Function Point Analysis for Software Enhancement

Guidelines

Version 1.0

www.nesma.org

Professional guide of the Netherlands Software Metrics Users Association

Table of Contents

NESMA

The Netherlands Software Metrics Users Association, NESMA was founded in May 1989. Its main goals are to:

- bring together individuals and organisations to exchange knowledge and experience in the development and application of software metrics;
- promote accountability in the use of software metrics;
- support the formulation and adoption of standards in software metrics; and
- encourage the development and application of software metrics.

NESMA is working towards these goals through:

- the activities of its study groups and working groups;
- research undertaken by its members;
- organising presentations, training, symposia, and the like;
- forming recommendations on the use of software metrics;
- collecting and publishing literature on software metrics;
- collaborating with organisations with similar interests; and
- liasing and collaborating with the other software metrics user groups and associations world wide, including IFPUG (USA), ASMA (Australia) and FESMA (Europe).

NESMA has developed and maintains its own counting practices, embodying the same concepts, methodology and counting rules as those published by the International Function Point Users Group (IFPUG). In addition, NESMA has developed many guidelines for use by function point analysts to provide practical guidance in the application and use of function point analysis.

You can learn more about NESMA by visiting the multilingual NESMA web site www.nesma.org (Dutch, English, Japanese). For more information contact NESMA at office@nesma.org.

FOREWORD

Benefits of Function Point Analysis

The management of information systems through all stages of their development and operational lifecycles becomes more difficult as systems grow more complex. Managers need to closely monitor and measure the functional capabilities of systems in order to better understand:

- the functions a particular system offers the user;
- the organisation structures and support mechanisms necessary to deliver these functions to the user; and
- the organisation structures and support mechanisms necessary to manage operational systems and enhance their functional capabilities to meet changing user requirements.

Function point analysis can be used to:

- describe the scope of a system and measure its functional size independent of the technologies the system employs;
- derive productivity and process performance metrics, estimate resource requirements and assist in project management;
- identify factors that influence productivity and process performance and so provide a basis for improving the development and maintenance processes; and
- determine the extent and scope of system enhancement and assist in managing system changes.

FPA for Software Enhancement

In 1991 NESMA formed the working group 'FPA for Enhancement and Maintenance' to develop and publish guidelines for the application of function point analysis to software enhancement and maintenance. After several years of research and development, the working group published these guidelines in 1998. The English translation was published in 2001.

The members of the working group are:

J. T. Engelhart

P. L. Langbroek

A. J. E. Dekkers

H. J. G. Peters

P. H. J. Reijnders

The following people have also contributed to these guidelines:

F. X. Granneman

P. J. M. Hickendorff

J. W. ter Veld

Foreword to the English translation

There is broad international interest in the application of function point analysis to software enhancement although, to date, there have been few publications on the subject. People from all over the world have requested NESMA to translate this guide "Function Point Analysis for Software Enhancement" into English. This English version is an accurate translation of the Dutch guide and uses the terminology defined in the NESMA and IFPUG Counting Practices Manuals.

The translation process has literally been a worldwide project. The NESMA Board thanks all people who participated in the translation of these guidelines. Firstly NESMA thanks Oliver Hague (past President of ASMA, Australia), an expert in FPA and software metrics. Oliver wrote the first English language draft from a verbatim translation of the Dutch text and edited the revised text to produce this published version.

NESMA also thanks Adri Timp (Netherlands, chair of the NESMA Counting Practices Committee and vice chair of the IFPUG Counting Practices Committee) and David Garmus (USA, IFPUG President 2000-2001). Adri Timp carried out a meticulous comparison of the translated English text and the original Dutch text, adding many improvements and clarifications. David Garmus reviewed the translation and provided valuable feedback. Finally, Oliver Hague and Adri Timp formatted the translated guide and made final adjustments to the text.

By offering this guide to the international functional software measurement community, NESMA expects to stimulate further analysis of the measurement of software enhancement. NESMA hopes, that these guidelines will be applied and tried out worldwide. (Please note the disclaimer with respect to this method in section 1.6).

NESMA would like to hear about your experiences with this method and welcomes your suggestions. The practical experiences of users may lead to the publication of revised versions of this guide. Please forward your suggestions and comments to NESMA at office@nesma.org.

Netherlands, August 2001

The NESMA Board

1 INTRODUCTION

1.1 Purpose of this Document

Function point analysis is established internationally as a method for determining the scope and functional size of software from an assessment of the user requirements. The IFPUG "Function Point Counting Practices Manual" [IFPUG, 1] and the NESMA FPA counting practices manual “Definitions and Counting Guidelines for the Application of Function Point Analysis” [NESMA, 1] both follow the “Albrecht” method and describe how to apply the method to implemented systems, software development and software enhancement. Application of the method to software enhancement is not well developed; other priorities have prevented a more considered treatment of this aspect of its application in the past.

