Effort and Cost Estimation using COCOMO II model For Exam Pilot

Prepared by

Mir Mohammod Tahasin | MUH2025007M

Irfanul Haque Nabil | MUH1925021M

IMTIAZ CHOWDHURY | MUH2025027M

Institute of Information Technology

Noakhali Science and Technology University

Group Project Report for

SE 3204 - Software Metrics

Submitted to

Md Hasan Imam

Lecturer

Institute of Information Technology

Noakhali Science and Technology University

Date of Submission: 26 December 2023

Table Of Contents

Effort and Cost Estimation using COCOMO II model

The Constructive Cost Model (COCOMO) is an algorithmic software cost estimation model developed by Barry Boehm. The model uses a basic regression formula, with parameters that are derived from historical project data and current project characteristics. COCOMO was first published in 1981 Barry W.

References to this model typically call it COCOMO 81. In 1997 COCOMO II was developed and finally published in 2000 in the book Software Cost Estimation with COCOMO II. COCOMO II is the successor of COCOMO 81 and is better suited for estimating modern software development projects. It provides more support for modern software development processes and an updated project database.

COCOMO II is tuned to modern software life cycles. The original COCOMO model has been very successful, but it doesn't apply to newer software development practices as well as it does to traditional practices. COCOMO II targets modern software projects, and will continue to evolve over the next few years.

COCOMO II has three different models:

1. The Application Composition Model

Suitable for projects built with modern GUI-builder tools. Based on new Object Points.

2. The Early Design Model

You can use this model to get rough estimates of a project's cost and duration before you've determined it's entire architecture. It uses a small set of new Cost Drivers, and new estimating equations. Based on Unadjusted Function Points or KSLOC.

3. The Post-Architecture Model

This is the most detailed COCOMO II model. You'll use it after you've developed your project's overall architecture. It has new cost drivers, new line counting rules, and new equations.

Stage I: Application composition estimation

Effort estimation:

Basically we can estimate the effort with application composition estimation for the application which includes GUI, Database management, Central package for domains.
Our Massier application supports GUI and Database facilities, that’s why we can estimate the effort with Application composition estimation efforts calculation strategy. At very early stages for our project we estimated our Application composition effort with the below approach.

Stage 1: Access Object Points
Here we have to count the number of estimated screens, reports and 3GL according to our project. In our project, we have 34 screens, 4 reports included screens.

Stage 2: Classify complexity levels for each object
Now, let’s define our object complexity level according to the below table.

No. of views contain	Sources of data tables		
	Total < 4 (< 2 servers < 3 clients)	Total < 8 (2 - 3 servers 3-5 clients)	Total 8 + (> 3 servers > 5 clients)
< 3	Simple	Simple	Medium
3 - 7	Simple	Medium	Difficult
> 8	Medium	Difficult	Difficult

For Screens

According to the information, for our project the complexity level would be like:

For Screens:

Number of views = 4

Number of data tables = 11

Number of Servers = 7

Number of Client = 5

So, according to the above table (For screens), **Complexity level= Difficult**

No. of section contain	Sources of data tables		
	Total < 4 (< 2 servers < 3 clients)	Total < 8 (2 - 3 servers 3-5 clients)	Total 8 + (> 3 servers > 5 clients)
0 - 1	Simple	Simple	Medium
2 - 3	Simple	Medium	Difficult
4 +	Medium	Difficult	Difficult

For Reports

For Reports:

Number of sections = 20

Number of data tables = 6

Number of servers = 3

Number of clients = 3

Now, using above table information we can find the

complexity for each report = Difficult

Step 3: Assign complexity weights to each object

For our project, we found two types of objects (Screens, reports). At this stage, we will estimate the complexity weights according to their level as we found at step 2.

Object Type	Complexity Weight		
	Simple	Medium	Difficult
Screen	1	2	3
Report	2	5	8
3GL Components	-	-	10

Complexity Weight

For Screen:

The complexity weights to each object(Dificult) = 3

For Reports:

The complexity weights to each object(Difficult) = 8

Step 4: Determine Object Points

The number of Object Points =

$$\sum(Number\ of\ Object\ instance)\ *\ \ (Complexity\ weights\ for\ that\ object)$$

= (34*4)+(3*8)

= 160

Step 5: Compute New Object Point(NOP)

For our project, we have 15% code that we get reused throughout the whole project.

