Department of Computer Engineering

Academic Term: First Term 2023-24

Class: T.E /Computer Sem – V / Software Engineering

Practical No:	**5**
Title:	**Estimating project cost using COCOMO Model**
Date of Performance:	31-08-2022
Roll No:	9595
Team Members:	Atharva Dalvi

Rubrics for Evaluation:

Sr. No	Performance Indicator	Excellent	Good	Below Average	Total Score
1	On time Completion & Submission (01)	01 (On Time)	NA	00 (Not on Time)	
2	Theory Understanding(02)	02(Correct)	NA	01 (Tried)	
3	Content Quality (03)	03(All used)	02 (Partial)	01 (rarely followed)	
4	Post Lab Questions (04)	04(done well)	3 (Partially Correct)	2(submitted)	

Signature of the Teacher:

Department of Computer Engineering

Academic Term: First Term 2022-23

Class: T.E /Computer Sem – V / Software Engineering

Signature of the Teacher:

Lab Experiment 05

Experiment Name:
Estimating Project Cost Using COCOMO Model in Software Engineering

Objective:
The objective of this lab experiment is to introduce students to the COCOMO (Constructive Cost Model) estimation technique for estimating software project cost and effort. Students will gain practical experience in using the COCOMO model to estimate the development effort, duration, and resources required for a sample software project.

Introduction: COCOMO is a widely used algorithmic cost estimation model in software engineering. It helps in quantifying the effort and resources needed for software development based on project size, complexity, and other factors.

Lab Experiment Overview:
1. Introduction to COCOMO Model: The lab session begins with an introduction to the COCOMO model, explaining the different versions (Basic, Intermediate, and Advanced) and their application in software cost estimation.
2. Defining the Sample Project: Students are provided with a sample software project along with its functional and non-functional requirements, complexity, and size metrics.
3. COCOMO Parameters: Students learn about the COCOMO model parameters, such as Effort Adjustment Factor (EAF), Scale Factors, and Cost Drivers, and how they influence the project's effort estimation.
4. Effort and Duration Estimation: Using the COCOMO model formula, students estimate the effort and duration required to complete the sample project based on the provided size and complexity metrics.
5. Resource Allocation: Students estimate the number of required resources, such as developers, testers, and project managers, based on the calculated effort and project duration.
6. Sensitivity Analysis: Students perform sensitivity analysis by varying the COCOMO parameters to observe their impact on the project cost estimation.
7. Conclusion and Reflection: Students discuss the significance of COCOMO in software project estimation and reflect on their experience in estimating project cost using the COCOMO model.

Learning Outcomes: By the end of this lab experiment, students are expected to:
Understand the COCOMO model and its application in software cost estimation.
Gain practical experience in using the COCOMO model to estimate effort, duration, and resources for a software project.
Learn to consider various project factors and adjust COCOMO parameters for accurate cost estimation.
Develop estimation skills for resource allocation and project planning.
Appreciate the importance of data accuracy and project size metrics in project cost estimation.

Pre-Lab Preparations:
Before the lab session, students should familiarize themselves with the COCOMO model, its parameters, and the cost estimation formula. They should also review the factors that influence the project's size and complexity.
Materials and Resources:
Project brief and details for the sample software project
COCOMO model guidelines and cost estimation formula
Calculators or spreadsheet software for performing calculations
Conclusion: The lab experiment on estimating project cost using the COCOMO model provides

students with practical insights into software cost estimation techniques. By applying the COCOMO model to a sample software project, students gain hands-on experience in assessing effort, duration, and resource requirements. The sensitivity analysis allows them to understand the impact of various factors on cost estimation. The lab experiment encourages students to use COCOMO in real-world scenarios, promoting informed decision-making in software project planning and resource allocation. Accurate cost estimation using COCOMO enhances project management and contributes to the successful execution of software engineering projects.