Function point analysis has been applied extensively to the development of new software. Its use in this respect is well established and is supported by a wealth of research and practical experience. It is now appropriate to explore in greater depth the application of FPA to software enhancement and maintenance. Users of software metrics need to know whether FPA can be successfully applied to software enhancement and, if so, in what way and within what constraints. Consideration of these issues led NESMA to form the working group on “FPA for Enhancement and Maintenance”.

The first results of this working group were published in August 1993 and led subsequently to publication of the method described in these guidelines. A number of organisations used the method and their experiences resulted in further refinement of the guidelines and the inclusion of examples to illustrate the application of the method.

1.2 Objectives of the Guidelines

These guidelines are intended for anyone with an interest in the management of enhancements to an information system.

The guidelines describe an objective and replicable method for assessing the scope and size of an enhancement project. The method is objective in that the results obtained are independent of the person applying the method; the result obtained is *bona fide* in that two different people using the same guidelines obtain the same result. The method is replicable in that a particular outcome can be determined *a priori,* and the same outcome can be produced on the second and subsequent applications of the method.

1.3 Starting Point

These guidelines build on the FPA methodology described in the IFPUG publication "Function Point Counting Practices Manual" [IFPUG, 1], and the NESMA publication “Definitions and Counting Guidelines for the Application of Function Point Analysis” [NESMA, 1]. Throughout these guidelines the method described in [IFPUG, 1] and [NESMA, 1] is referred to as "*the standard FPA method*".

1.4 Intended Audience

These guidelines are intended for anyone who performs function point analysis and wants to measure the size of enhancement projects more precisely. It is assumed that the reader is familiar with the standard FPA method (see section 1.3).

1.5 Scope of the Research

NESMA considered the application of FPA to software enhancement from the perspective of the standard function point analysis method. The result of this work, embodied in these guidelines, is a method applicable to software enhancement and testing that is strongly related to the standard FPA method. The term Enhancement Function Point Analysis (EFPA) is used to differentiate the method from the standard function point analysis method.

1.6 Disclaimer

The method described in this guide has been tried in practice. However, NESMA does not claim that the method in its current form has been validated scientifically. Additional research and practical use is necessary to demonstrate the validity of the method.

By offering this guide to the international functional software measurement community, NESMA wants to advance the application of function point analysis to enhancement projects and to broaden the understanding of measurement applied to software enhancement. NESMA is not responsible for any use of this method or for the results obtained from its application.

Comments and suggestions for further improvement of this method may be sent to office@nesma.org.

1.7 Future versions

When changes and supplements to this guide prove necessary in the future, a new version will be released. The latest version may be downloaded from the NESMA web site www.nesma.org. Please forward your comments to office@nesma.org. This will help NESMA to eventually improve and refine this method.

1.8 Organisation of the Guide

Following this introduction, Chapter 2 discusses conditions limiting the application of the enhancement FPA methodology (EFPA). Chapter 3 describes the enhancement FPA methodology. Chapter 4 discusses EFPA in relation to testing. Chapter 5 describes the use of the enhancement size measure for budgeting purposes. Application of the enhancement FPA methodology is illustrated by examples in Chapter 6. Chapter 7 is a glossary of the more important concepts and abbreviations and Chapter 8 contains a concise Bibliography.

2 GENERAL CONSIDERATIONS

2.1 Limiting Conditions

This chapter discusses limits to the use of the methodology set out in Chapter 3.

Work carried out on implemented systems, excluding new development, is commonly termed maintenance. In order to understand this class of work, it is necessary to clearly distinguish between:

- activities that enable a system to deliver the same functional capabilities in a changed technological environment;
- activities that maintain a system in its existing operational state; and
- activities that extend or change the functional capabilities of a system.

In this document, these three types of activities are termed conversion, maintenance, and enhancement respectively.

Conversion

Conversion encompasses a broad range of activities including, for example, restructuring data definitions to accommodate revised technical requirements, translating source code to a new or updated language, transferring a system to a totally different operating environment or changing the storage of physical data to accommodate the introduction of a new database management system. These guidelines do not discuss the application of FPA to conversion.

When conversion is undertaken, it is often necessary to develop systems to perform specific conversion functions. If new conversion enabling functions are developed, their functional scope and size may be determined using the standard FPA method.

Maintenance

Maintenance includes all those activities necessary to sustain the operation of a system without modifying its functionality. Maintenance work may extend to the environment in which the system operates and may include a multitude of activities, sometimes concerning technical infrastructure, that are not related directly to the functionality of the system.

The NESMA working group concluded that analysis of maintenance could not be divorced from consideration of the specific technologies used in the operating environment. Consequently the group decided to focus its research exclusively on the application of function point analysis to software enhancement and testing.

A separate NESMA working group was formed to examine “System Operation and Maintenance”. In 1998 this working group published the manual "Budgeting the Operational Costs of Information Systems" [NESMA, 2]. This manual is available in Dutch only.