So, now as we know computing

$NOP = \{object\ points\ * (100 - \%reuse)\}/100$

= {160 * (100 - 15)}/100

= 136

Step 6: Calculate Productivity rate(PROD)

Productivity rate is calculated on the basis of information given about developer's experience and capabilities.

Developers experience & capability	Productivity (PROD)
Very Low	4
Low	7
Nominal	13
High	25
High	50

Productivity Rate

For developing our project our developer has Nominal experience and capability. So, according to the above table we can measure our

Productivity Rate(PROD) is 13.

Step 7: Compute the estimated Effort

For develop our project the effort we have to provide with,

$Effort = NOP/PROD$

= 136 / 13

= **10.46 person-month**

Cost estimation:

Step 1: Determine the average salary

The average salary we calculated for our project was $1000 a month.

Step 2: Calculate the PM(Hours)
Effort in hours PM(Hours) = 10.46 person - months * 40 hours/month = 418.4 hours

Step 3: Estimate the Cost
Cost = PM(Hours) * Average Salary
= 418.4 * 1000
= $4,18,400 $

Early Stage & Post Architecture Stage:

Size of our project (KLOC):

Language	Files	Code	Comment	Blank	Total	Measurement
PHP	95	8592	530	1985	11707	Automated Tool
CSS	61	7133	325	1974	9432	Automated Tool
Javascript	30	1407	275	384	2030	Automated Tool
HTML	11	903	34	190	1127	Automated Tool

Totall:	197	18035	1164	4497	23696	

Here we can see that there are 23,696 line of code in our project. So we can measure 24 kloc in our project.

Scale Factor For Our Project:

Precedentness :

It reflects the experience on similar projects previously. This is applicable to individuals as well as organisations both in terms of expertise and experience. High value would imply that organization is quite familiar with the application formula and very low value means no previous experience or expertise.

Rating	Very Low	Low	Average	High	Very High	Extra High
Scale Factor	6.20	4.96	3.72	2.48	1.25	0.00

Our System Scale factor for Precedentness is : 3.72 (Average)

Measurement Preocedure: **Expert Interview**

Development Flexibility :

It reflects degree of flexibility in development process. Low value would imply well defined process being used. Very high value would imply that the client offers very general idea of the product or project.

Rating	Very Low	Low	Average	High	Very High	Extra High
Scale Factor	5.07	4.05	3.04	2.03	1.02	0.00

Our System Scale factor for **Development Flexibility** is : 4.05(Low)

Measurement Preocedure: **Review Of Requirements and contracts**

Architecture Risk and Resolution :

It represents degree of risk analysis being carried out during course of project. Low value would indicate little analysis and very high value would represent complete and thorough risk analysis.

Rating	Very Low	Low	Average	High	Very High	Extra High
Scale Factor	7.07	5.65	4.25	2.83	1.41	0.00

Our System Scale factor for**Architecture Risk and Resolution** is : 4.25(Average)

Measurement Preocedure: **Architecture Reviews and Review Of Prototype**

Team Cohesion :

Reflects the team management skills of the employees developing the project. Very low value would imply very low interaction and hardly any relationship among the members however high value would imply great relationship and good team work.

Rating	Very Low	Low	Average	High	Very High	Extra High
Scale Factor	5.48	4.38	3.29	2.19	1.10	0.00

Our System Scale factor for **Team Cohesion** is : 1.10(Very High)

Measurement Preocedure: **Observing team interactions, problem-solving approaches, and conflict resolution styles.**

Process maturity :

Reflects process maturity of organization. Very low value would imply organization has no level at all and high value would imply that organization is rated as highest level of the SEI-CMM.

