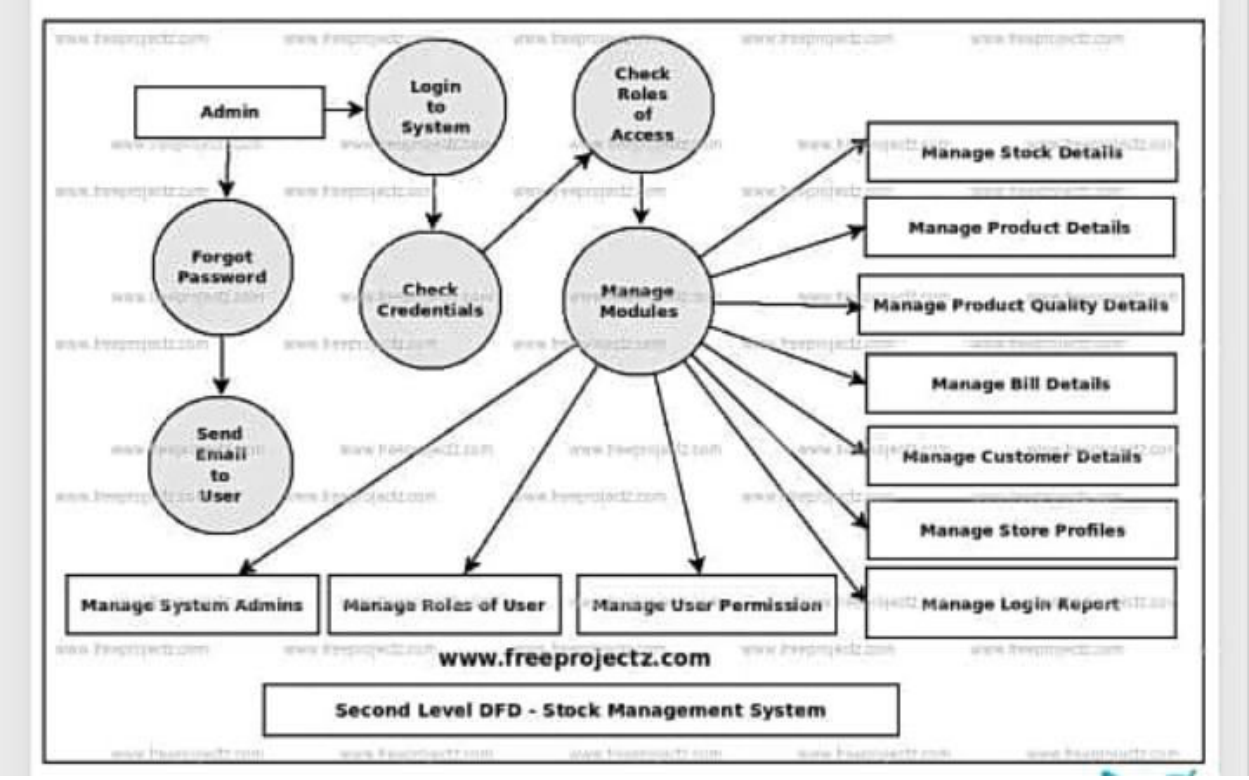


## 5.Attendance Management System Project using CONSTRUCTIVE COST ESTIMATION MODEL (COCOMO) Estimate the effort.

### Objective

To estimate effort for developing Attendance Management System project using COCOMO model. The COCOMO model reflects your software development environment and produces more accurate estimates. It computes software development effort(cost) as a function of program size expressed in estimated lines of code(LOC).The main objective of basic COCOMO model gives an approximate estimate of the project parameters. Several COCOMO packages allow the user to estimate the tasks outside the scope using a percentage of the estimated software development effort.

### Overview

The Constructive Cost Model (**COCOMO**) is an algorithmic software cost estimation model developed by Barry W. Boehm. The model uses a basic regression formula with parameters that are derived from historical project data and current as well as future project characteristics.

Boehm postulated that any software development project can be classified into one of the

following three categories based on the development complexity: organic, semidetached, and embedded. In order to classify a product into the identified categories, Boehm not only consideredthe characteristics of the product but also those of the development team and development environment. Roughly speaking, these three product classes correspond to application, utility and system programs, respectively. Normally, data processing programs are considered to be application programs. Compilers, linkers, etc., are utility programs. Operating systems and real- time system programs, etc. are system programs. System programs interact directly with the hardware and typically involve meeting timing constraints and concurrent processing.

Organic:

A development project can be considered of organic type, if the project deals with developing a well understood application program, the size of the development team is reasonably small, and the team members are experienced in developing similar types of projects.

Semi-detached:

A development project can be considered of semidetached type, if the development consists of a mixture of experienced and inexperienced staff. Team members may have limitedexperience on related systems but may be unfamiliar with some aspects of the system being developed.

Embedded:

A development project is considered to be of embedded type, if the software beingdeveloped is strongly coupled to complex hardware, or if the stringent regulations on the operational procedures exist.