Enhancement

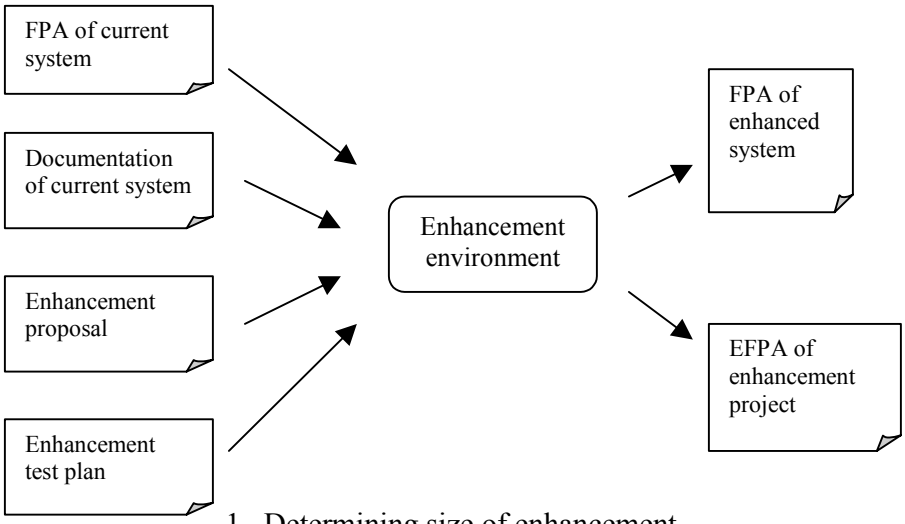
In the context of these guidelines, ‘enhancement’ means changes to the functionality of an information system. Function point analysis expresses the functionality of an information system in terms of transactional functions and data functions. The working group directed its research towards the application of the FPA method to enhancement. Consequently, these guidelines are applicable to the enhancement of existing operational information systems only.

The working group concluded that enhancements resulting only in the addition of new functions, with no changes being made to existing functions, should be treated as new development. This situation can arise when a system is expanded to support the requirements of organisational processes not currently within the scope of a system and often results in the addition of a new component. Development of this type can be analysed using the standard FPA method.

2.2 Prerequisites

The following are required to carry out a function point analysis of an enhancement project in accordance with these guidelines:

- the function point analysis details of, at least, the part of the current system affected by the enhancement (a current FPA count);
- documentation describing the affected part of the current system; and
- an enhancement proposal that describes the modifications to be made and a test plan specifying the tests to be carried out.



1. Determining size of enhancement

This information is required to determine the scope and size of the enhancement project; without it, the enhancement function point analysis cannot be carried out.

FPA details of the affected parts of the existing system

The method presented in these guidelines takes as its starting point the function point analysis results obtained for the existing system. As a minimum, the analysis of the current system must identify the individual functions supported by the system and:

- the size of each data function (ILF or EIF) and the DETs it uses;
- the size of each transactional function (EI, EO, EQ) and the DETs and File Types Referenced it uses (the ILFs/EIFs read and/or maintained by the transaction).

If this information is not available, an analysis of the current system must be carried out (or at least of the part of the system affected by the enhancement) in order to establish the foundation for analysis of the proposed enhancement.

Documentation of the current system

Proposed changes need to be assessed and current functions need to be compared with proposed functions in order to detect changes in business rules as well as changes to DETs, FTRs (ILFs and EIFs) and the user interface. Good functional system documentation is crucial to identify the scope of the enhancement.

The analyst needs to consider:

- the extent of changes to individual transactional and data functions; and
- the wider implications of the specified changes on other transactional and data functions.

An enhancement proposal

The enhancement proposal, together with the documentation of the current system, must specify the enhancements to be carried out in sufficient detail to enable the effects on each impacted transactional and data function to be assessed.

Sufficient detail is required to remove all ambiguity concerning the scope of the enhancement, the data and transactional functions affected and the extent of the impact on each function. If the enhancement proposal does not provide this level of detail, it must be further refined.

A test plan

The test plan must specify the transactional and data functions to be tested and define the scope of the tests to be carried out. The smallest unit of testing is often a system

component or a sub-system, regardless of the extent of the enhancement. A test plan should identify both the components and the functions to be tested.

2.3 Summary of the Enhancement FPA Methodology

There are eight steps in the function point analysis of an enhancement project:

1. Identify the transactional and data functions within the scope of the enhancement project.
2. Determine the enhancement size of the transactional and data functions to be added.
3. Determine the enhancement size of the transactional and data functions to be deleted.
4. Determine the enhancement size of the data functions to be changed.
5. Determine the enhancement size of the transactional functions to be changed.
6. Calculate the unadjusted function point size of the enhancement project.
7. Apply the value adjustment factor and calculate the adjusted function point size of the enhancement *project*.
8. Calculate the adjusted function point size of the *system* after enhancement.

Characteristic of this method is the treatment of deleted and changed functions (step 3, 4 and 5). For each data and transactional function the degree of change is taken into account. This degree is expressed in an impact factor. The enhancement function point size of a deleted or a changed function is determined by multiplying the number of function points of the function (according to the standard FPA method) by the impact factor.

The analysis proceeds by first determining the size of each function added, changed or deleted using the standard FPA method. The result is the unadjusted size of the function (UFP_{BASE}). UFP denotes the unadjusted function point size measured in accordance with the standard FPA methodology.