Rating	Very Low	Low	Average	High	Very High	Extra High
Scale Factor	7.80	6.24	4.68	3.12	1.56	0.00

Our System Scale factor for **Team Cohesion** is : 3.12(High)

Measurement Preocedure: **Audit & Reviews**

Cost Drivers :

Early Design Cost Driver	Post Architecture Cost Drivers
PERS	ACAP, PCAP, PCON
RCPX	RELY, DATA, CPLX, DOCU
RUSE	RUSE
PDIF	TIME, STOR, PVOL
PREX	APEX, PLEX, LTEX
FCIL	TOOL, SITE
SCED	SCED

Product Factors

1.Required Software Reliability (RELY)

This is the measure of the extent to which the software must perform its intended function over a period of time. If the effect of a software failure is only slight inconvenience then RELY is very low. If a failure would risk human life then RELY is very high.

Rating	Very Low	Low	Nominal	High	Very High
Effort Multipliers	0.82	0.92	1.00	1.10	1.26

Value for RELY cost driver is : 0.82 (Very Low)

Because a failure in the functionality wouldn't risk human life .

Measurement Procedure : By reviewing the software functional requirements

2.Data Base Size (DATA) :

This cost driver attempts to capture the effect large test data requirements have on product development. The rating is determined by calculating D/P, the ratio of bytes in the testing database to SLOC in the program. The reason the size of the database is important to consider is because of the effort required to generate the test data that will be used to exercise the program.

Rating	Very Low	Low	Nominal	High	Very High
Effort Multipliers	n/a	0.90	1.00	1.14	1.28

Value for DATAcost driver is : 1 (Nominal)

Because the ratio of database size in bytes/SLOC is within 100 .

Measurement Procedure : By calculating the database size and SLOC

3.Product Complexity (CPLX):

Complexity is divided into five areas: control operations, computational operations, device-dependent operations, data management operations, and user interface management operations.The complexity rating is the subjective weighted

Rating	Very Low	Low	Nominal	High	Very High
Effort Multipliers	0.73	0.87	1.00	1.17	1.34

Value for CPLX cost driver is : 1 (Nominal)

Because there aren't complex computational operations or data management operations in our project .

Measurement Procedure : By Reviewing the overall complexity of the system

4.Developed for Reusability (RUSE) :

This cost driver accounts for the additional effort needed to construct components intended for reuse on current or future projects. This effort is consumed with creating more generic design of software, more elaborate documentation, and more extensive testing to ensure components are ready for use in other applications. “Across project” could apply to reuse across the modules in a single financial applications project. “Across program” could apply to reuse across multiple financial applications projects for a single organization. “Across product line” could apply if the reuse is extended across multiple organizations. “Across multiple product lines” could apply to reuse across financial, sales, and marketing product lines

Rating	Very Low	Low	Nominal	High	Very High
Effort Multipliers	n/a	0.95	1.00	1.07	1.15

Value for RUSE cost driver is : 1 (Nominal)

Because it could apply to reuse across the modules in a single financial applications project (Across project)

Measurement Procedure : By Reviewing the Reusability of the System

5.Documentation Match to Life-Cycle Needs (DOCU) :

Several software cost models have a cost driver for the level of required documentation. In COCOMO II, the rating scale for the DOCU cost driver is evaluated in terms of the suitability of the project’s documentation to its life-cycle needs. The rating scale goes from Very Low (many life-cycle needs uncovered) to Very High (very excessive for life-cycle needs)

Rating	Very Low	Low	Nominal	High	Very High
Effort Multipliers	0.81	0.91	1.00	1.11	1.

Value for DOCU cost driver is : 1 (Nominal)

Because our Documentation or SRS fully followed the SDLC life cycle.

Measurement Procedure : By Reviewing the SRS and SDLC

Platform Factors

6.Execution Time Constraint (TIME):

This is a measure of the execution time constraint imposed upon a software system. The rating is expressed in terms of the percentage of available execution time expected to be used by the system or subsystem consuming the execution time resource. The rating ranges from nominal, less than 50% of the execution time resource used, to extra high, 95% of the execution time resource is consumed.

Rating	Very Low	Low	Nominal	High	Very High
Effort Multipliers	n/a	n/a	1.00	1.11	1.29

Value for TIME cost driver is : 1.11 (High)

Because we used 70% of our available execution time.

