parameters:

n1 = number of unique operators

n2 = number of unique operands

N1 = total occurrences of operators

N2 = total occurrences of operands

Metrics:

Program Length (N) = N1 + N2

Program Vocabulary (n) = n1 + n2

Volume (V) = N × log₂(n)

Effort (E) = V / (2 × (n1/n2)) (simplified)

This helps measure the logical complexity of the code and effort needed for development.

3. Comparison Table

Estimation Model Pros Cons

COCOMO Simple and good for early planning Requires accurate LOC estimate

Halstead's Useful for complexity analysis Needs knowledge of program structure

F. Case Study – Movie Recommendation System

Parameter Estimated Value

Estimated LOC 10,000 lines

Estimated Effort 27 person-months

Estimated Duration ~9.5 months

Team Size 3-4 developers

Technologies Used Python, Flask/Django, Machine Learning libraries (scikit-learn, pandas), SQLite/MySQL

Tools Jupyter, VS Code, Git, Trello, UML tools

G. Simulation (if available)

Using a project planning tool like Gantt Chart or ProjectLibre, we can simulate the development cycle:

Week 1–2: Requirements & Design

Week 3–6: Backend & Database Integration

Week 7–10: Recommendation Engine Implementation

Week 11–13: Testing & Debugging

Week 14: Final Deployment

H. Conclusion

Project estimation is a critical step in software engineering. Using techniques like COCOMO and Halstead’s Metrics, we were able to:

Predict the time and cost for developing a Movie Recommendation System.

Understand the complexity of the project.

Allocate resources and create a development schedule.