Transactional and data functions can be added, changed or deleted and as such may have impacts beyond those apparent from the enhancement proposal. For example, the change of a logical file or transaction may impact other transactions or logical files. This is why every impacted function needs to be carefully analysed to identify the extent of the impact of enhancement on the function. All data and transactional functions impacted by the enhancement proposal must be identified. The impact factor (I) reflects the degree of change of each identified (data and transactional) function.

The enhancement size of each affected transactional and data function is calculated by multiplying its base size (UFP_{BASE}) by its impact factor (I). The enhancement size is measured in "Enhancement Function Points" (EFP), not standard function points, which is a different measure. It is imperative to maintain the distinction between the standard function point unit used to express the size of software (FP) and the unit used to express the size of an enhancement (EFP).

3 METHODOLOGY

As outlined in the previous sections, six steps are carried out to determine the scope and size (expressed in unadjusted enhancement function points) of an enhancement project.

1. Identify the transactional and data functions within the scope of the enhancement project.
2. Determine the enhancement size of the transactional and data functions to be added.
3. Determine the enhancement size of the transactional and data functions to be deleted.
4. Determine the enhancement size of the data functions to be changed.
5. Determine the enhancement size of the transactional functions to be changed.
6. Calculate the unadjusted function point size of the enhancement project.

Two additional steps are required to calculate the *adjusted* function point size of the project and of the system after enhancement:

7. Apply the value adjustment factor and calculate the adjusted function point size of the enhancement *project*.
8. Calculate the adjusted function point size of the *system* after enhancement.

3.1 Identify the functions within the scope of the enhancement project

The enhancement proposal, the functional documentation of the current system and the function point analysis of the existing system are used to identify the transactional and data functions within the scope of the enhancement project. A function point analysis of the existing system is an essential prerequisite because all existing functions that are affected either directly or indirectly by the enhancement contribute to the function point size of the enhancement. If, for any reason, a function point analysis of the existing system is not available, one must be undertaken to identify, as a minimum, the functions affected by the enhancement.

The size of the existing system, or that part impacted by the enhancement project, is expressed in standard (unadjusted) function points, ΣUFP_{BASE}.

3.2 Determine the enhancement size of the transactional and data functions added

The enhancement proposal should specify the transactional and data functions to be added to the application. From the proposal it should be possible to calculate the size of the functions added by applying the standard FPA methodology. The total is expressed as UFP_{ADDED}. The impact factor for added functions is always 1. Hence, the number of unadjusted enhancement function points for a single added function is determined as follows:

$$UEFP_{ADDED} = UFP_{ADDED}$$

This means, for example, that 3 function points added will result in 3 enhancement function points.

See also example 5 in Chapter 6.

3.3 Determine the enhancement size of the transactional and data functions deleted

The (data and transactional) functions that will be deleted from the existing system are identified from the enhancement proposal and the number of function points they represent is determined. The size of the deleted functions is expressed as $UFP_{DELETED}$.

For deleted functions an impact factor of 0.4 is used. The number of unadjusted enhancement function points for a single deleted function is determined as follows:

$$UEFP_{DELETED} = UFP_{DELETED} \times 0.4$$

This means, for example, that 6 (unadjusted) function points deleted will result in 2.4 (unadjusted) enhancement function points.

3.4 Determine the enhancement size of the data functions changed

The data functions that change are identified and the size of each data function *after the change* is determined.

A data function can be either an internal logical file (ILF) or an external interface file (EIF). Each type of data function is assessed to identify:

- data functions that change internally: DETs added, deleted or changed; and
- data functions that change type but do not change internally (that is, an EIF is changed into a ILF or *vice versa*).

The analysis determines the data functions that will change and how many function points each data function represents *after the change*. The standard FPA guidelines (as described in [IFPUG, 1] and [NESMA, 1]) are used to determine the size of the data function after change. The (unadjusted) function point size of the changed data function is $UFP_{CHANGED}$.

For data functions that *change internally* an impact factor is calculated from the percentage of DETs changed. The percentage change is defined as the ratio of DETs changed divided by the original number of DETs (see also example 10 in Chapter 6).

$$\text{Percentage change} = \frac{\text{Number of DETs added/changed/deleted} \times 100}{\text{Number of DETs in original data function}}$$

The impact factor ($I_{CHANGED}$) is taken from Table 1 using the percentage change in the number of DETs.

Percentage DETs	≤ 33%	≤ 67%	≤ 100%	> 100%
Impact factor	0.25	0.50	0.75	1.00

Table 1 - Data Function Impact Factors

If a data function *changes type*, (for example, an external interface file becomes an internal logical file), a value of 0.4 is used for the impact factor.

Changes of type need to be assessed also to identify changes in numbers of DETs. If the number of DETs changes as well as the type, the impact factor due to the change in the number of DETs must be determined. The value of the impact factor due to the change in type is compared with that due to the change in the number of DETs and the higher value is used in the calculation of the enhancement function point size (see example 3 in Chapter 6).

