Practical-8 27-Oct-2017

Function Point Model

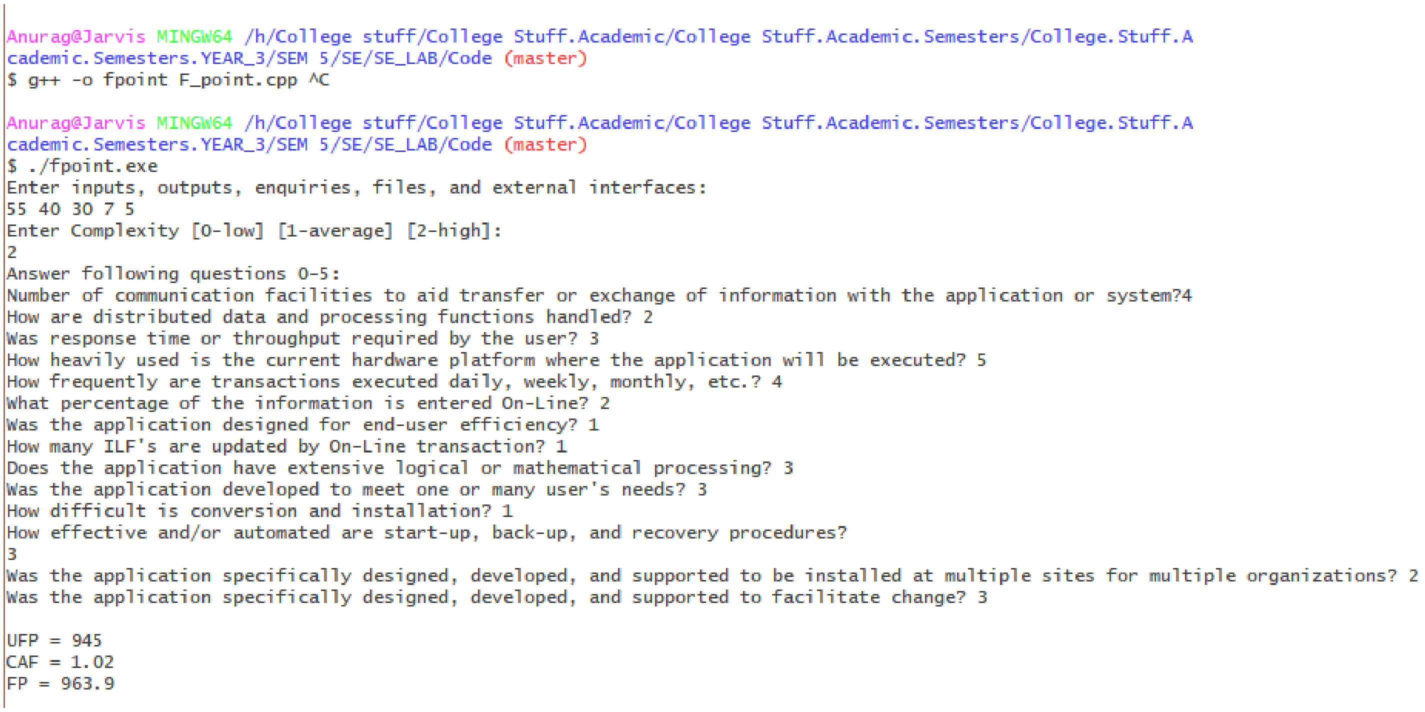
Introduction

The functional user requirements of the software are identified and each one is categorized into one of five types: outputs, inquiries, inputs, internal files, and external interfaces. Once the function is identified and categorized into a type, it is then assessed for complexity and assigned a number of function points. Each of these functional user requirements maps to an end-user business function, such as a data entry for an Input or a user query for an Inquiry. This distinction is important because it tends to make the functions measured in function points map easily into user-oriented requirements, but it also tends to hide internal functions (e.g. algorithms), which also require resources to implement.

