Chapter 3 exercise (3.2)

Student Name: Amishkumar Navadia

Student ID: 40298046

Course: Software Project Management (SOEN 6841)

Exercise URL: https://github.com/AmishNavadia/SOEN-6841-Amish-Navadia

Exercise 3.2:

Describe the COCOMO technique for deriving effort and cost estimates for software projects?

Solution :

The COCOMO (COnstructive COst MOdel) technique is a software cost estimation model used to predict the effort, time, and cost required to develop a software project. It uses mathematical equations to estimate these factors based on the size of the software and other relevant project characteristics. Here's a clear explanation:

Key Components of COCOMO:

1. **Effort**: The total amount of work required, typically measured in **person-months** (i.e., the amount of work one person can do in one month).
2. **Time**: The time required to complete the project, typically measured in **months**.
3. **Cost**: The total cost of the project, calculated by multiplying the effort by the cost per person-month.

Steps for Using COCOMO:

1. **Estimate Software Size:**
   - The size is usually measured in Kilo Delivered Source Instructions (KDSI) (thousands of lines of code) or sometimes in Function Points.

2. **Classify the Project Type**: COCOMO divides software projects into three categories based on their complexity:
   - Organic: Simple projects with experienced teams.
   - Semi-Detached: Moderately complex projects with mixed team experience.
   - Embedded: Large, complex projects with strict constraints (e.g., real-time systems).

3. **Estimate Effort Using the Formula:** The basic formula for estimating the effort required is:

$$\boldsymbol{Effort = a \cdot (Size)b}$$

Where:

- **Size** is the software size (e.g., in KDSI).
- **a** and **b** are constants that vary depending on the project type (organic, semi-detached, or embedded).

4. **Estimate Time Using the Formula**: Once the effort is calculated, COCOMO estimates the project duration using the formula:

$$\boldsymbol{Time = c \cdot (Effort)\, d}$$

Where:

- **c** and **d** are constants based on the project type.

5. **Calculate the Cost**: The cost is estimated by multiplying the **effort** by the cost per person-month:

$$\boldsymbol{Cost = Effort \cdot Cost\, per\, Person - Month}$$

Types of COCOMO Models:

1. **Basic COCOMO**: This model provides a rough estimate using just the size of the software and its category (organic, semi-detached, or embedded).
2. **Intermediate COCOMO**: This model refines the basic estimate by including **cost drivers**—factors such as product complexity, team experience, and development tools that affect the effort.
3. **Detailed COCOMO**: This model provides the most precise estimates by breaking the project into smaller components, considering interactions between different parts of the project, and applying additional cost drivers.

Example Calculation:

Imagine We are developing a **semi-detached** software project with **20 KDSI**:

- Constants for **semi-detached** project: **a = 3.0**, **b = 1.12**
- **Effort** estimation:

$$\boldsymbol{Effort = 3.0 \cdot (20)1.12 = 3.0 \cdot 26.8 \approx 80.4 person - months}$$

- **Time** estimation using **c = 2.5** and **d = 0.38**

$$\boldsymbol{Time = 2.5 \cdot (80.4)0.38 \approx 2.5 \cdot 10.8 \approx 27 months}$$

- **Cost** estimation (assuming a cost of $5,000 per person-month):

$$\boldsymbol{Cost = 80.4 \cdot 5{,}000 = 402{,}000 USD}$$

Advantages of COCOMO:

- **Structured Approach**: It provides a well-defined methodology for estimating effort, time, and cost.
- **Scalability**: It works for a variety of project sizes and complexities.
- **Realistic Estimates**: Based on historical data, making it relatively accurate for projects of similar nature.

Limitations of COCOMO:

- **Size Estimation:** The accuracy depends on how well the software size (KDSI) can be predicted early in the project.
- **Traditional Models:** COCOMO assumes a waterfall-style development process, which may not suit Agile or modern development methodologies.
- **Adjustments Required:** The constants and cost drivers may need to be adjusted to reflect changes in technology or modern practices.