The table for constants for **Basic COCOMO mode**l is as follows:

Software Product	A_b	B_b	C_b	D_b
Organic	2.4	1.05	2.5	0.38
Semi- Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

$E = a(KLOC)^b$

Function Point of project : 78.1

for average LOC/FP for Java : 63

LOC = 63 x 78.1 = 4920

KLOC = 4920/1000 = 4.920

Software Project is taken to be **Organic** type
taking values of a = 2.4 and b = 1.05
substituting all values in the equation
$E = 2.4(4.920)^{1.05}$
$= 2.4 \times 5.327$
$= 12.784$

Effort for basic COCOMO model = 12.784

$D(Time) = c(E)^d$
$= 2.5 \times (12.784)^{0.38}$
$= 2.5 \times 2.63$
D = 6.575

Average Staff Size : E/T = 12.784/6.575 = 2

Cost Drivers	RATINGS					
	Very low	Low	Nominal	High	Very high	Extra high
Personnel Attributes						
ACAP	1.46	1.19	1.00	0.86	0.71	..
AEXP	1.29	1.13	1.00	0.91	0.82	..
PCAP	1.42	1.17	1.00	0.86	0.70	..
VEXP	1.21	1.10	1.00	0.90	..	..
LEXP	1.14	1.07	1.00	0.95	..	..
Project Attributes						
MODP	1.24	1.10	1.00	0.91	0.82	..
TOOL	1.24	1.10	1.00	0.91	0.83	..
SCED	1.23	1.08	1.00	1.04	1.10	..

Figure 6. Ratings for cost drivers under personnel attributes and project attributes

Cost Drivers	RATINGS					
	Very low	Low	Nominal	High	Very High	Extra High
Product Attributes						
RELY	0.75	0.88	1.00	1.15	1.40	..
DATA	..	0.94	1.00	1.08	1.16	..
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
Computer Attributes						
TIME	..	..	1.00	1.11	1.30	1.66
STOR	..	..	1.00	1.06	1.21	1.56
VIRT	..	0.87	1.00	1.15	1.30	..
TURN	..	0.87	1.00	1.07	1.15	..

For intermediate COCOMO model

E = a(KLOC)^b * (EAF)

COST DRIVERS:

Personal attributes:

Analyst Capability (ACAP) , Low = 1.19
Application Experience (AEXP), Nominal = 1.00
Software Engineering capability (PCAP), Low = 1.17
Experience using VM (VEXP), Nominal = 1.00
Programming language experience (LEXP), High = 0.95

Project attributes:

Applications of Software Eng Methods (MODP), Nominal = 1.00
Applications of Software Tools (TOOL), High = 0.91
Required Development Schedule (SCED), Nominal = 1.00

Product attributes:

Required Software reliability (RELY), HIgh= 1.15
Database size (DATA), High = 1.08
Product Complexity (CPLX), Very High = 1.30

Computer attributes:

Execution time constraints (TIME), High = 1.11
Main storage constraints (STOR), High = 1.06
Virtual Machine Volatility (VIRT), Nominal = 1.00
Required turnaround time (TURN), High = 1.07

Calculating EAF:

EAF = ACAP x AEXP x PCAP x VEXP x LEXP x MODP x TOOL x SCED x RELY x DATA x CPLX x TIME x STOR x VIRT x TURN)

= 1.19 x 1.00 x 1.17 x 1.00 x 0.95 x 1.00 x 0.91 x 1.00 x 1.15 x 1.08 x 1.30 x 1.11 x 1.06 x 1.00 x 1.07

EAF =2.446

Calculating E:

substituting values in the equation:

E = 12.784 x 2.446
E = 31.269

Effort for intermediate COCOMO model = 31.269

Time Required:
$D=c(E)^d$
$D= 2.5(31.269)^{0.38}$
D= 9.24

Average Staff Size : E/T = 31.269/9.24 = 3

<u>For Detailed COCOMO model:</u>

$E_p = \mu_p * E$

$D_P = \tau_P *$

D

It is a **Organic small** model

For Plan and Requirements:
$E_p = 0.06 \times 31.269 =$ **1.876**
$D_P = 0.10 \times 9.24 =$ **0.924**

For System Design:
$E_p = 0.16 \times 31.269 =$ **5.003**
$D_P = 0.19 \times 9.24 =$ **1.755**

For Detailed Design:
$E_p = 0.26 \times 31.269 =$ **8.129**
$D_P = 0.24 \times 9.24 =$ **2.217**

For Module Code and Test:
$E_p = 0.42 \times 31.269 =$ **13.132**
$D_P = 0.39 \times 9.24 =$ **3.603**

For Integration and Test:
$E_p = 0.16 \times 31.269 =$ **5.003**
$D_P = 0.18 \times 9.24 =$ **1.663**

POSTLABS:

a) Analyse the COCOMO model and its different modes (Organic, Semidetached, Embedded) to determine the most suitable mode for a specific project type.