Code

```
#include <iostream>
#include <cmath>

float FP[][3] = {
        3,  4,  6,
        4,  5,  7,
        3,  4,  6,
        7, 10, 15,
        5,  7, 10
};

int main() {
    float inp[5];
    int complex = 0;
    float ufp = 0, caf = 0;
    float influence[14];
    std::cout << "Enter inputs, outputs, enquiries, files, and external inter
faces:\n";
    std::cin >> inp[0] >> inp[1] >> inp[2] >> inp[3] >> inp[4];

    std::cout << "Enter Complexity [0-low] [1-average] [2-high]:\n";
    std::cin >> complex;

    std::cout << "Answer following questions 0-5:\n";
    std::string questions[14] = {
            "Number of communication facilities to aid transfer or exchange o
f information with the application or system?",
            "How are distributed data and processing functions handled?",
            "Was response time or throughput required by the user?",
            "How heavily used is the current hardware platform where the appl
ication will be executed?",
            "How frequently are transactions executed daily, weekly, monthly, etc.?",
            "What percentage of the information is entered On-Line?",
            "Was the application designed for end-user efficiency?",
            "How many ILF's are updated by On-Line transaction?",
            "Does the application have extensive logical or mathematical processing?",
            "Was the application developed to meet one or many user's needs?",
            "How difficult is conversion and installation?",
            "How effective and/or automated are start-up, back-up, and recovery procedures?",
            "Was the application specifically designed, developed, and supported to be installed at multiple sites for multiple organizations?",
            "Was the application specifically designed, developed, and supported to facilitate change?"
    };
    for(int i=0; i<14; i++){
        std::cout << questions[i] << " ";
        std::cin >> influence[i];
    }


    for(int i=0; i<5; i++){
        ufp += FP[i][complex] * inp[i];
    }

    for(int i=0; i<14; i++){
        caf += influence[i];
    }
    caf *= 0.01;
    caf += 0.65;
    std::cout << "\nUFP = " << ufp << "\nCAF = " << caf << "\nFP = " << ufp*caf << std::endl;

    return 0;
}
```

Output



```
Anurag@Jarvis MINGW64 /h/College stuff/College Stuff.Academic/College Stuff.Academic.Semesters/College.Stuff.A
cademic.Semesters.YEAR_3/SEM 5/SE/SE_LAB/Code (master)
$ g++ -o fpoint F_point.cpp ^C

Anurag@Jarvis MINGW64 /h/College stuff/College Stuff.Academic/College Stuff.Academic.Semesters/College.Stuff.A
cademic.Semesters.YEAR_3/SEM 5/SE/SE_LAB/Code (master)
$ ./fpoint.exe
Enter inputs, outputs, enquiries, files, and external interfaces:
55 40 30 7 5
Enter Complexity [0-low] [1-average] [2-high]:
2
Answer following questions 0-5:
Number of communication facilities to aid transfer or exchange of information with the application or system?4
How are distributed data and processing functions handled? 2
Was response time or throughput required by the user? 3
How heavily used is the current hardware platform where the application will be executed? 5
How frequently are transactions executed daily, weekly, monthly, etc.? 4
What percentage of the information is entered On-Line? 2
Was the application designed for end-user efficiency? 1
How many ILF's are updated by On-Line transaction? 1
Does the application have extensive logical or mathematical processing? 3
Was the application developed to meet one or many user's needs? 3
How difficult is conversion and installation? 1
How effective and/or automated are start-up, back-up, and recovery procedures?
3
Was the application specifically designed, developed, and supported to be installed at multiple sites for multiple organizations? 2
Was the application specifically designed, developed, and supported to facilitate change? 3

UFP = 945
CAF = 1.02
FP = 963.9
```

Learning

We implement the function point model through code where we estimate the functionality metrics by accepting different data from user. A function point is a "unit of measurement" to express the amount of business functionality an information system (as a product) provides to a user. 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.