**Practical No.2**

**ESTIMATION OF PROJECT METRICS**

**Aim :**

To estimate the cost, effort, and duration of a software project based on the identified requirements, and to select a suitable solution approach that fulfills the organizational goals.

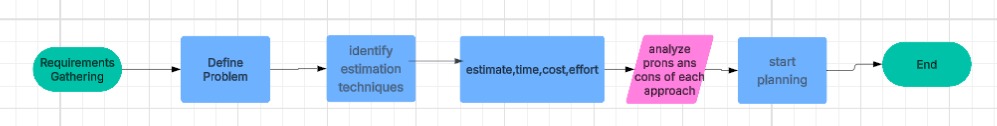
**Problem Definition:**

(Estimation of Project Metrics)

In this experiment, we will learn how to estimate cost, effort, and time for a software project. The goal is to identify the best solution strategy using project estimation techniques. This helps ensure that the selected approach is efficient and aligned with organizational needs.

**Introduction:**

After gathering all the requirements specific to a software project, we must think about various solution strategies. Expert business analysts evaluate these strategies by analyzing their benefits and drawbacks in terms of cost, time, and required resources to develop them.



**Theory:**

1. Objectives

After completing this experiment, you will be able to:

(a) Estimate cost, time, and effort of a project using standard techniques.

(b) Select the best solution approach for project development.

2. Project Estimation Techniques

You will study and apply the following techniques:

COCOMO (Constructive Cost Model):

Estimates project effort and cost using size (LOC - lines of code) as input.

Types: Basic, Intermediate, and Detailed COCOMO.

Types of Estimation Models:

* Algorithmic models (e.g., COCOMO)
* Expert judgment
* Analogy-based estimation
* Machine learning approaches (in advanced settings)

Halstead’s Complexity Metrics:

Measures complexity using program operators and operands to estimate effort, time, and bugs.

Metrics include:

* Program vocabulary
* Program length
* Volume, Difficulty, and Effort

Advantages:

* Helps in better planning and resource allocation
* Reduces risks
* Enables cost control

Drawbacks:

* Depends on accurate input data
* May not work well for new/innovative projects without historical data
* Manual estimation can be biased or inconsistent

Basic COCOMO Estimation:

Organic mode constants:

a = 2.4, b = 1.05, c = 0.38

Effort = a × (KLOC)^b

→ Effort = 2.4 × (10)^1.05 ≈ 27 PM (Person-Months)

Development Time = 2.5 × (Effort)^c

→ Tdev = 2.5 × (27)^0.38 ≈ 8.7 Months

➤ Intermediate COCOMO Estimation (using EAF):

Assuming the following cost driver weights:

Product Reliability: Nominal (1.0)

Complexity: Nominal (1.0)

Team experience: Very High (0.7)

Required Development Schedule: Tight (1.15)

Platform experience: High (0.85)

Use of Software Tools: High (0.85)

Real-time constraints: Low (0.9)

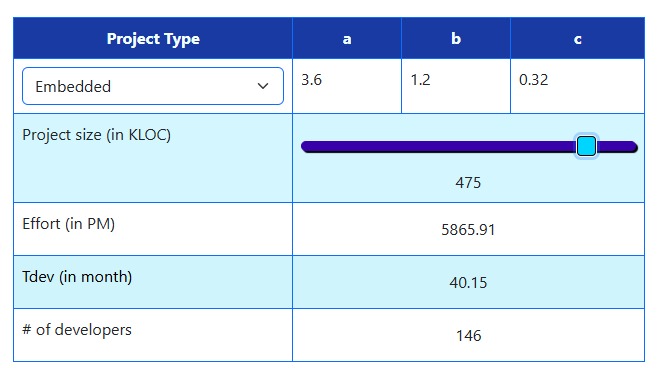
Effort Adjustment Factor (EAF) =

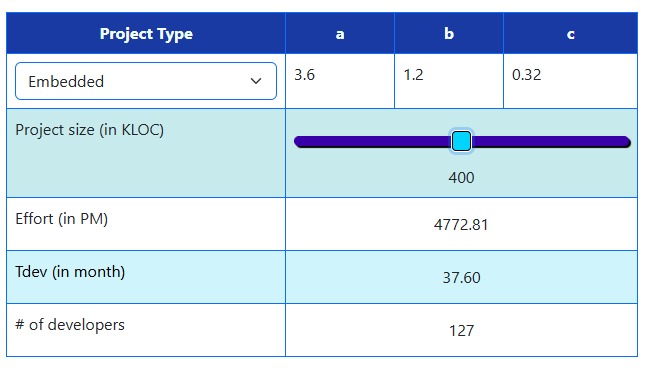
1.0 × 1.0 × 0.7 × 1.15 × 0.85 × 0.85 × 0.9 ≈ 0.53

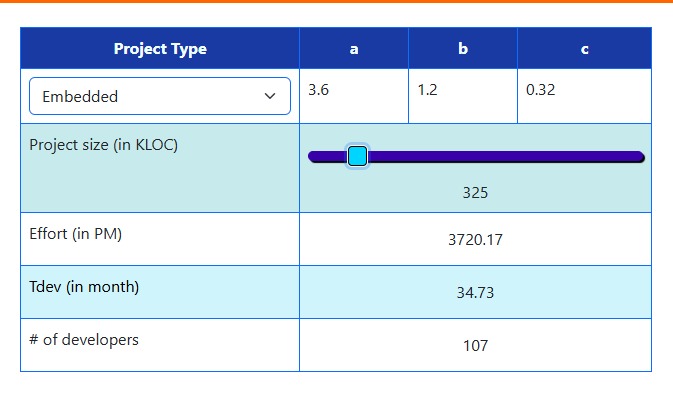
Revised Estimates:

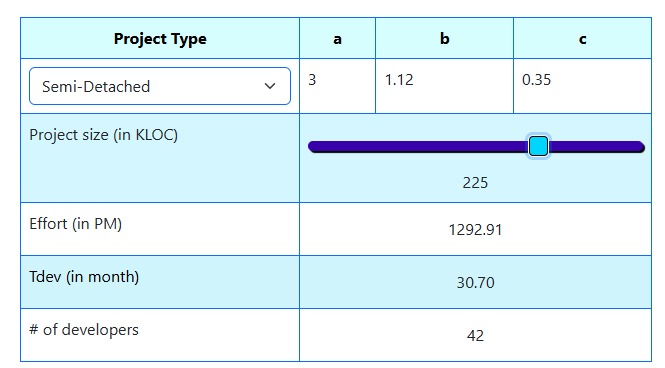
Effort|corrected = 27 × 0.53 = 14.3 PM

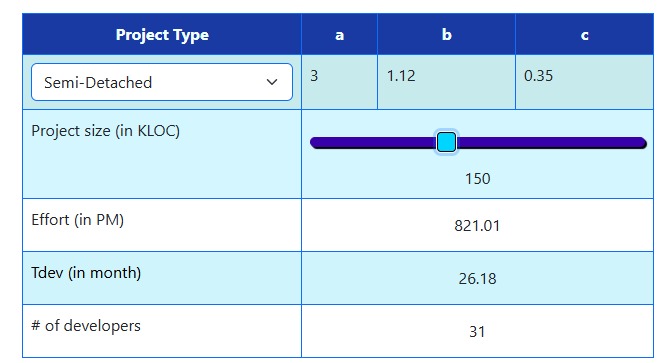
Tdev|corrected = 2.5 × (14.3)^0.38 ≈ 7 months

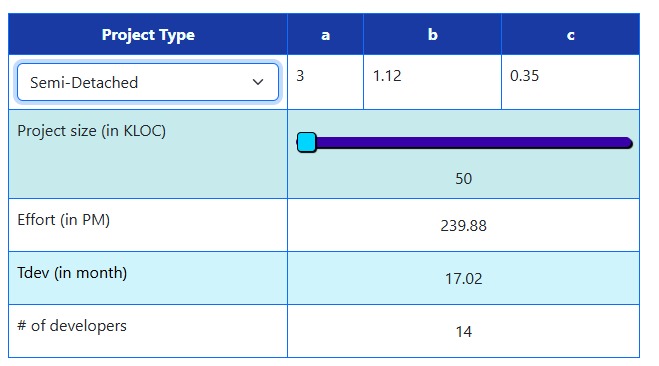


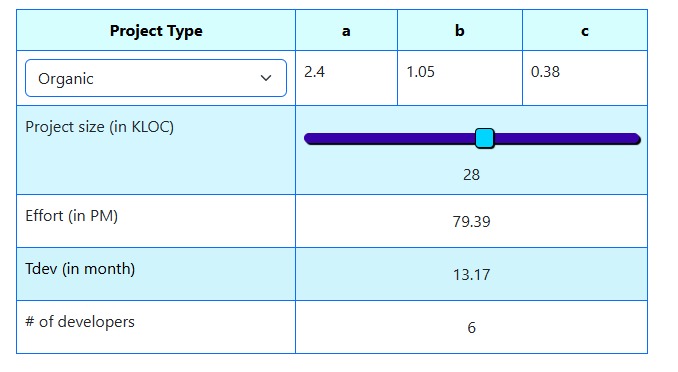


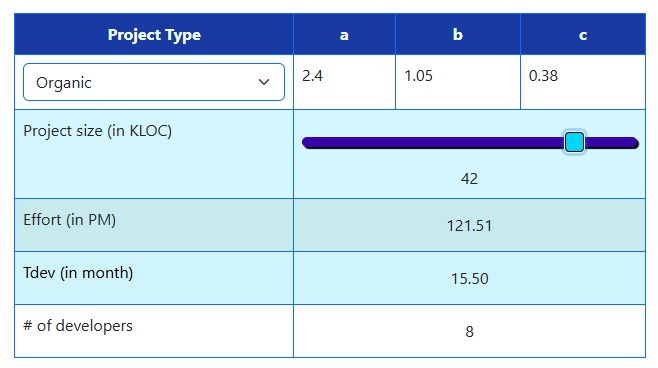


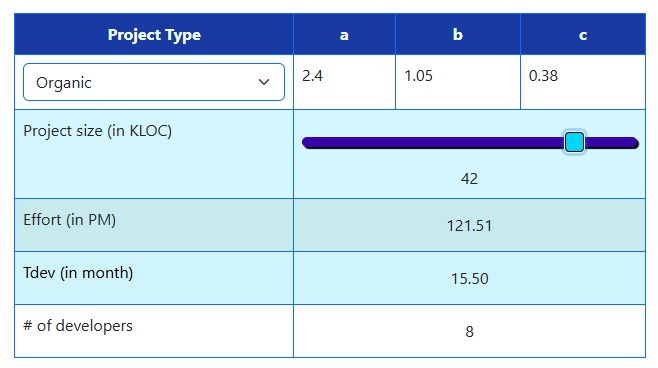












**Conclusion :**

This experiment helped us understand how to apply project estimation techniques like COCOMO and Halstead’s Metrics. We learned how to calculate cost, effort, and time based on software requirements and how to evaluate different solution strategies. Proper estimation is crucial to avoid project delays, budget overruns, and mismanagement.