The number of enhancement function points for a single changed data function is determined as follows:

$$UEFP_{CHANGED} = UFP_{CHANGED} \times I_{CHANGED}$$

The number of enhancement function points arising from a change in data functions therefore depends on the extent of the change in the data function. See also examples 1, 2 and 3 in Chapter 6.

If an EIF or an ILF is *divided* into two (or more) data functions, one deleted data function and two (or more) added data functions are counted. See also example 4 in Chapter 6.

If an EIF and an ILF are *combined*, two deleted data functions and one added data function are counted.

3.5 Determine the enhancement size of the transactional functions changed

The transactional functions that change are identified and the size of each transaction *after the change* is determined.

A transactional function is considered changed if it is altered in some way but retains the same name and purpose *after enhancement* as *before enhancement*. The standard FPA guidelines (as described in [IFPUG, 1] and [NESMA, 1]) are used to determine the size of the transaction after change. The number of (unadjusted) function points after the change is expressed as $UFP_{CHANGED}$.

A transactional function may be affected by changes to data functions. All transactional functions specified in the enhancement proposal and those affected by changes to data functions are included in the scope of the analysis. This means that a transaction is counted when one of the following conditions is satisfied:

- the transaction is identified in the enhancement proposal; or;
- the transaction undergoes a function change as a consequence of other changes defined in the enhancement proposal.

Generally, a transaction must be counted if the user can identify that the transaction has changed. This means that at least one of the following criteria is met:

- a transaction is affected by a DET that is added, changed or deleted;
- a transaction is affected by an FTR **(**ILF or EIF) that is added, changed or deleted;
- the user interface is changed (for example, the composition of a screen or a report);
- the business logic supporting a transaction is changed (for example, edit rules or calculations performed on the transaction data);
- a cosmetic change visible in the user interface is made, for example:
    - a change to a total in a report (see also example 9 in Chapter 6);
    - a heading is replaced or changed in a report or on a screen.

A change to the name of a DET is not regarded as a change in a transaction (see example 8 in Chapter 6). The nature of the DET does not change if the name only is changed.

There are four steps to calculating the enhancement function point size of a change to a transaction:

1. Identify the DETs and FTRs used by the transaction.
2. Determine the percentages of DETs and FTRs changed as a result of the enhancement.
3. Determine the impact factor for the transaction.
4. Calculate the number of enhancement function points.

1. Identify the DETs and FTRs used by the transaction

The enhancement function point size of a changed transactional function is calculated from the function point size of the function *after the change* and the change impact factor. The impact factor is determined by the percentage changes in the numbers of DETs and FTRs used by the transaction. Examples 6 and 7 in Chapter 6 illustrate how a changed transaction is assessed.

If the change is cosmetic only, the number of changed DETs and FTRs is nil. The impact of such a change is considered minimal and the value of the impact factor

(0.25) reflects a relatively low impact. However, the change should be included in the scope of the enhancement project.

2. Determine the percentage of DETs and FTRs changed as a result of the enhancement.

The impact factor is determined by the percentage changes to the numbers of DETs and FTRs used by the transaction compared with the original numbers of DETs and FTRs (see example 11 in Chapter 6).

$$\text{Percentage DETs} = \frac{\text{Number of DETs added/changed/deleted} \times 100}{\text{Number of DETs in original transaction}}$$

$$\text{Percentage FTRs} = \frac{\text{Number of FTRs added/changed/deleted} \times 100}{\text{Number of FTRs in original transaction.}}$$

Changes in excess of 100% are possible when DETs and FTRs are added to a transaction.

3. Determine the impact factor for the transaction.

The impact factor ($I_{CHANGED}$) for a transaction is determined from the percentage changes in the numbers of DETs and FTRs from Table 2:

Change:	**Percentage DETs**		
Percentage FTRs	≤ 67%	≤ 100%	> 100%
≤ 33%	0.25	0.50	0.75
≤ 67%	0.50	0.75	1.00
≤ 100%	0.75	1.00	1.25
> 100%	1.00	1.25	1.50

Table 2. Transactional Function Impact Factors

If the impact factor is 1.00 or greater, it may be more appropriate to treat the enhancement as the deletion of the existing function and the addition of a new replacement function.

4. Calculate the enhancement function point size of a transaction

The unadjusted enhancement function point size of a single transactional function is calculated as follows (see also example 12 in Chapter 6):

$$UEFP_{CHANGED} = UFP_{CHANGED} \times I_{CHANGED}$$

3.6 Calculate the size of the enhancement project.

The (unadjusted) size of the enhancement project is the sum of the number of enhancement function points for all the affected transactional and data functions.

$$UEFP_{PROJECT} = \Sigma UEFP_{ADDED} + \Sigma UEFP_{CHANGED} + \Sigma UEFP_{DELETED}$$

Note: If functions are needed for conversion functionality, their size should be added. They are considered as new functions, so $UEFP_{CONV} = UFP_{CONV}$

3.7. Apply the Value Adjustment Factor and calculate the adjusted function point size of the enhancement *project*

The Value Adjustment Factor (VAF) must be applied to the enhancement project size in organisations that adhere to the use of this factor. The fourteen system characteristics must be reassessed for the enhanced system.

