

**Office of the  
Government Chief Information Officer**

**AN INTRODUCTION  
TO  
FUNCTION POINT ANALYSIS**

**A Technique for Measuring Software Size**

**[G19a]**

**Version : 2.6**

**Oct 2009**

© The Government of the Hong Kong Special Administrative Region  
of the People's Republic of China

The contents of this document remain the property of and may not be reproduced in whole  
or in part without the express permission of the Government of the HKSAR

<b>Amendment History</b>				
Change Number	Revision Description	Pages Affected	Revision Number	Date
1	Layout was revised according to Document Style Manual [S14] version 2.0	All	2.0	Feb 1999
2	Section 6 and 7.1 added to elaborate basic concepts of FPA	6-1, 7-1	2.0	Feb 1999
3	Section 7.3 and sub-section headings in section 7 was refined for clarification.	7-1, 7-2	2.0	Feb 1999
4	The document was reviewed and it was confirmed that the document was still valid. The expiry date was thus extended for another 3 years. The document was also revised according to the new requirements of the Document Style Manual S14.	All	2.1	Jun 2002
5	Remove the banner “Standards & Methods Document”	Front Cover Page	2.2	Jun 2003
6	Remove the Distribution and Release Page	Distribution and Release Page	2.2	Jun 2003
7	Remove the page number of Amendment History and Table of Content	Amendment History and Table of Contents	2.2	Jun 2003
8	Several sections were updated according to the latest reference documents.	3-1, 6-1, 7-2, 7-3	2.2	Jun 2003
9	Section 9 was updated due to change of roles and responsibilities of support services	9-1	2.2	Jun 2003

<b>Amendment History</b>				
<b>Change Number</b>	<b>Revision Description</b>	<b>Pages Affected</b>	<b>Revision Number</b>	<b>Date</b>
10	Merging of ITSD into the Commerce, Industry and Technology Bureau on 1 July 2004.		2.3	Jul 2004
10.1	Replace 'Information Technology Services Department (ITSD)' by 'The Office of the Government Chief Information Officer (OGCIO)'.	Whole document		
10.2	Replace logo of ITSD by OGCIO.	Front Cover Page		
11	Review the update-ness		2.4	Sep 2005
12	Review the update-ness		2.5	Aug 2008
13	Review the update-ness		2.6	Sep 2009

---

**TABLE OF CONTENTS**

<b>1. PURPOSE.....</b>	<b>1-1</b>
<b>2. SCOPE .....</b>	<b>2-1</b>
<b>3. REFERENCES.....</b>	<b>3-1</b>
3.1 Standards.....	3-1
3.2 Other References.....	3-1
<b>4. DEFINITIONS AND CONVENTIONS.....</b>	<b>4-1</b>
4.1 DEFINITIONS .....	4-1
4.2 CONVENTIONS .....	4-1
<b>5. PURPOSE OF FPA.....</b>	<b>5-1</b>
<b>6. COMPONENTS OF FUNCTION POINT ANALYSIS .....</b>	<b>6-1</b>
<b>7. STEPS OF FUNCTION POINT ANALYSIS.....</b>	<b>7-1</b>
7.1 DETERMINE THE TYPE OF FUNCTION POINT COUNT .....	7-1
7.2 IDENTIFY THE APPLICATION BOUNDARY .....	7-1
7.3 DETERMINE THE UNADJUSTED FUNCTION POINT .....	7-1
7.4 DETERMINE THE VALUE ADJUSTMENT FACTOR .....	7-3
7.5 CALCULATE THE FINAL ADJUSTED FUNCTION POINT.....	7-3
<b>8. FPA FOR MAINTENANCE AND ENHANCEMENT .....</b>	<b>8-1</b>
<b>9. APPLICATION OF FPA IN OGCIO .....</b>	<b>9-1</b>

## 1. PURPOSE

This paper is aimed to introduce the Function Point Analysis (FPA), a technique for measuring size of a system from a user's point of view.

## **2. SCOPE**

This document gives a concise picture of the important points of FPA (Function Point Analysis).

In brief, this document firstly introduces FPA and then discusses the major steps in counting Function Point. Application of FPA in OGCIO is also briefly described.

### 3. REFERENCES

#### 3.1 STANDARDS

None.

#### 3.2 OTHER REFERENCES

Counting Practices Manual, release 4.1.1, International Function Point User Group, Apr. 2000.

Resources Estimation Guide [G19], OGCIO.

## **4. DEFINITIONS AND CONVENTIONS**

### **4.1 DEFINITIONS**

None.

### **4.2 CONVENTIONS**

None.

## 5. 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.

## 6. 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 *low*, *average* or *high* 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 system characteristics* 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 system characteristics. 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.

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

## 7. 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.

### 7.1 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.

### 7.2 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 as to set up the scope for the related functions to be identified.

### 7.3 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) is a user identifiable group of logically related data or control information maintained within the boundary. The primary intent of an ILF is to hold data maintained through one or more elementary processes of the application being counted.
- (b) An External Interface File (EIF) is a user identifiable group of logically related data or control information referenced by the application, but maintained within the boundary of another application. The primary intent of an EIF is to hold data transferred through one or more elementary processes within the boundary of the application counted.
- (c) An External Input (EI) processes data or control information that comes from outside the boundary of the application being counted. The primary intent of an EI is to maintain one or more ILF and/or to alter the behaviour of the system.
- (d) The External Output (EO) sends data or control information outside the boundary of the application being counted. The primary intent of an EO is to present information to user through processing logic other than or in addition to the retrieval of data or control information. The processing logic must contain at least one mathematical formula or calculation, or create derived data. An external output may also maintain one or more ILFs and/or alter the behaviour of the system.
- (e) An External Inquiry (EQ) sends data or control information outside the application boundary. The primary intent of an external inquiry is to present information to the user through the retrieval of data or control information. The processing logic contains no derived data. No ILF is maintained during the processing, nor is the behaviour of the system altered.

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

- (i) EI, EO and EQ are given complexity ratings depending on the number of file type referenced (FTR) and data element types (DET) maintained/referenced; and
- (ii) EIF and ILF 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 a 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		
	Low	Average	High
External Input (EI)	3	4	6
External Output (EO)	4	5	7
External Inquiry (EQ)	3	4	6

External Interface File (EIF)	5	7	10
Internal Logical File (ILF)	7	10	15

#### 7.4 DETERMINE THE VALUE ADJUSTMENT FACTOR

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

- |                                 |                         |
|---------------------------------|-------------------------|
| (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 characteristics 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.

#### 7.5 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 a system, can be obtained by multiplying the two figures.

## **8. 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 line with the count obtained during system development.

## 9. APPLICATION OF FPA IN OGCIO

In March 1988, the FPA technique was announced as the standard methodology in OGCIO with twofold usage:

- (a) To measure the size of system as the original purpose of Albrecht's FPA; and
- (b) To estimate resources required to develop a system.

All resource estimates for systems development, enhancement and on-going support are calculated based on the function points derived from FPA technique. After obtaining the adjusted function points, resource estimation can be made by multiplying the function points with the appropriate work effort per function point as defined in the OGCIO Estimation Model.

Hence it is of good FPA that estimation for project can be reliable and measurable. The standard had to be adhered if it is required to have a fair comparability between systems and projects.