The COCOMO (COnstructive COst MOdel) is a well-established software cost estimation model developed by Barry Boehm. It helps project managers and software developers estimate the effort, time, and cost required to complete a software project. COCOMO has three different modes: Organic, Semidetached, and Embedded, which are designed to be used for different types of software projects. Let's analyze each mode and determine the most suitable one for a specific project type:

1. *Organic Mode*:
   - Characteristics:
     - Small to medium-sized projects.
     - A well-understood problem domain.
     - A small team of experienced developers.
     - Low or no strict requirements for performance and reliability.
   - Equation: E = a * (KLOC) ^ b
   - Parameters:
     - a = 2.4
     - b = 1.05

   Suitability:
   The organic mode is suitable for projects where simplicity and familiarity are dominant. It's ideal for projects like internal tools, small web applications, or straightforward mobile apps with a well-understood problem and a small, skilled team. It's not appropriate for complex, mission-critical systems.

2. *Semidetached Mode*:
   - Characteristics:
     - Medium-sized projects with some elements of unfamiliarity.
     - A mix of experienced and inexperienced team members.
     - Moderate requirements for performance and reliability.
   - Equation: E = a * (KLOC) ^ b
   - Parameters:
     - a = 3.0
     - b = 1.12

   Suitability:
   The semidetached mode suits projects that have a moderate level of complexity and where some aspects are not entirely familiar. Projects that need moderate performance and reliability, such as business applications or medium-sized software systems, are often suitable for this mode.

3. *Embedded Mode*:
   - Characteristics:
     - Large, complex projects with a high level of unfamiliarity.
     - A large team with varying levels of expertise.
     - High requirements for performance and reliability.
   - Equation: E = a * (KLOC) ^ b
   - Parameters:
     - a = 3.6

- b = 1.20

Suitability:
The embedded mode is appropriate for large and complex software projects that involve a significant level of uncertainty. It's suitable for systems like embedded software in aerospace, defense, or medical devices, where performance, reliability, and complexity are critical.

To determine the most suitable mode for a specific project type, consider the project's size, complexity, team expertise, and requirements for performance and reliability. If your project aligns more with the characteristics of one mode over the others, use the associated COCOMO mode to estimate the effort and cost. Keep in mind that these are general guidelines, and actual project conditions may vary, so it's essential to tailor the estimation based on the specific project's nuances.

b) Apply the COCOMO model to estimate the project cost and effort required for a given software development project.

To apply the COCOMO model to estimate the project cost and effort for a specific software development project, you'll need to gather relevant information about the project and then use the appropriate COCOMO mode (Organic, Semidetached, or Embedded) based on the project's characteristics. Here's a step-by-step guide:

Step 1: Define Project Characteristics

Before you can apply COCOMO, you need to understand the key characteristics of your project. Consider the following factors:

1. *Size of the project*: Usually measured in KLOC (Kilo Lines of Code) or other appropriate size metrics.

2. *Complexity*: Determine whether your project is small, medium, or large, and how familiar or unfamiliar the problem domain is.

3. *Team Expertise*: Assess the experience level of your development team. Are they mostly experienced or a mix of experienced and inexperienced developers?

4. *Requirements*: Identify the performance and reliability requirements of the project. Are they low, moderate, or high?

Step 2: Select the COCOMO Mode

Based on the characteristics of your project, select the appropriate COCOMO mode (Organic, Semidetached, or Embedded). Refer to the descriptions in the previous answer to make this determination.