The VAF for the system *after enhancement* (VAF_{AFTER}) is applied to the sizes of the added and changed functions and the VAF for the system *before enhancement* (VAF_{BEFORE}) is applied to the size of deleted functions.

$$\begin{aligned} EFP_{PROJECT} = {} & \Sigma UEFP_{ADDED} \text{ x } VAF_{AFTER} \\ & + \Sigma UEFP_{CHANGED} \text{ x } VAF_{AFTER} \\ & + \Sigma UEFP_{DELETED} \text{ x } VAF_{BEFORE} \end{aligned}$$

Note: If functions are added for conversion, their size should be added as an extra component on the right side of the equation: $\Sigma UEFP_{CONV} \text{ x } VAF_{AFTER}$

3.8. Calculate the size of the *system* after enhancement

The functional size of a system may change as a result of the enhancement. The size after enhancement can be calculated by analysing the whole application anew or by taking account of the changes from the original FPA analysis. Steps to take are:

1. Calculate the unadjusted function point size of the application prior to the change (UFP_{BEFORE}) using the standard FPA methodology as described in [IFPUG, 1] and [NESMA, 1].
2. Identify the transactional and data functions deleted from the existing application and determine their function point size ($UFP_{DELETED}$).
3. Determine the transactional and data functions changed. Calculate the number of function points these represent before and after the enhancement (UFP_{AFTER} and UFP_{BEFORE}).
4. Determine the transactional and data functions added to the system and calculate how many function points these represent (UFP_{ADDED}).
5. Calculate the value adjustment factor for the system after enhancement (VAF_{AFTER}).
6. Calculate the size of the system after enhancement (UFP_{AFTER}).

The size of the system in unadjusted function points after enhancement is:

$$UFP_{AFTER} = UFP_{BEFORE} + (\Sigma UFP_{ADDED} + \Sigma UFP_{AFTER\ CHANGE}) - (\Sigma UFP_{BEFORE\ CHANGE} + \Sigma UFP_{DELETED})$$

The adjusted function point size of the *system* after enhancement is:

$$FP_{AFTER} = UFP_{AFTER} \times VAF_{AFTER}$$

Note: If functions are added for conversion, their size should *not* be added. They do not contribute to the size of the implemented system after enhancement.

4 TESTING

The range of transactional and data functions that have to be tested can be much greater than the number of transactional and data functions within the scope of an enhancement.

The size of the functions to be tested is measured in *test function points* (TFPs). The number of TFPs takes into account all functions subject to testing. The size of each function tested is measured by using the standard FPA methodology and is not moderated by impact factors for each function. The functions measured are those in the system to be tested (i.e., after the enhancement), so no account is taken of whether a function was added, changed, combined with other functions or otherwise impacted by the enhancement. Deleted functions are excluded from the count of test function point size (TFP).

In general, testing exercises discrete components of the system and encompasses unchanged functions as well as changed and added functions. Each function included in the scope of a test is measured *after enhancement*.

The number of TFPs is determined as follows.

Transactional functions:

- determine the FP size of each transaction directly involved in a test;
- calculate the total number of function points for all transactions involved in the test.

Data functions:

- determine the FP size of each data function directly involved in a test;
- calculate the total number of function points for all data functions involved in the test.

The total number of test function points is the sum of the transactional TFPs and data TFPs.

The size of each function is derived using the standard FPA counting practices for new software development: the number of function points (UFP_{TEST}) is the sum of the sizes of the transactional and data functions involved in the tests.

The test function point size is equal to the size of the tested functions, where 1 UFP results in 1 TFP. The total test function point size is:

$$TFP = \Sigma UFP_{TEST}$$

5 USING ANALYSIS RESULTS

The technique described in these guidelines is based on the standard FPA method. Correlations are postulated between the EFP size and enhancement effort and between the TFP size and testing effort. As with the standard FPA measure, there are likely to be wide variations above and below norms, especially for small enhancement projects.

Enhancement function point sizes and test function point sizes can be used to derive productivity metrics, for example, hours per EFP and hours per TFP. Values for hours per EFP and hours per TFP will, in general, differ from the hours per function point for new system developments measured using the standard FPA method.

The total enhancement effort, including testing, can be expressed as follows:

Total enhancement effort = (EFP size x hours per EFP) + (TFP size x hours per TFP)

6 EXAMPLES

Example 1

Situation
An internal logical file consisting of 1 RET and 37 data element types is to be expanded by adding 14 new data element types.

Question
Which impact factor should be used, and how many enhancement function points does the change generate?

Answer
The size of the internal logical file, after change, is 10 function points (ILF with average complexity). The change impact expressed as a percentage of DETs is 14/37 x 100% = 37.8% (rounded). From table 1, a change of 37.8% gives in an impact factor of 0.5. The enhancement function point size is 10 x 0.5 = 5 EFP.

