SOFTWARE PROJECT MANAGEMENT LAB - G2
EXPERIMENT 5

- ASHISH KUMAR
- 2K18/SE/041

AIM:- Write a program to Implement detailed COCOMO model to determine cost and schedule estimates for different phases.

THEORY:-

Detailed COCOMO incorporates all characteristics of the intermediate version with an assessment of the cost driver's impact on each step of the software engineering process. The detailed model uses different effort multipliers for each cost driver attribute. In detailed COCOMO, the whole software is divided into different modules and then we apply COCOMO in different modules to estimate effort and then sum the effort.

The Six phases of detailed COCOMO are:

1. Planning and requirements
2. System design
3. Detailed design
4. Module code and test
5. Integration and test
6. Cost Constructive model

The effort is calculated as a function of program size and a set of cost drivers are given according to each phase of the software lifecycle.

CODE:-

```
#include <iostream>
#include<bits/stdc++.h>
using namespace std;

int main()
{
   cout << "\n---------- DETAILED COCOMO MODEL ----------";
   cout << "\n\n Enter Estimated Size  (in KLOC): ";
   double S;
   cin >> S;
   int mode = -1;
   if (S < 50)
      mode = 0;
```

```
else if (S >= 50 && S < 300)
   mode = 1;
else
   mode = 2;
double coeffs[3][4] = {{2.4, 1.05, 2.5, 0.38},
                       {3.0, 1.12, 2.5, 0.35},
                       {3.6, 1.20, 2.5, 0.32}};
double costDriversVal[15][6] = {{0.75, 0.88, 1.00, 1.15, 1.40, 0.00},
                                {0.00, 0.94, 1.00, 1.08, 1.16, 0.00},
                                {0.70, 0.85, 1.00, 1.15, 1.30, 1.65},
                                {0.00, 0.00, 1.00, 1.11, 1.30, 1.66},
                                {0.00, 0.00, 1.00, 1.06, 1.21, 1.56},
                                {0.00, 0.87, 1.00, 1.15, 1.30, 1.65},
                                {0.00, 0.87, 1.00, 1.07, 1.15, 0.00},
                                {1.46, 1.19, 1.00, 0.86, 0.71, 0.00},
                                {1.29, 1.13, 1.00, 0.91, 0.82, 0.00},
                                {1.42, 1.17, 1.00, 0.86, 0.70, 0.00},
                                {1.21, 1.10, 1.00, 0.90, 0.00, 0.00},
                                {1.14, 1.07, 1.00, 0.95, 0.00, 0.00},
                                {1.24, 1.10, 1.00, 0.91, 0.82, 0.00},
                                {1.24, 1.10, 1.00, 0.91, 0.83, 0.00},
                                {1.23, 1.08, 1.00, 1.04, 1.10, 0.00}};
string tags[15] = {"Required software reliability extent              : ",
                   "Size of the application database                  : ",
                   "The complexity of the product                     : ",
                   "Run-time performance constraints                  : ",
                   "Memory constraints                                : ",
                   "The volatility of the virtual machine environment      : ",
                   "Required turnabout time                           : ",
                   "Analyst capability                                : ",
                   "Software engineering capability                   : ",
                   "Applications experience                           : ",
                   "Virtual machine experience                        : ",
                   "Programming language experience                   : ",
                   "Use of software tools                             : ",
                   "Application of software engineering methods       : ",
                   "Required development schedule                     : "};
string types[3] = {"Organic", "Semi-Detached", "Embedded"};
string phase_names[5] = {"Plan/requirement", "System Design", "Detail Design", "Module
                         code & Test", "Integration and Test"};
double u_coeffs[6][5] = {{0.06, 0.16, 0.26, 0.42, 0.06},
                         {0.06, 0.16, 0.24, 0.38, 0.22},
                         {0.07, 0.17, 0.25, 0.33, 0.25},
                         {0.07, 0.17, 0.24, 0.31, 0.28},
                         {0.08, 0.18, 0.25, 0.26, 0.31},
                         {0.08, 0.18, 0.24, 0.24, 0.34}};
```

```
    double t_coeffs[6][5] = {{0.10, 0.19, 0.24, 0.39, 0.18},
                             {0.12, 0.19, 0.21, 0.34, 0.26},
                             {0.20, 0.26, 0.21, 0.27, 0.26},
                             {0.22, 0.27, 0.19, 0.25, 0.29},
                             {0.36, 0.36, 0.18, 0.18, 0.28},
                             {0.40, 0.38, 0.16, 0.16, 0.30}};

    int phase_mode = -1;

    if (mode == 0)
       phase_mode = abs(S - 2) < abs(S - 32) ? 0 : 1;
    else if (mode == 1)
       phase_mode = abs(S - 32) < abs(S - 128) ? 2 : 3;
    else if (mode == 2)
       phase_mode = abs(S - 128) < abs(S-320) ? 4 : 5 ;
          cout << "\n It is " << types[mode] << " Mode. \n";
    int costs_idx[15];
    cout << "\n Enter the Cost Drivers of the following 15 attributes on a scale from (0-5)\n";
    for (int i = 0; i < 15; i++)
    {
       cout << " " << i + 1 << ". " << tags[i];
       cin >> costs_idx[i];
    }

    double eaf = 1;
    for (int j = 0; j < 15; j++)
       eaf *= costDriversVal[j][costs_idx[j]];
    cout <<"\nEAF = "<<eaf<<endl;
    double Effort = coeffs[mode][0] * pow(S, coeffs[mode][1]) * eaf;
    double Time = coeffs[mode][2] * pow(Effort, coeffs[mode][3]);
    cout << "\nTotal Effort = " << Effort << " person-months";
    cout << "\nTotal Development Time   = " << Time << " months";
    cout << "\nTotal Persons Required = " << ceil(Effort / Time) << " persons\n\n";
    cout << "----------------------------------------------------------------\n\n";
    cout << "EFFORT AND DEVELOPMENT ESTIMATED IN DIFFERENT PHASES \n\n";

    for (int i = 0; i < 5; i++)
    {
       cout<<" "<<i+1<<". "<<phase_names[i]<<" Phase  \n";
       double Ep = u_coeffs[phase_mode][i] * Effort;
       double Dp = t_coeffs[phase_mode][i] * Time;
       cout<<"     Effort          : "<<Ep<<" person-months\n";
       cout<<"     Development Time : "<<Dp<<" months\n\n";
    }
    return 0;
}
```

OUTPUT:-

Finding & Learning: - We have successfully implemented detailed COCOMO model and calculated effort, development time and cost estimates for different phases. COCOMO model is a very easy procedural cost estimation model for software projects.