

SOFTWARE PROJECT MANAGEMENT LAB - G2

EXPERIMENT 6

- ASHISH KUMAR

- 2K18/SE/041

AIM:- Write a program to implement Application Composition Estimation Model (COCOMO-II) for effort estimation.

THEORY:- Application Composition Estimation Model allows one to estimate the cost, effort at the stage 1 of the COCOMO II Model.

In this model size is first estimated using Object Points. Object Points are easy to identify and count. Object Points defines screen reports, third generation (3GL) modules as objects. Object Point estimation is a new size estimation technique but it is well suited in the Application Composition Sector.

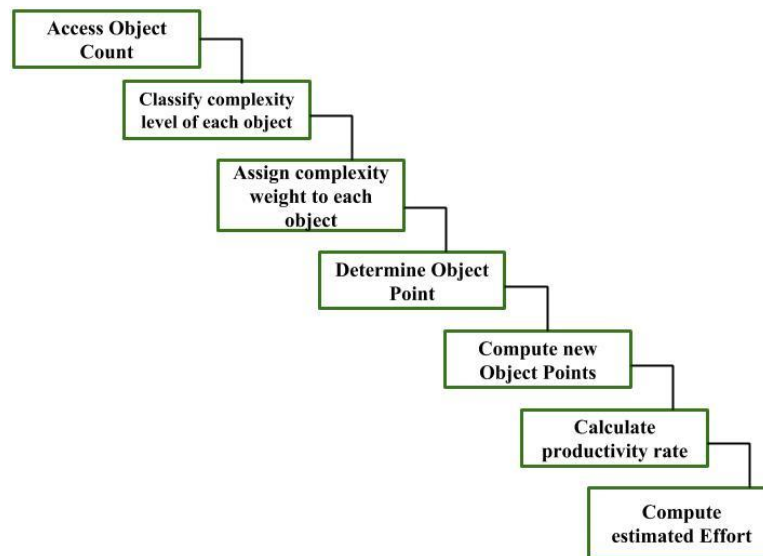


Fig. Steps are taken to estimate effort to develop a project

CODE:-

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

int screens, views, s_dataTables;
int reports, sections, r_dataTables;
int developerExperience;
int reuse;
double OP; // Object Points
double NOP; // New Object Points
double Effort;
string screen_type[3][3] = {
    {
        "simple",
        "simple",
        "medium"
    },
    {
        "simple",
        "medium",
        "difficult"
    },
    {
        "medium",
        "difficult",
        "difficult"
    }
};
string report_type[3][3] = {
    {
        "simple",
        "simple",
        "medium"
    },
    {
        "simple",
        "medium",
        "difficult"
    },
    {
        "medium",
        "difficult",
        "difficult"
    }
};
```

```

int complexity_weight[3][3] = {
    {
        1,
        2,
        3
    },
    {
        2,
        5,
        8
    },
    {
        0,
        0,
        10
    }
};
int developerprodValues[6] = {-1, 4, 7, 13, 25, 50};
int main() {
    cout << "\nEnter Number of Screens: ";
    cin >> screens;
    cout << "Enter Number of Views in each screen: ";
    cin >> views;
    cout << "Enter Total Number of Sources and Data Table for each screen: ";
    cin >> s_dataTables;
    cout << "\n\nEnter Number of Reports: ";
    cin >> reports;
    cout << "Enter Number of Sections in each Report: ";
    cin >> sections;
    cout << "Enter Total Number of Sources and Data Table: ";
    cin >> r_dataTables;
    cout << "\nOn a scale from ( 1 - 5 ) rate the Developer Experience(1 - Very Low, 2 - Low, 3 - Nominal, 4 - High, 5 - Very high): ";
    cin >> developerExperience;
    cout << "\nEnter percentage of code being reused: ";
    cin >> reuse;
    int i = 0, j = 0;
    int s_weight;
    if (views < 3)
        i = 0;
    else if (views >= 3 && views <= 7)
        i = 1;
    else
        i = 2;
    if (s_dataTables < 4)
        j = 0;

```

```

else if (s_dataTables < 8)
    j = 1;
else
    j = 2;
if (screen_type[i][j] == "simple")
    s_weight = 0;
else if (screen_type[i][j] == "medium")
    s_weight = 1;
else
    s_weight = 2;
s_weight = complexity_weight[0][s_weight];
int r_weight;
if (sections <= 1)
    i = 0;
else if (sections >= 2 && sections <= 3)
    i = 1;
else
    i = 2;
if (r_dataTables < 4)
    j = 0;
else if (r_dataTables < 8)
    j = 1;
else
    j = 2;
if (report_type[i][j] == "simple")
    r_weight = 0;
else if (report_type[i][j] == "medium")
    r_weight = 1;
else
    r_weight = 2;
r_weight = complexity_weight[1][r_weight];

OP = screens * s_weight + reports * r_weight ;
NOP = (OP * (100 - reuse)) / 100;
cout << "\n\n-----";
cout << "\n\nObject Points OP = " << OP;
cout << "\nNew Object Points NOP = " << NOP;
cout << "\nEffort = " << NOP / developerprodValues[developerExperience] << " Person
Months ";
return 0;
}

```

OUTPUT:-

```
C:\Users\Ashish\Downloads\SPM Lab Expt\SPM_LAB_06.exe

Enter Number of Screens: 4
Enter Number of Views in each screen: 4
Enter Total Number of Sources and Data Table for each screen: 7

Enter Number of Reports: 2
Enter Number of Sections in each Report: 6
Enter Total Number of Sources and Data Table: 7

On a scale from ( 1 - 5 ) rate the Developer Experience(1 - Very Low, 2 - Low, 3 - Nominal, 4 - High, 5 - Very high): 2

Enter percentage of code being reused: 10

-----

Object Points OP = 24
New Object Points NOP = 21.6
Effort = 3.08571 Person Months
-----

Process exited after 36.09 seconds with return value 0
Press any key to continue . . .
```

Finding & Learning: - We have successfully implemented Application Composition Estimation Model (COCOMO-II) for effort estimation. COCOMO-II model is a very easy procedural cost estimation model for software projects.