Measurement Procedure : By Reviewing the Total Time Frame of the Project

7.Main Storage Constraint (STOR):

This rating represents the degree of main storage constraint imposed on a software system or subsystem. Given the remarkable increase in available processor execution time and main storage, one can question whether these constraint variables are still relevant. However, many applications continue to expand to consume whatever resources are available---particularly with large and growing COTS products---making these cost drivers still relevant. The rating ranges from nominal (less than 50%),

Rating	Very Low	Low	Nominal	High	Very High
Effort Multipliers	n/a	0.87	1.00	1.05	1.17

Value for STOR cost driver is : 1 (Nominal)

The system has a normal amount of main storage.

Measurement Procedure : By Reviewing the Main storage

8.Platform Volatility (PVOL):

"Platform" is used here to mean the complex of hardware and software (OS, DBMS, etc.) the software product calls on to perform its tasks. If the software to be developed is an operating system then the platform is the computer hardware. If a database management system is to be developed then the platform is the hardware and the operating system. If a network text browser is to be developed then the platform is the network, computer hardware, the operating system, Version 2.1 31 © 1995 – 2000 Center for Software Engineering, USC and the distributed information repositories. The platform includes any compilers or assemblers supporting the development of the software system. This rating ranges from low, where there is a major change every 12 months, to very high, where there is a major change every two weeks

Rating	Very Low	Low	Nominal	High	Very High
Effort Multipliers	n/a	0.87	1.00	1.15	1.30

Value for PVOL cost driver is : 0.87 (Low)

There is a low level of volatility, with the expectation of a major change approximately once a year.

Measurement Procedure : By Analysis the future scope of the project

9.Analyst Capability (ACAP):

Analysts are personnel who work on requirements, high-level design and detailed design. The major attributes that should be considered in this rating are analysis and design ability, efficiency and thoroughness, and the ability to communicate and cooperate. The rating should not consider the level of experience of the analyst; that is rated with APEX, LTEX, and PLEX. Analyst teams that fall in the fifteenth percentile are rated very low and those that fall in the ninetieth percentile are rated as very high

Rating	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.42	1.19	1.00	0.85	0.71	n/a

Value for PVOL cost driver is : 0.85 (High)

Because Analysts are highly experienced and capable.

Measurement Procedure : By Analysis the Capability of Analysts

10.Programmer Capability (PCAP) :

Current trends continue to emphasize the importance of highly capable analysts. However the increasing role of complex COTS packages, and the significant productivity leverage associated with programmers' ability to deal with these COTS packages, indicates a trend toward higher importance of programmer capability as well. Evaluation should be based on the capability of the programmers as a team rather than as individuals. Major factors which should be considered in the rating are ability, efficiency and thoroughness, and the ability to communicate and cooperate. it is rated with APEX, LTEX, and PLEX. A very low rated programmer team is in the fifteenth percentile and a very high rated programmer team is in the ninetieth percentile

Rating	Very Low	Low	Nominal	High	Very High
Effort Multipliers	1.34	1.15	1.00	0.88	0.76

Value for PCAP cost driver is : 0.88 (High)

Because Programmers are highly experienced and capable.

Measurement Procedure : By Analysis the Capability of Programmer, Problem solving approach, Recent works etc.

11.Personnel Continuity (PCON):

The rating scale for PCON is in terms of the project's annual personnel turnover: from 3%, very high continuity, to 48%, very low continuity,

Rating	Very Low	Low	Nominal	High	Very High
Effort Multipliers	1.29	1.12	1.00	0.90	0.81

Value for PCONcost driver is : 0.81 (Very High)

The project team has exceptional continuity, and turnovers are rare.

Measurement Procedure : By Analysis the Project Team Dedication and Work rate.

12.Applications Experience (APEX) :

The rating for this cost driver (formerly labeled AEXP) is dependent on the level of applications experience of the project team developing the software system or subsystem. The ratings are defined in terms of the project team's equivalent level of experience with this type of application. A very low rating is for application experience of less than 2 months. A very high rating is for experience of 6 years or more

Rating	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.22	1.10	1.00	0.88	0.81	n/a

Value for APEX cost driver is : 0.88 (High)

The project team has more than 2 years of experience in application development

Measurement Procedure : By Analysis the Project Team Portfolios.

