**PRACTICAL NO.02 Estimation of Project Metrics**

Aim :

In this experiment, we will learn how to estimate cost, effort and duration for a software project, and then select one solution approach which will be found suitable to fulfill the organizational goal.

**Objectives**

After completing this experiment you will be able to:

Categorize projects using COCOMO, and estimate effort and development time required for a project

Estimate the program complexity and effort required to recreate it using Halstead's metrics

**Theory:**

**Project Estimation Techniques**

A software project is not just about writing a few hundred lines of source code to achieve a particular objective. The scope of a software project is comparatively quite large, and such a project could take several years to complete. However, the phrase "quite large" could only give some (possibly vague) qualitative information. As in any other science and engineering discipline, one would be interested to measure how complex a project is.

* **Project size:** What would be the size of the code written say, in number of lines, files, modules?
* **Cost:** How much would it cost to develop a software? A software may be just pieces of code, but one has to pay to the managers, developers, and other project personnel.
* **Duration:** How long would it be before the software is delivered to the clients?
* **Effort:** How much effort from the team members would be required to create the software?

COCOMO

COCOMO (Constructive Cost Model) was proposed by Boehm. According to him, there could be three categories of software projects: organic, semidetached, and embedded. The classification is done considering the characteristics of the software, the development team and environment. These product classes typically correspond to application, utility and system programs, respectively.

* **Organic:** A development project is said to be of organic type, if
  + The project deals with developing a well understood application
  + The development team is small
  + The team members have prior experience in working with similar types of projects
* **Semidetached:** A development project can be categorized as semidetached type, if
  + The team consists of some experienced as well as inexperienced staff
  + Team members may have some experience on the type of system to be developed
* **Embedded:** Embedded type of development project are those, which
  + Aims to develop a software strongly related to machine hardware
  + Team size is usually large

**Intermediate COCOMO Model**

The basic COCOMO model considers that effort and development time depends only on the size of the software. However, in real life there are many other project parameters that influence the development process.

**Complete COCOMO Model**

Both the basic and intermediate COCOMO models consider a software to be a single homogeneous entity -- an assumption, which is rarely true. In fact, many real life applications are made up of several smaller sub-systems.

**Advantages of COCOMO**

COCOMO is a simple model, and should help one to understand the concept of project metrics estimation.

**Drawbacks of COCOMO**

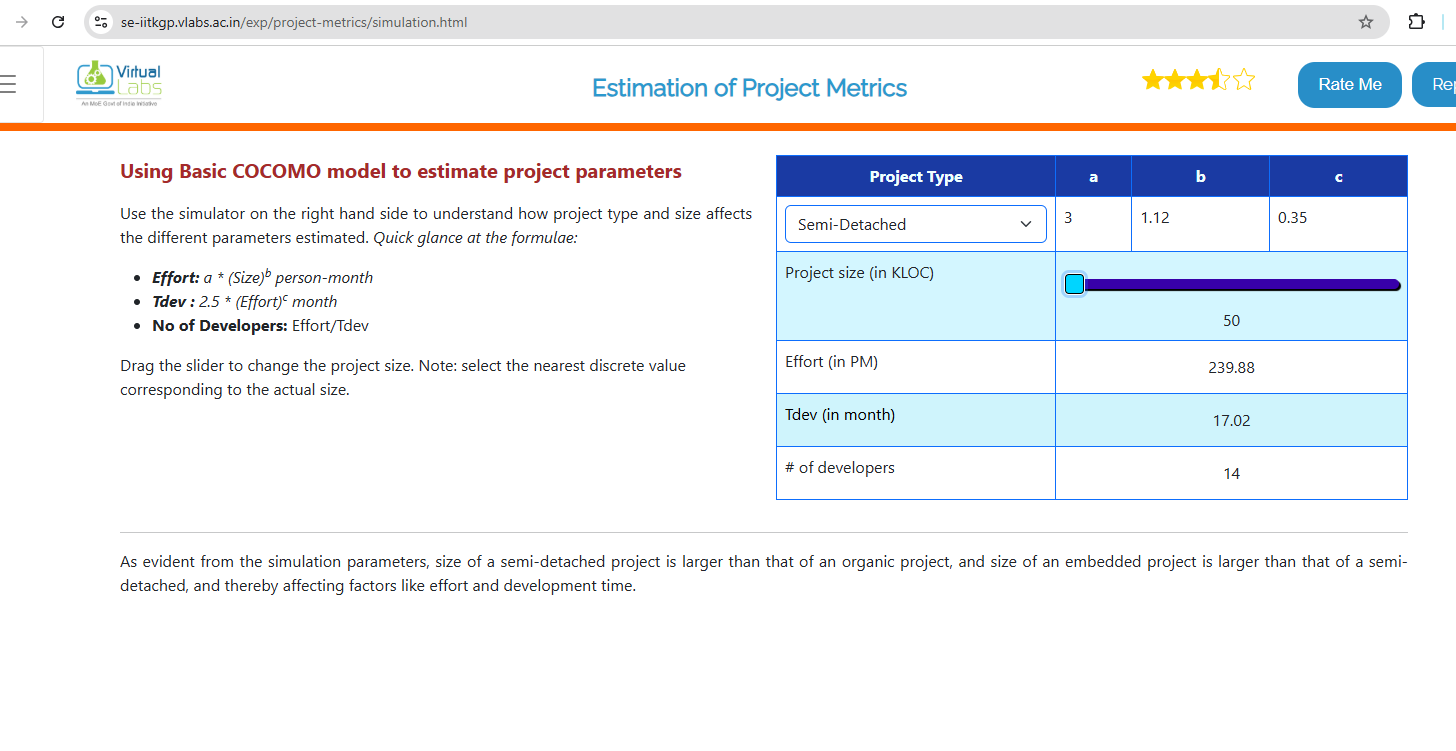
COCOMO uses KDSI, which is not a proper measure of a program's size. Indeed, estimating the size of a software is a difficult task, and any slight miscalculation could cause a large deviation in subsequent project estimates. Moreover, COCOMO was proposed in 1981 keeping the waterfall model of project life cycle in mind. It fails to address other popular approaches like prototype, incremental, spiral, agile models.

**Sample Calculations (can match with simulator):**

Effort=3.0×(50)1.12= 259 PM

Tdev=2.5×(259)0.35= 19.5 months

Tdev=2.5×(259)0.35= 19.5 months



**Calculations:**

Effort = 2.4 × (2)^1.05 ≈ 2.4 × 2.07 = **4.97 PM**

Tdev = 2.5 × (4.97)^0.38 ≈ 2.5 × 1.87 = **4.68 months**

Developers = 4.97 / 4.68 ≈ **1.06 ≈ 1 developer**

