

Experiment No. 5

Aim: Use of metrics to estimate the cost

Problem Statement: To estimate the cost using Function Point.

Theory:

Software Cost Estimation:

- Software cost estimation is the process of predicting the amount of effort required to build a software system and time develop it.
- Models provide mathematical algorithms to compute cost as a function of a number of variables such as size (using lines of code, function points, etc.) and/or complexity (using cyclomatic complexity, etc.).

Function Point:

Allan J. Albrecht initially developed Function Point Analysis in 1979 at IBM and it has been further modified by the International Function Point Users Group (IFPUG). FPA is used to make estimate of the software project, including its testing in terms of functionality or function size of the software product. However, functional point analysis may be used for the test estimation of the product. The functional size of the product is measured in terms of the function point, which is a standard of measurement to measure the software application.

Objectives of FPA

The basic and primary purpose of the functional point analysis is to measure and provide the software application functional size to the client, customer, and the stakeholder on their request. Further, it is used to measure the software project development along with its maintenance, consistently throughout the project irrespective of the tools and the technologies.

Following are the points regarding FPs

FPs of an application is found out by counting the number and types of functions used in the applications. Various functions used in an application can be put under five types, as shown in Table:

Types of FP Attributes

Measurements Parameters	Examples
1.Number of External Inputs(EI)	Input screen and tables
2. Number of External Output (EO)	Output screens and reports
3. Number of external inquiries (EQ)	Prompts and interrupts.
4. Number of internal files (ILF)	Databases and directories

5. Number of external interfaces (EIF)	Shared databases and shared routines.
--	---------------------------------------

All these parameters are then individually assessed for complexity.

Value Adjustment Factor:

The value adjustment factor (VAF) is based on 14 general system characteristics (GSC's) that rate the general functionality of the application being counted. Each characteristic has associated descriptions to determine the degrees of influence.

Rating:

The degrees of influence range on a scale of zero to five, from no influence to strong influence. Each characteristic is assigned the rating based upon detail descriptions provided by the IFPUG 4.1 Manual. They ratings are:

0 - Not present, or no influence

1 - Incidental influence

2 - Moderate influence

3 - Average influence

4 - Significant influence

5 - Strong influence throughout

Value adjustment equation

$$VAF = 0.65 + [(\sum_{i=1}^{14} Ci) / 100]$$

GSC's at a Glance:

General System Characteristic		Brief Description
1	Data communications	How many communication facilities are there to aid in the transfer or exchange of information with the application or system?
2	Distributed data processing	How are distributed data and processing functions handled?

3	Performance	Did the user require response time or throughput?
4	Heavily used configuration	How heavily used is the current hardware platform where the application will be executed?
5	Transaction rate	How frequently are transactions executed daily, weekly, monthly, etc.?
6	On-Line data entry	What percentage of the information is entered On-Line?
7	End-user efficiency	Was the application designed for end-user efficiency?
8	On-Line update	How many ILF's are updated by On-Line transaction?
9	Complex processing	Does the application have extensive logical or mathematical processing?
10	Reusability	Was the application developed to meet one or many user's needs?
11	Installation ease	How difficult is conversion and installation?
12	Operational ease	How effective and/or automated are start-up, back up, and recovery procedures?
13	Multiple sites	Was the application specifically designed, developed, and supported to be installed at multiple sites for multiple organizations?
14	Facilitate change	Was the application specifically designed, developed, and supported to facilitate change?

Calculating FP:Computing Function points (FPs):-

Information Domain value	Count	Weight Factor			
		Simple	Average	Complex	
1. External Inputs (EI)	20	3	(4)	6	= 80
2. External Outputs (EO)	20	4	(5)	7	= 100
3. External Inquiries (EQ)	13	3	(4)	6	= 52
4. Internal Logical Files (ILF)	19	7	(10)	15	= 190
5. External Interface Files (EIF)	3	5	(7)	10	= 21
Count total:-					→ 443

Value adjustment factors:-

Factor	Value
1) Backup & Recovery	5
2) Data Communications	3
3) Distributed Processing	0
4) Performance Critical	4
5) Existing operating Environment	4
6) Online data entry	5
7) Input transaction over multiple screen	5
8) Master files updated online	4
9) Information domain values complex	3
10) Internal processing complex	4
11) Code designed for reuse	2

Factor	Value
12) Conversion / installation in design	2
13) Multiple installations	5
14) Application design for change	5
$\sum F_i$	51

$$F.P = \text{Counts-total} * \text{Value Adjustment Factor (CAF)}$$

C.A.F also known as Complexity Adjustment Factor:

$$C.A.F = 0.01 + 0.65 + 0.01 * (\sum F_i)$$

$$C.A.F = 1.16$$

$$\therefore F.P = 443 * 1.16 = \boxed{513.88}$$

Conclusion: We have successfully estimated the cost of project using Function Point Metrics