13.Platform Experience (PLEX) :

The Post-Architecture model broadens the productivity influence of platform experience, PLEX (formerly labeled PEXP), by recognizing the importance of understanding the use of more powerful platforms, including more graphic user interface, database, networking, and distributed middleware capabilities

Rating	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.19	1.09	1.00	0.91	0.85	n/a

Value for PLEX cost driver is : 1.00 (Nominal)

Our project team has a typical level of experience with the development platform.

Measurement Procedure : By Analysis the Project Team member Portfolios.

14.Language and Tool Experience (LTEX) :

This is a measure of the level of programming language and software tool experience of the project team developing the software system or subsystem. Software development includes the use of tools that perform requirements and design representation and analysis, configuration management, document extraction, library management, program style and formatting, consistency checking, planning and control, etc. In addition to experience in the project's programming language, experience on the project's supporting tool set also affects development effort. A low rating is given for experience of less than 2 months. A very high rating is given for experience of 6 or more years

Rating	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.20	1.09	1.00	0.91	0.84	n/a

Value for LTEX cost driver is : 0.91 (High)

Our project team is highly experienced and familiar with the programming language and tools.
Measurement Procedure : By Analysis the Project Team member Portfolios and live assessment.

Project Factors

15.Use of Software Tools (TOOL):

Software tools have improved significantly since the 1970s' projects used to calibrate the 1981 version of COCOMO. The tool rating ranges from simple edit and code, very low, to integrated life-cycle management tools, very high. A Nominal TOOL rating in COCOMO 81 is equivalent to a Very Low TOOL rating in COCOMO II. An emerging extension of COCOMO II is in the process of elaborating the TOOL rating scale and breaking out the effects of TOOL capability, maturity, and integration

Rating	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.17	1.09	1.00	0.90	0.78	n/a

Value for TOOL cost driver is : 1.00 (Nominal)
Our project used some tools with limited automation support.
Measurement Procedure : By Analysis the Code

16.Multisite Development (SITE) :

Given the increasing frequency of multisite developments, and indications that multisite development effects are significant, the SITE cost driver has been added in COCOMO II. Determining its cost driver rating involves the assessment and judgement-based averaging of two factors: site collocation (from fully collocated to international distribution) and communication support (from surface mail and some phone access to full interactive multimedia). For example, if a team is fully collocated, it doesn't need interactive multimedia to achieve an Extra High rating.

Rating	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.22	1.09	1.00	0.93	0.86	0.80

Value for SITE cost driver is : 0.83 (High)

Our team was located on Same city or metro area and there were not miscommunication.

Measurement Procedure : By Analysis the location of our team members.

17.Required Development Schedule (SCED) :

SCED is the only cost driver that is used to describe the effect of schedule compression / expansion for the whole project. The scale factors are also used to describe the whole project. All of the other cost drivers are used to describe each module in a multiple module project.

Rating	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.43	1.14	1.00	1.00	1.00	n/a

Value for SCED cost driver is : 1.14 (Low)

The schedule is somewhat constrained; there is some pressure to accelerate development.

Measurement Procedure : By Analysis the project deadline and completion.

Cost Drivers For Early Stage

18.PERS Cost Driver:

Rating	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	2.12	1.62	1.26	0.83	0.63	0.50

Value for PERS cost driver is : 0.63 (Very High)

Measurement Procedure : By Analysis the project deadline and completion.

19.Product Reliability and Complexity (RCPX):

This Early Design cost driver combines the four Post-Architecture cost drivers Required software reliability (RELY), Database size (DATA), Product complexity (CPLX), and Documentation match to life-cycle needs (DOCU). Unlike the PERS components, the RCPX components have rating scales with differing width. RELY and DOCU range from Very Low to Very High; DATA ranges from Low to Very High, and CPLX ranges from Very Low to Extra High.

Rating	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	0.60	0.83	1.00	1.33	1.91	2.72

RCPX= RELY+DATA + CPLX + DOCU = Very Low + Nominal + Nominal + Nominal = 1+3+3+3 = 10

Effort Multipliers rating is Low and value is 0.93

20.Platform Difficulty (PDIF):

This Early Design cost driver combines the three Post-Architecture cost drivers Execution time constraint (TIME), Main storage constraint (STOR), and Platform volatility (PVOL). TIME and STOR range from Nominal to Extra High; PVOL ranges from Low to Very High.

