Practical No.2

Pharmacy automation system- Project Estimation

Team Members: Hitesha shahane & Sharyu Charde

Aim of the Experiment

In this experiment, we will learn how to estimate the cost, effort, and duration for developing a Pharmacy Automation System using software estimation techniques like COCOMO and Halstead's Metrics, and determine the most suitable development strategy to fulfill the organization's business goals.

Introduction

Real estate has become one of the most dynamic and data-driven sectors. A Pharmacy Automation System helps users and stakeholders (buyers, sellers, agents) predict real estate prices based on several factors such as location, square footage, amenities, past prices, and market trends.

After collecting requirements for such a system (data ingestion, model training, prediction UI, etc.), different solution approaches are evaluated by business analysts in terms of cost, effort, and development time.

Estimation of Project Metrics

Objectives

- Classify the project under COCOMO and estimate the effort, cost, and time.
- Estimate the software complexity using Halstead's metrics.
- Understand how project estimation influences model and system development decisions.

Theory

Understanding **Pharmacy Automation System**

The system aims to estimate real estate prices using machine learning. The project typically includes:

- Data Collection and Preprocessing Module
- Model Training and Validation Module
- Web Interface for Users
- Database and API Backend
- Deployment & Monitoring Tools

Project Parameters

- Project Size: Based on lines of code for backend APIs, ML pipeline, and frontend.
- Cost: Includes data scientists, developers, DevOps, cloud costs.
- Duration: Time from initial development to stable deployment.
- Effort: Total person-months needed.

COCOMO Estimation for Pharmacy Automation System

Let us now try to classify this system using COCOMO models:

Classification: Semi-Detached Project

This project typically involves a mixed team of data scientists and developers. It is moderately complex due to:
- Model integration with UI
- Moderate team size
- Use of known tools (e.g., Scikit-learn, React)

Hence, we assume it to be a semi-detached project.

Basic COCOMO Estimation

For semi-detached:
a = 3.0, b = 1.12, c = 0.35

Suppose the project is estimated to have 8 KDSI (Kilo Delivered Source Instructions).

Effort (PM) = $3.0 \times (8)^{1.12} \approx 28.23$ PM
Development Time (Months) = $2.5 \times (28.23)^{0.35} \approx 7.8$ months
Team Size = $28.23 / 7.8 \approx 4$ people

Intermediate COCOMO (with EAF)

Assume EAF = 1.10 based on cost drivers.

Corrected Effort = $28.23 \times 1.10 = 31.05$ PM
Corrected Time = $2.5 \times (31.05)^{0.35} \approx 8.1$ months

Halstead's Complexity Metrics for ML Module

Sample Python Code Analysis:
$n1 = 25$ (unique operators), $n2 = 45$ (unique operands)
$N1 = 180$ (total operators), $N2 = 220$ (total operands)

Program Length = $N1 + N2 = 400$
Vocabulary = $n1 + n2 = 70$
Volume = $400 \times \log2(70) \approx 2452$ bits
Difficulty = $(25 \times 220) / (2 \times 45) \approx 61.11$

Effort = 61.11 × 2452 ≈ 149,890
Time = Effort / 18 ≈ 8327 seconds ≈ 2.31 hours

Advantages of This Estimation Approach

- Quantitative insight into development feasibility.
- Helps in resource allocation and planning.
- Identifies bottlenecks and prioritizes modules.
- Useful for budgeting and proposals.

Drawbacks and Limitations

- Less effective for ML projects due to experimentation loops.
- KDSI is hard to define in ML pipelines.
- Doesn't consider training/data-specific complexity.
- Originally designed for waterfall model, not agile.

Conclusion

In this experiment, we successfully applied COCOMO and Halstead's metrics to estimate the effort, time, and complexity of Pharmacy Automation System. We learned how software estimation models help plan real-world projects and how different modules within an ML system may vary in complexity and cost. Such early predictions can help teams make informed decisions, optimize architecture, and reduce project risks.