Example 2

Situation
Information system A uses an external interface file maintained by information system B. A decision is made that, in future, maintenance of this function will be carried out by information system A (the structure of the data function does not change).

Question 1
Which impact factor should be used?

Answer
An external interface file (EIF) is imported into system A and converted to an internal logical file (FTR). An impact factor of 0.4 is used when an EIF is changed to an ILF or *vice versa*.

Question 2
How many enhancement function points result from the change to the data function?

Answer
Assuming the data function is a low complexity ILF, its size after the change is 7 function points. The impact factor is 0.4, therefore the change results in 7 x 0.4 = 2.8 EFP.

Example 3

Situation
Information system A uses an external interface file that is maintained by system B and contains 45 data element types. A decision is made that system A will maintain

the data function and as a consequence 25 data element types will be removed from the data function.

Question 1
Which impact factor should be used?

Answer
An impact factor of 0.4 is used when an EIF is changed to an ILF or *vice versa*. However in this case the affect of the structural change to the data function must also be taken into account. The percentage change is:

25/45 x 100% = 55.6% (rounded).

From Table 1, a change of 55.6% gives an impact factor of 0.5. This is greater than the impact factor for a change of type (from an EIF to an ILF), so the higher value of 0.5 is used.

Question 2
How many enhancement function points result from the change in the data function?

Answer
Assuming the file is a low complexity ILF, the size of the data function after change is 7 function points. The impact factor is 0.5 and so the change gives rise to 7 x 0.5 = 3.5 EFP.

Example 4

Situation
An external interface file is divided into two separate external interface files.

Question
How do we account for this change?

Answer
Count one deleted external interface file and two added external interface files. The +analysis must also take into account all the transactional functions affected by the change to the data function.

Example 5

Situation
A new external interface file is to be added to a system.

Question
How do we account for this change?

Answer
Count one added external interface file (impact factor = 1) and take into account the

transactions that use the data function. These transactions must be reassessed as part of the enhancement project.

Example 6

Situation
3 DETs in a data function used by a transaction are to be changed. Two of the changed DETs will be used by the transaction.

Question
How do we account for the impact of the change on the transaction?

Answer
One changed data function and two changed DETs are counted to determine the impact factor for the changed transaction.

Example 7

Situation
3 DETs in a data function (ILF or EIF) used by a transaction are to be changed. One of the of DETs will be used by the transaction.

Question
Do we need to account for this change?

Answer
The transaction must be included in the scope of the enhancement.

The data function is an FTR for the transaction and will result in a logic change in the process underlying the transactional function.

Example 8

Situation
The name of a DET is changed in a data function used by a transaction. The DET is used in the transaction.

Question
Do we need to account for this change?

Answer
The transactional and the data function are not counted.

Example 9

Situation
An external output prints a report that lists employee information. The heading of one of the columns is changed from "Name labourer" into "Name employee". There is no change in the DETs of the data function, so the change is cosmetic.

Question
Do we need to account for this change?

Answer
The function will change as a result of the cosmetic change. As no DETs are changed the impact factor = .25.

Example 10

Situation
Two DETs are added to a data function, 1 DET is deleted and 3 DETs are changed.

Question
How many DETs will be counted in this change?

Answer
The number of DETs counted is 2 + 1 + 3 = 6 DETs.

Example 11

Situation
A report (including a calculated total) displaying 16 DETs will have 3 new DETs added to it, 3 DETs will be changed and 2 DETs will be deleted. The number of affected DETs is 3 + 3 + 2 = 8. The output transaction uses 2 FTRs. The new DETs belong to a new (to be added) FTR. Only one FTR is affected by the changed and deleted DETs.

Question 1
How do we determine the percentage changes in the DETs and FTRs?

Answer
Measure the transactional function DETs changed relative to the original number of DETs for the transaction (16). The calculation is therefore 8/16 x 100 = 50%.

Measure the transaction FTRs changed relative to the original number of FTRs (2). The calculation is therefore 2/2 x 100 = 100%.

Question 2
What is the size of the change to the transaction?

Answer

After enhancement the number of DETs is 17 and there are 3 FTRs. The complexity of the function remains average and its function point size does not change (5 function points). The impact factor is taken from Table 2. The percentage changes in DETs (50% - first column) and FTRs (100% - third row) give an impact factor of 0.75. The enhancement size is:

$$EFP_{CHANGED} = 5 \times 0.75 = 3.8 \text{ EFP}.$$

7 GLOSSARY

Note: This paper assumes, that the reader is familiar with the standard FPA methodology, its concepts and terms. For more information, reference the IFPUG publication "Function Point Counting Practices Manual" [IFPUG, 1] or the NESMA publication "Definitions and Counting Guidelines for the Application of Function Point Analysis" [NESMA, 1].

Cosmetic change
A change only in the visible user interface presented by a transactional function, without any change in the processing logic underlying the transaction (independent of any descriptions of addition, change, or removal of DETs or FTRs).

Data function
An internal logical file (ILF) or an external interface file (EIF).

