SOFTWARE PROJECT MANAGEMENT LAB - G2
EXPERIMENT 7

- ASHISH KUMAR
- 2K18/SE/041

AIM:- Write a program to implement Early Design Model and calculate the effort for the development of project.

THEORY:-

Early design model 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 models is as follows –
P_Mnominal = A * (size) ^B
where,
P_Mnominal = Effort for the project in person months
A = constant representing the nominal productivity where A = 2.5
B = Scale Factor
Size = size of the Software

The value of B can be calculated as -
B = 0.91 + 0.01 * (Sum of rating scaling factors for project)

When all scaling factors of project are rated as extra high, best value of B is obtained which is equal to 0.91.
When all scaling factors are very low worst case values of B is obtained as 1.23. Hence value of B varies from 0.91 to 1.23.

The early design model uses Unadjusted Function Points (UFP) as measure of size. This model is used at the early stages of software project when there is not enough information available about size of product which has to be developed, nature of target platform and nature of employees to be involved in development of projector detailed specifications of process to be used. This model can be used either in Application Generator, System Integration, or Infrastructure Development Sector.

Here the basic assumption stands as effort on project usually increases faster than size of product. However, value of 'B' is computed on basis of scaling factors that may cause loss of productivity corresponding to an increase in the size as follows –

1. **Precedentness :**
   It reflects the experience on similar projects previously. This is applicable to individuals as well as organizations both in terms of expertise and experience. High value would imply that

organization is quite familiar with application formula and very low value means no previous experience or expertise. The value for scale factor is 6.20(Very Low), 4.96(Low), 3.72(Average), 2.48(High), 1.25(Very High) and 0.00(Extra High).

2. 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. Value for the scale factor is 5.07(Very Low), 4.05(Low), 3.04(Average), 2.03(High), 1.02(Very High) and 0.00(Extra High).

3. 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. Value for the scale factor is 7.07(Very Low), 5.65(Low), 4.24(Average), 2.83(High), 1.41(Very High) and 0.00(Extra High).

4. 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. The value for scale factor is 5.48(Very Low), 4.38(Low), 3.29(Average), 2.19(High), 1.10(Very High) and 0.00(Extra High).

5. 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. The value for scale factor is 7.80(Very Low), 6.24(Low), 4.68(Average), 3.12(High), 1.56(Very High) and 0.00(Extra High).

CODE:-

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

int main() {
  double loc, B = 0, E = 1, PM_nominal, PM_adjusted, y;
  int x;
  double scaling_factors[5][6] = {
    {     //for "Precedentness "
      6.20,
      4.96,
      3.72,
      2.48,
      1.24,
      0.00
```

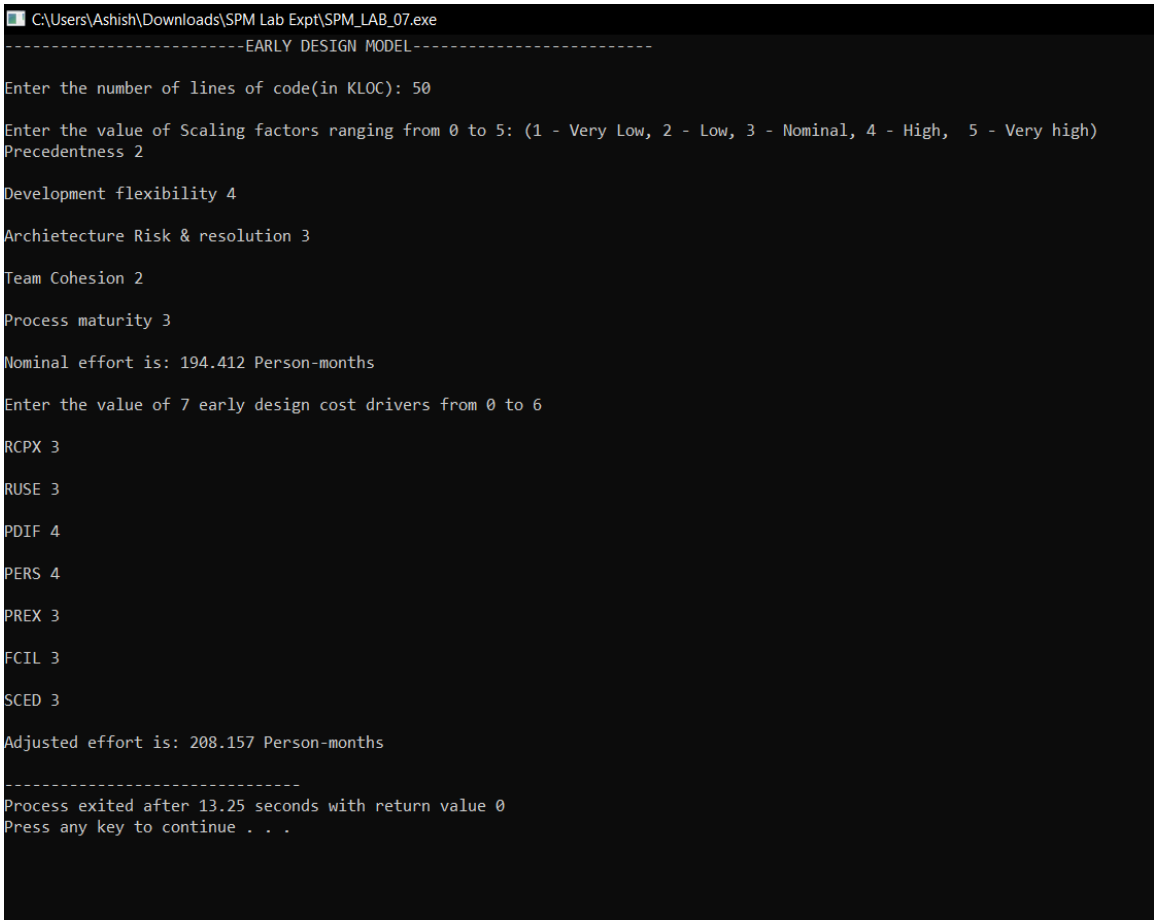
```
  },
  {  //for "Development flexibility "
    5.07,
    4.05,
    3.04,
    2.03,
    1.01,
    0.00
  },
  {    //for "Archietecture Risk & resolution "
    7.07,
    5.65,
    4.24,
    2.83,
    1.41,
    0.00
  },
  {   //for "Team Cohesion "
    5.48,
    4.38,
    3.29,
    2.19,
    1.10,
    0.00
  },
  {    //for "Process maturity "
    7.80,
    6.24,
    4.68,
    3.12,
    1.56,
    0.00
  }
};
double cost_drivers[7][7] = {
  {
    0.73,
    0.81,
    0.98,
    1.0,
    1.3,
    1.74,
    2.38
  },
```