The COCOMO cost estimation model is used by thousands of software project managers, and is based on a study of hundreds of software projects. Unlike other cost estimation models, COCOMO is an open model, so all of the details are published, including:

- The underlying cost estimation equations
- Every assumption made in the model (e.g. "the project will enjoy good management")
- Every definition (e.g. the precise definition of the Product Design phase of a project)
- The costs included in an estimate are explicitly stated (e.g. project managers are included, secretaries aren't)

The development time versus the product size in KLOC, it can be observed that the development time is a sub linear function of the size of the product, i. e. when the size of the product increases by two times, the time to develop the product does not double but rises moderately. This can be explained by the fact that for larger products, a larger number of activities which can be carried out concurrently can be identified. The parallel activities can be carried out simultaneously by the engineers. This reduces the time to complete the project. It canbe observed that the development time is roughly the same for all the three categories of products.For example, a 60 KLOC program can be developed in approximately 18 months,

regardless of whether it is of organic, semidetached, or embedded type.From the effort estimation, the project cost can be obtained by multiplying the required effort by the manpower cost per month. But, implicit in this project cost computation is the assumption that the entire project cost is incurred on account of the manpower cost alone. In addition to manpower cost, a project would incur costs due to hardware and software required for the project and the company overheads for administration, office space, etc.It is important to note that the effort and the duration estimations obtained using the COCOMO model are called as nominal effort estimate andnominal duration estimate. The term nominal implies that if anyone tries to complete the project in a time shorter than the estimated duration, then the cost will increase drastically. But, if anyone completes the project over a longer period of time than the estimated, then there is almost no decrease in the estimated cost value.

## Procedure

**BASIC COCOMO MODEL:** □

COCOMO consists of a hierarchy of three increasingly detailed and accurate forms.

1. The first level, Basic COCOMO is good for quick, early, rough order of magnitude estimatesof software costs, but its accuracy is limited due to its lack of factors to account for difference in project attributes (Cost Drivers).
2. Intermediate COCOMO takes these Cost Drivers into account and Detailed COCOMOadditionally accounts for the influence of individual project phases.
3. The Constructive Cost Model (COCOMO) is an algorithmic software cost estimation modeldeveloped by Barry Boehm. The model uses a basic regression formula, with parameters that are derived from historical project data and current project characteristics.

The basic COCOMO model gives an approximate estimate of the project parameters. The basic COCOMO estimation model is given by the following expressions: □

Effort = a1x (KLOC)^a2 pm

Tdev = b1x (Effort)^b2

MonthsP= Effort/ Tdev

where

- KLOC is the estimated size of the software product expressed in Kilo Lines of Code.
- P is the no of persons required to complete the work .
- a1, a2 , b1, b2 are constants for each category of software products.
- Tdev is the estimated time to develop the software, expressed in months.
- Effort is the total effort required to develop the software product, expressed in person-months (PMs).

The coefficients a1, a2 , b1, b2 for various types of software projects

| Software Projects | a1 | a2 | b1 | b2 |
|---|---|---|---|---|
| Organic | 2.4 | 1.05 | 2.5 | 0.35 |
| Semi-detached | 3.0 | 1.12 | 2.5 | 0.32 |
| Embedded | 3.6 | 1.20 | 2.5 | 0.38 |

**Estimation of development effort** :-for the three classes of software products, the formulas for estimating the effort based on the code size are shown below:

Organic : Effort = 2.4(KLOC)^1.05PM

Semi-detached: Effort =

3.0(KLOC)^1.12PMEmbedded : Effort =

3.6(KLOC)^1.20PM

For the three classes of software products, the formulas for estimating the development timebased on the effort are given below:

Organic: Tdev = 2.5(Effort)^0.38Months

Semidetached: Tdev = 2.5(Effort)^0.35Months

Embedded: Tdev = 2.5(Effort)^0.32Months.

Example:-

Effort Calculation for Attendance Maintenance System.

Consider Lines of Code = 10000

i.e value of KLOC is 10

Organic : Effort = $2.4(KLOC)^{1.05}$ PM

$= 2.4*(10)^{1.05}$

$= 2.4 * 11.220$

$= 26.92$ pm

Semi-detached: Effort = $3.0(10)^{1.12}$ PM

$=3.0*13.18$

$=39.5$ pm

Embedded : Effort = $3.6(10)^{1.20}$ PM

$= 3.6*15.84$

$= 57.02$PM

## 6).Calculating effort for Attendance Management System using FP Function Point Oriented estimation model.

### Objective

Calculating effort for Attendance Management System using Function Point oriented estimation model. It is a method to break systems into smaller components, so they can be better understood and analyzed. It is used to express the amount of business functionality, an information system (as a product) provides to a user. Fps measure software size. They are widely accepted as an industry standard for functional sizing. Function points are used to compute a functional size measurement (FSM) of software. The cost (in dollars or hours) of a single unit is calculated from past projects.Function Point Analysis can provide a mechanism to track and monitor scope creep. Function Point Counts at the end of requirements, analysis, design, code, testing and implementation can be compared. The function point count at the end of requirementsand/or designs can be compared to function points actually delivered. The amount of growth is an indication of how well requirements were gathered by and/or communicated to the project team. If the amount of growth of projects declines over time it is a natural assumption that communication with the user has improved.