DET
Data element type.

EFP
Enhancement function point.

EIF
External interface file.

Enhancement
Enhancement is the work necessary to bring about a change in an operational information system or in the structure of a data store of an operational information system.

The definition of enhancement is derived from that of Vollmar: "Enhancement is effecting a change in an information system or in the structure of the data store". The definition is independent of the life cycle in which the information system may exist and allows for the addition of new functions, the removal of existing functions and changes to existing functions to be included.

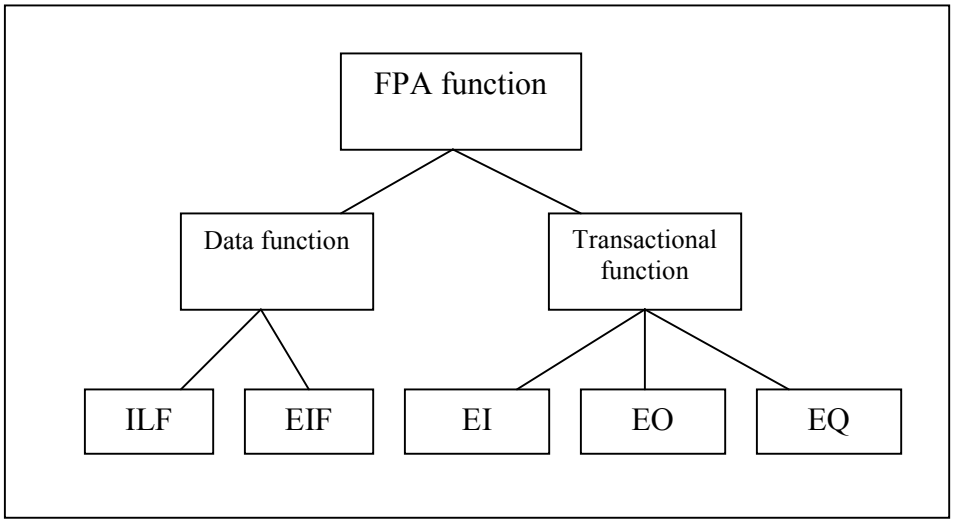
Enhancement function point
A unit of measurement of the size of an enhancement.

Enhancement proposal
A formal request for enhancement of an operational information system. The proposal must be sufficiently comprehensive to enable the scope and impact of the enhancement to be determined.

FTR (File Type Referenced)

- A internal logical file (ILF) read or maintained by a transactional function.
- An external interface file (EIF) read by an transactional function.

Function
An external input (EI), external output (EO), external inquiry (EQ), internal logical file (ILF) or external interface file (EIF) as defined in the standard FPA methodology (see for example [IFPUG, 1] or [NESMA, 1]). In function point analysis a function is either a *transactional function* or a *data function*.



ILF
Internal logical file.

Impact factor
A unit of measurement of the degree of change in a transactional function or data function. The value of the impact factor may vary according to the nature and extent of the change.

LF (Logical file)
A generic name for an internal logical file (ILF) and external interface file (EIF); synonymous with data function.

Maintenance
In the context of these guidelines, maintenance encompasses all the activities necessary to operate an automated information system and manage the associated technical, organisational and financial aspects. Maintenance involves performing the work necessary to ensure the continued operation of the system without altering the scope or structure of the system or its associated data stores (after Looijen).

Standard Function Point Count

A function point count performed using the standard FPA methodology as described in [IFPUG, 1] and [NESMA, 1].

Test function point

A unit of measure of the size of the functions subject to testing.

TFP

Test function point.

Transactional function

An external input, external output or external inquiry function.

UEFP

Unadjusted enhancement function point. UEFP multiplied by the Value Adjusted Factor (see [IFPUG, 1] or [NESMA, 1]) results in the EFP.

UFP

Unadjusted function points, calculated using the standard FPA methodology without taking into account the impact factor.

8 BIBLIOGRAPHY

[IFPUG, 1] **IFPUG Function Point Counting Practices Manual**,
release 4.1

This manual describes the standard FPA methodology and may be used together with the NESMA manual.
For more information, see the IFPUG web site www.ifpug.org.

Date of issue: January 1999
ISBN 0-963-1742-7-4

[NESMA, 1] **Definitions and counting guidelines for the application of function point analysis**.
A practical manual, version 2.0. (English)

This manual is also called NESMA Counting Practices Manual. It describes the standard FPA methodology, and many aspects related to the application of FPA. It may be used together with the IFPUG manual.

For more information, see the NESMA web site www.nesma.org (English).

Date of issue: November 1997
164 pages A4-format
ISBN 90-76258-02-3

[NESMA, 2] **Begrotingsmodel voor de exploitatielasten van informatiesystemen** (**Budgeting the Operational Costs of Information Systems**; not yet translated into English)

Date of issue: November 1998
Size: 117 pages A4-format

ISBN 90-76258-11-2

[NESMA, 3] **Function Point Analysis for Software Enhancement**

Date of issue: August 2001.
ISBN 90-76258-13-9