```
{
    -1,
    -1,
    0.95,
    1.0,
    1.07,
    1.15,
    1.24
  },
  {
    -1,
    -1,
    0.87,
    1.0,
    1.29,
    1.81,
    2.61
  },
  {
    2.12,
    1.62,
    1.26,
    1.0,
    0.83,
    0.63,
    0.5
  },
  {
    1.59,
    1.33,
    1.12,
    1.0,
    0.87,
    0.71,
    0.62
  },
  {
    1.43,
    1.30,
    1.10,
    1.0,
    0.87,
    0.73,
    0.62
  },
```

```
{
    -1,
    1.43,
    1.14,
    1.0,
    1.0,
    1.0,
    -1
   }
  };
  string factors[5] = {
   "Precedentness ",
   "Development flexibility ",
   "Archietecture Risk & resolution ",
   "Team Cohesion ",
   "Process maturity "
  };
  string drivers[7] = {
   "RCPX ",
   "RUSE ",
   "PDIF ",
   "PERS ",
   "PREX ",
   "FCIL ",
   "SCED "
  };
  cout<<"-------------------------EARLY DESIGN MODEL-------------------------"<<endl;
  cout << "\nEnter the number of lines of code(in KLOC): ";
  cin >> loc;
  cout << endl <<"Enter the value of Scaling factors ranging from 0 to 5: (1 - Very Low, 2 - Low,
  3 - Nominal, 4 - High,  5 - Very high) ";
  for (int i = 0; i < 5; i++) {
   cout << endl << factors[i];
   cin >> x;
   B += scaling_factors[i][x];
  }
B * = 0.01;
B + = 0.91;
PM_nominal = 2.5 * pow(loc, B);
cout << endl << "Nominal effort is: " << PM_nominal << " Person-months"<< endl;
  x = 0;
cout << endl << "Enter the value of 7 early design cost drivers from 0 to 6" << endl;
  for (int i = 0; i < 7; i++) {
   do {
    cout << endl << drivers[i];
    cin >> x;
```

```
      y = cost_drivers[i][x];
      if (y ! =  -1) {
        E *= y;
      } else {
        cout << endl << "Entered value is not permissable, please enter another value";
      }
    } while (y = = -1);
  }
  PM_adjusted = PM_nominal * E;
  cout << "\nAdjusted effort is: " << PM_adjusted << " Person-months" << endl;
  return 0;
}
```

OUTPUT:-



Finding & Learning: - We have successfully implemented Early Design Model for effort estimation. COCOMO-II model is a very easy procedural cost estimation model for software projects.