**COST ANALYSIS MODEL**

**(considering lines of codes as the driving factor )**

**COCOMO Project Categories:**

Basic COCOMO computes software development effort (and cost) as a function of program size. Program size is expressed in estimated thousands of source lines of code (SLOC)

COCOMO applies to three classes of software projects:

■ Organic projects - In an area in which the organization has considerable expertise and requirements are less stringent e.g. simple data processing system

■ Semi-detached projects - Projects that fall between the above two categories, "medium" teams with mixed experience working with a mix of rigid and less than rigid requirements.

■ Embedded projects - Projects are ambitious and novel and the organization has little or

no prior experience in those areas and there are stringent requirements to be met e.g. embedded avionics system.

**Calculating Basic COCOMO for our project:**

Our project comes under the category “Semi-detached” because some of our group members have a little prior experience of the programming languages used i.e. PHP, CSS and Javascript which will be used in creating Learn-Hub .

Now, estimation of an initial effort is given by following formula:

Ei = a \* (KDLOC)b

where, KDLOC = estimated line of code in thousands

Lines of codes is the primary cost driving factor and since our project is semi-detached the value of the variables a and b are: a = 3.0 and b = 1.12. The Lines of code in our project is roughly 8,000.

So,the calculation is

Ei = 3 \* (8)1.12 = 30.8022

**COCOMO Cost Driver attributes**:

Intermediate COCOMO computes software development effort as function of program size and a set of "cost drivers" that include subjective assessment of product, hardware, personnel and project attributes. This extension considers a set of four "cost drivers", each with a number of subsidiary attributes:-

**Product attributes** : concerned with the required characteristics of the software product being developed.

■ Required software reliability

■ Size of application database

■ Complexity of the product

**Hardware attributes** : constraints imposed on the software by the hardware platform.

■ Run-time performance constraints

■ Memory constraints

■ Volatility of the virtual machine environment

■ Required turnabout time

**Personnel attributes** : Multipliers that take the experience and capabilities of the people working on the project into account .

■ Analyst capability

■ Software engineering capability

■ Applications experience

■ Virtual machine experience

■ Programming language experience

**Project attributes** : concerned with the particular characteristics of the software development project .

■ Use of software tools

■ Application of software engineering methods

■ Required development schedule

Each of the 15 attributes receives a rating on a six-point scale that ranges from "very low"

to "extra high" (in importance or value). An effort multiplier from the table below applies to the rating. The product of all effort multipliers results in an effort adjustment factor (EAF). Typical values for EAF range from 0.9 to 1.4.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Cost Drivers |  | | | | | |
| Very Low | Low | Nominal | High | Very High | Extra High |
| **Product attributes** | **Ratings** |  |  |  |  |  |
| Required software  Reliability | **0.75** | 0.88 | 1.00 | 1.15 | 1.40 |  |
| Size of application  Database |  | 0.94 | 1.00 | **1.08** | 1.16 |  |
| Complexity of the  Product | 0.70 | 0.85 | 1.00 | **1.15** | 1.30 | 1.65 |
| **Hardware attributes** |  |  |  |  |  |  |
| Run-time performance  constraints |  |  | **1.00** | 1.11 | 1.30 | 1.66 |
| Memory constraints |  |  | 1.00 | 1.06 | **1.21** | 1.56 |
| Volatility of the virtual machine environment |  | **0.87** | 1.00 | 1.15 | 1.30 |  |
| Required turnabout  time |  | 0.87 | **1.00** | 1.07 | 1.15 |  |
| **Personnel attributes** |  |  |  |  |  |  |
| Analyst capability | 1.46 | 1.19 | 1.00 | **0.86** | 0.71 |  |
| Applications  experience | 1.29 | **1.13** | 1.00 | 0.91 | 0.82 |  |
| Software engineer  capability | 1.42 | **1.17** | 1.00 | 0.86 | 0.70 |  |
| Virtual machine  experience | 1.21 | 1.10 | **1.00** | 0.90 |  |  |
| Programming  language experience | 1.14 | **1.07** | 1.00 | 0.95 |  |  |
| **Project attributes** |  |  |  |  |  |  |
| Application of software engineering methods | 1.24 | 1.10 | **1.00** | 0.91 | 0.82 |  |
| Use of software tools | 1.24 | 1.10 | **1.00** | 0.91 | 0.83 |  |
| Required development schedule | 1.23 | 1.08 | 1.00 | **1.04** | 1.10 |  |

So,

Effort adjustment Factor =0.75 \* 1.08\*1.15\*1.00\*1.21\*0.87\*1.00\*0.86\*1.13\*1.17\*1\*1.07\*1.00\*1.00\*1.04= 1.24

The final effort which is driven by the line of code is given by :

EAF \* Ei = 30.8022 \*1.24 =38.19 person-months

**Inference :**

The risk factors in our project included steep learning curve , considering the total effort that will be required in our project is 38.19 person- months and we are a group of 9 people.

Therefore , the time required to complete the project will be : 38.19/9 = 4.24 months.

Hence, we will require 4.24 “working “ months to complete our project .