

## 1. PURPOSE OF FPA

In 1979, Allan Albrecht of IBM developed the Function Point Analysis (FPA) in response to a number of problems with other system size measures, such as lines of codes. Before the launch of FPA, the “Lines of Code” method could be considered as one of the main measurement techniques for system size. However, its results varied greatly depending on programming languages used. It was also not applicable at the early stage of the development life cycle.

Albrecht wanted to have a measure of size that would allow different types of systems to be compared and would be independent of the technology applied. The measure should be meaningful to the end-user or purchaser of the system, and could be easily extracted early on in the system development cycle. It measured a system through the analysis of functions related to the user.

## 2. APPLICATION OF FPA

- A tool to determine the size of a purchased application package by counting all the functions included in the package
- A tool to help users determine the benefit of an application package to their organization by counting functions that specifically match their requirements
- A tool to measure the units of a software product to support quality and productivity analysis
- A vehicle to estimate cost and resources required for software development and maintenance
- A normalization factor for software comparison

### 3. COMPONENTS OF FUNCTION POINT ANALYSIS

FPA measures the size of an application system in 2 areas: the *specific user functionality* and the *system characteristics*.

The specific user functionality, as the name implies, is a measurement of the functionality delivered by the application as for user request. The 5 function types identified are *external input*, *external output*, *external enquiries*, *internal logical files* and *external interface files*. For each function identified under one of the above 5 function types, the function is further classified as *simple*, *average* or *complex* and a weight is given to each. The sum of the weights quantifies the size of information processing and it is referred as the *Unadjusted Function Points*.

$$\text{Function Point} = (\text{User Functionality}) \times (\text{System Characteristics})$$

The general functionality of the systems will be affected by some system characteristics. Fourteen *general applications attributes* are identified to rate the general functionality of the system. A *degree of influence (DI)* ranges from zero to five, from no influence to strong influence, is determined for each of the general applications attributes. The sum of all these DIs will in turn determine a Value Adjustment Factor for the whole projects.

The product of the Unadjusted Function Point and Value Adjustment Factor gives the size of the application expressed in term of Adjusted Function Point.

$\begin{aligned} &\text{Adjusted Function Point} \\ &= (\text{Unadjusted Function Point}) \times (\text{Value Adjustment Factor}) \end{aligned}$
--

## 4. STEPS OF FUNCTION POINT ANALYSIS

The major steps of the Function Point Analysis are :-

- (a) Determine the type of function point count;
- (b) Identify the application boundary;
- (c) Determine the unadjusted function point;
- (d) Determine the value adjustment factor; and
- (e) Calculate the final adjusted function point.

### 4.1 DETERMINE THE TYPE OF FUNCTION POINT COUNT

FPA technique applies different formula when measuring system size for software development and for software maintenance. Therefore, the type of function point count should be determined at the outset.

There are 3 types of function point counts:

- (a) Development project function point count<sup>1</sup>
- (b) Enhancement project function point count<sup>2</sup>
- (c) Application function point count<sup>3</sup>

Function counts at the early stage of a project are essentially estimation of the delivered functionality only. Towards completion, the scope is clarified and the delivered functionality might be changed. Hence update to function counts would be required.

### 4.2 IDENTIFY THE APPLICATION BOUNDARY

A boundary, which defines the system viewed by the users and determines any interaction with other systems, should first be determined so to set up the scope for the related functions to be identified.

### 4.3 DETERMINE THE UNADJUSTED FUNCTION POINT

The unadjusted function point reflects the functionality of a **logical** system provided to the user. Only **user-requested** and approved business functions are counted.

---

<sup>1</sup> The *development project function point count* measures the functions provided to the end users with the first installation of the software delivered when the project is complete.

<sup>2</sup> The *enhancement project function point count* measures the modifications to the existing application that add, change, or delete user functions delivered when the project is complete.

<sup>3</sup> The *application function point count* is associated with an installed application. It is also referred to as the baseline or installed function point count. This count provides a measure of the current functions the application provides the end-user. This number is initialised when the development project function point count is completed. It is updated every time completion of an enhancement project alters the application's functions.

As mentioned in section 6, the unadjusted function point is determined based on five function types:

- (a) An Internal Logical File (ILF)<sup>4</sup> is a user identifiable group of logically related data or control information maintained within the boundary of the application being counted.
- (b) An External Interface File (EIF)<sup>4</sup> is a user identifiable group of logically related data or control information maintained outside the boundary of the application being counted.
- (c) An External Input (EI)<sup>4</sup> processes data or control information that comes from outside the boundary of the application being counted.
- (d) The External Output (EO)<sup>4</sup> generates data or control information send outside the boundary of the application being counted.
- (e) An External Enquiry (EQ)<sup>4</sup> represents a combination of input (request) and output (retrieval).

Each individual function type is then assessed for its complexity (simple, average or complex) as follows:-

- (i) IT, OT and QT are given complexity ratings depending on the number of file type referenced (FTR) and data element types (DET) maintained/referenced; and
- (ii) EI and FT are given complexity ratings depending on the number of record element types (RET) and data element types (DET) in the file.

According to the function's complexity, function point count of an user function is assigned based on the rate table below. The summation of all function point counts will then render the unadjusted function point of the system.

Function Type	Complexity		
	Simple	Average	Complex
External Input (IT)	3	4	6
External Output (OT)	4	5	7
External Enquiry (QT)	3	4	6
External Interface File (EI)	5	7	10
Internal Logical File (FT)	7	10	15

#### 4.4 DETERMINE THE VALUE ADJUSTMENT FACTOR

<sup>4</sup> In ITSD, the alias of the 5 function types are as follows.

FT : Internal Logical File  
 EI : External Interface File  
 IT : External Input  
 OT : External Output  
 QT : External Inquiry.

There are in addition 14 general system attributes which account for the overall influences that will affect the size and complexity of the system to be provided to the users. These includes:-

- |                                 |                         |
|---------------------------------|-------------------------|
| (a) Data Communication;         | (h) On-line Update;     |
| (b) Distributed Processing;     | (i) Complex Processing; |
| (c) Performance;                | (j) Reusable Code;      |
| (d) Heavily Used Configuration; | (k) Installation Ease;  |
| (e) Transaction Rate;           | (l) Operational Ease;   |
| (f) On-line Data Entry;         | (m) Multiple Sites; and |
| (g) End-User Efficiency;        | (n) Ease of Change.     |

Each of the general system attribute will be assigned a value from 0 to 5 to show its degree of influence. The values of the degree of influence represent:

- 0 = Not present, or no influence when present
- 1 = Insignificant influence
- 2 = Moderate influence
- 3 = Average influence
- 4 = Significant influence
- 5 = Strong influence at all development stages

The adjustment factor is then calculated as a percentage of the sum of the degree of influence from standard solution (values 65) and the total degree of influence of the system. The factor will range from 0.65 to 1.35.

#### 4.5 CALCULATE THE FINAL ADJUSTED FUNCTION POINT

After determining the unadjusted function points and adjustment factor, the adjusted function points (i.e. final function points), which represent the size of an system, can be obtained by multiplying the two figures.

## **5. FPA FOR MAINTENANCE AND ENHANCEMENT**

Maintenance and enhancement after system live may change the functions of a system and therefore may modify its original function point count. There are rulings that apply to the addition, modification and deletion of function in order to derive the function point count that is in consistent with the count obtained during system development.