Step 3: Calculate Effort and Cost

Once you've selected the COCOMO mode, use the relevant COCOMO equation for effort estimation. The general equation for effort estimation in COCOMO is:

Effort (E) = a * (Size) ^ b

Use the parameters (a and b) associated with the selected COCOMO mode. For example:

- If you're using Organic mode, a = 2.4 and b = 1.05.
- If you're using Semidetached mode, a = 3.0 and b = 1.12.
- If you're using Embedded mode, a = 3.6 and b = 1.20.

Calculate the effort based on the project size (in KLOC) and the parameters.

Step 4: Estimate Cost

To estimate cost, you'll need to take the estimated effort and factor in your organization's cost structure, which includes labor rates, overhead, and other project-specific costs. The cost estimation formula is usually organization-specific and can vary widely.

The general formula for cost is:

Cost = Effort * (Cost per Person-Month)

You'll need to determine the "Cost per Person-Month" based on your organization's cost structure and labor rates.

Step 5: Review and Adjust

Review your estimates, and if necessary, adjust them for specific project factors that COCOMO might not account for. These could include market conditions, tooling costs, or other project-specific variables.

c) Evaluate the factors influencing COCOMO estimates, such as project size,personnel capabilities, and development tools, and their implications on project planning and scheduling.

COCOMO (COnstructive COst MOdel) estimates are influenced by various factors that can significantly impact project planning and scheduling. These factors include project size, personnel capabilities, and development tools. Let's evaluate these factors and their implications on project planning and scheduling:

1. *Project Size*:

   - *Influence on Estimates*: The size of the software project, often measured in lines of code (LOC) or function points, is a primary driver of COCOMO estimates. Larger projects typically require more effort and resources.

   - *Implications on Project Planning and Scheduling*:
     - Larger projects tend to have longer development times, so project managers should allocate more time in the schedule.
     - Larger projects may also require more personnel, which can affect resource planning and team size.

- Extensive documentation and testing efforts may be needed for larger projects, affecting project phases and milestones.

2. *Personnel Capabilities*:

   - *Influence on Estimates*: The expertise and capabilities of the development team play a crucial role in estimating project effort and cost. Experienced teams tend to be more efficient.

   - *Implications on Project Planning and Scheduling*:
     - Highly skilled teams may complete tasks more quickly, allowing for faster project timelines.
     - Inexperienced teams may require additional time for training and guidance, potentially extending the schedule.
     - Consider team capabilities when assigning responsibilities and setting realistic milestones.

3. *Development Tools*:

   - *Influence on Estimates*: The tools, languages, and development environments used in a project can affect productivity and quality. Effective tools can reduce manual work and improve code quality.

   - *Implications on Project Planning and Scheduling*:
     - Efficient development tools can lead to faster development, allowing for shorter project schedules.
     - Compatibility issues or a learning curve associated with new tools may require additional time for training or debugging, affecting the schedule.
     - Regularly assess and update the toolset to ensure optimal productivity.

4. *Project Complexity*:

   - *Influence on Estimates*: The complexity of the software and the problem domain can significantly impact the estimates. Complex projects tend to require more effort.

   - *Implications on Project Planning and Scheduling*:
     - Complex projects may need a more detailed and extended analysis and design phase.
     - Extensive testing and debugging efforts are often necessary for complex projects, which can affect project timelines.
     - Project managers should allocate additional time for addressing complexity-related issues.

5. *Requirements and Changes*:

   - *Influence on Estimates*: Changing or evolving project requirements can impact estimates, especially if changes occur late in the project.

   - *Implications on Project Planning and Scheduling*:
     - Project managers should have a robust change management process to evaluate the impact of requirements changes on the project schedule.
     - Frequent requirement changes can lead to project delays if not managed effectively.

6. *Risk Management*:

   - *Influence on Estimates*: Effective risk management can reduce the impact of unforeseen issues on estimates.

- *Implications on Project Planning and Scheduling*:
  - Allocate time and resources for risk analysis and mitigation activities.
  - Have contingency plans in place to address potential risks that may affect the schedule.

In summary, these factors influence COCOMO estimates and, consequently, project planning and scheduling. Project managers must carefully consider these factors when creating project plans, allocating resources, and setting realistic schedules. Flexibility in project planning is essential, as adjustments may be required throughout the project's lifecycle to account for changing conditions and new information.