Practical No.02

Aim: Estimation Of Project Metrics.

**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. One of the major activities of the project planning phase, therefore, is to estimate various project parameters in order to take proper decisions. Some important project parameters that are estimated include:

* **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?

In this experiment we will focus on two methods for estimating project metrics: COCOMO and Halstead's method.

**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. Data processing programs could be considered as application programs. Compilers, linkers, are examples of utility programs. Operating systems, real-time system programs are examples of system programs. One could easily apprehend that it would take much more time and effort to develop an OS than an attendance management system.

COCOMO classifies projects into **organic, semidetached, and embedded** types:

* **Organic:** Small team, well-understood application, experienced members.
* **Semidetached:** Mix of experienced and inexperienced staff, some familiarity with the system.
* **Embedded:** Large team, software closely tied to hardware.

Boehm proposed **three estimation stages**: **Basic COCOMO, Intermediate COCOMO, and Complete COCOMO** for project parameter estimation.

In your project **"Prediction of Personality Traits Based on Handwriting Analysis"**, you are using **Machine Learning (ML) technology**, which involves **data processing, model training, and integration with a user interface**. Based on the **COCOMO model classification**, your project is best categorized as a **Semi-Detached** project because:

* The project involves **moderate complexity** (ML-based system but not as complex as real-time embedded systems).
* It requires a **mix of experienced and inexperienced team members** (ML experts, software developers, UI designers, etc.).
* It has a balance between **application-level software and system-level processing** (feature extraction, model training, prediction, and UI integration).

Thus, the **COCOMO constants for your project** would be:

| **Software Category** | **a** | **b** | **c** |
| --- | --- | --- | --- |
| **Semi-Detached** | **3.0** | **1.12** | **0.35** |

These constants will be used in **effort estimation** (person-months) and **time estimation** for your project.