Rating	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	0.87	1.00	1.29	1.81	n/a

PDIF = High + Nominal + Low = 4+3+2 = 9

Effort Multipliers rating is Nominal and value is 1.00

21.Personnel Experience (PREX):

This Early Design cost driver combines the three Post-Architecture cost drivers Application experience (APEX), Language and tool experience (LTEX), and Platform experience (PLEX). Each of these range from Very Low to Very High; as with PERS.

Rating	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.33	1.22	1.00	0.87	0.74	0.62

PREX = Sum of APEX, PLEX, and LTEX ratings = High + Nominal + High = 4+3+4 = 11

Effort Multipliers rating is High and value is 0.87

22.Facilities (FCIL):

This Early Design cost driver combines two Post-Architecture cost drivers: Use of software tools (TOOL) and Multisite development (SITE). TOOL ranges from Very Low to Very High; SITE ranges from Very Low to Extra High. Thus, the numerical sum of their ratings ranges from 2 (VL, VL) to 11 (VH, EH).

Rating	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.30	1.10	1.00	0.87	0.73	0.62

FCIL = TOOL + SITE = Nominal+High = 3+4 = 7

Effort Multipliers rating is High and value is 0.87

Early Design Cost Driver	**Post Architecture Cost Drivers**
PERS	ACAP, PCAP, PCON
RCPX	RELY, DATA, CPLX, DOCU

RUSE	RUSE
PDIF	TIME, STOR, PVOL
PREX	APEX, PLEX, LTEX
FCIL	TOOL, SITE
SCED	SCED

Stage2 : Early Design Model

It is used at the Stage – II in COCOMO -II models and supports estimation in early design stage of project. Base equation used in COCOMO II Model is as follows –

PMnominal = A * (size)^{B}

= 2.5 * (12)$^{1.0724}$

= 35.9131894836

where $PM_{nominal}$ = Effort for the project in person months

A = constant representing the nominal productivity where A = 2.5

B = Scale Factor = 0.91 + 0.01 * Sum of Scale Factors
(**Precedentness+Development Flexibility+Architecture Risk and Resolution +Team Cohesion+ Process Maturity)**

B = 0.91 + 0.01 * (3.72+4.05+4.25+1.10+3.12) = 1.0724

Size = 239.68 (Function Point of Our Project) * 50 = 12000 = 12 KLOC

Effort = PMnominal x EAF (Effort Adjustment Factor)
= 35.9131894836 * (PERS*RCPX*RUSE*PDIF*PREX*FCIL*SCED)
= 35.9131894836 * (0.63*0.93*1*1*0.87*0.87*1.14)
= 18.1560274 ~ 18 person - month

Schedule=C × $(\text{Effort})^{D}$
= 2.5 x $(18)^{1.05}$
=51.9967427744 ~ 52

As our project is a organic project so the value of
C = 2.5
D = 1.05

Cost=Effort×Average Cost Per Person-Month
= 18 x 1000$
= 18000$

Stage 2: Post- Architecture Model

The Post-Architecture model is the most detailed estimation model and it is intended to be used when a software lifecycle architecture has been developed. This model is used in the development and maintenance of software products in the Application Generators, System Integration, or Infrastructure sectors.

PMnominal = A * $(\text{size})^{B}$

= 2.5 * $(24)^{1.0724}$
= 75.5228802142

B = 0.91 + 0.01 * (3.72+4.05+4.25+1.10+3.12) = 1.0724

Effort = PMnominal x EAF (Effort Adjustment Factor for 17 cost drivers)
= 75.5228802142 *
(0.82*1*1*1*1*1.11*1*0.87*0.85*0.88*0.81*0.88*1*0.91*1*0.83*1.14)
= 35.9131894836 * (0.36353791)
= 13.0558 ~ 13 person - month

Schedule=C × $(\text{Effort})^{D}$
= 2.5 x $(13)^{1.05}$
=36.9471146803 ~ 37

As our project is a organic project so the value of
C = 2.5
D = 1.05

Cost=Effort × Average Cost Per Person-Month
= 13 x 1000$
= 13000$