FUNCTION POINT ANALYSIS
FOR SOFTWARE ENHANCEMENT

Guidelines

Version 2.3

Professional guide of the Nesma

CONTENTS

1 Introduction 5
1.1 Purpose of this Document 5
1.2 Objectives of the Guidelines 5
1.3 Starting Point 5
1.4 Intended Audience 6
1.5 Scope of the professional guide 6
1.6 Remarks 6
1.7 Future versions 7
1.8 Organisation of the Guide 7
2 General considerations 8
2.1 Limiting Conditions 8
2.2 Prerequisites 10
2.3 Summary of the Enhancement FPA Methodology 12
3 Methodology 14
3.1 Identify the functions within the scope of the enhancement project 14
3.2 Determine which transactional and data functions will be added 14
3.3 Determine which transactional and data functions will be deleted 15
3.4 Determine which data functions will be modified and determine the impact factor 15
3.5 Determine which transactional functions will be modified and determine the impact factor 17
3.6 Calculate the size of the enhancement project 19
3.7 Calculate the size of the system after enhancement 20
3.8 High level Function Point Analysis for Software Enhancements 21
4 Testing in enhancement projects 22
5 Budgeting enhancement 23

6 Examples 24
6.1 Example 1 - Expanding an ILF 24
6.2 Example 2 – ELF becomes ILF 24
6.3 Example 3 – ELF becomes ILF with modifications 25
6.4 Example 4 – Splitting an ELF 26
6.5 Example 5 – Adding an ELF 26
6.6 Example 6 – Modifying an ILF, 1 26
6.7 Example 7 – Changing the name of a DET 27
6.8 Example 8 – Modifying a heading in a report 27
6.9 Example 9 – Adding / modifying / deleting DET's 27
6.10 Example 10 – Determining the % of modified DET's and FTR's 27
7 Glossary 29
8 Bibliography 32

FOREWORD TO VERSION 1

You are reading the guideline "FPA for Software Enhancement", published by the Netherlands Software Metrics users Association (Nesma), and written by members of the working group "FPA for Enhancement and Maintenance".

To safeguard and control the development and operational status of information systems is a difficult task. This task becomes even more difficult as these systems become more complex. The reasons for this are closely aligned with measurements of (in terms of usable functionality):

- the functionality that is offered to the user by the system;
- the effort and materials needed to deliver this functionality to the user;
- the effort and materials needed to keep the system operational in the enhancement and maintenance phase.

Function point analysis (FPA) can be used to:

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

The Nesma (Netherlands Software Metrics Users Association) first known as the NEFPUG (Netherlands Function Point Users Group) 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;
- 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;
- making recommendations on the use of software metrics;
- collecting and publishing literature on software metrics;
- collaborating with organisations with similar interests;

- liaising and collaborating with the other software metrics user groups and associations worldwide, including IFPUG (USA), ASMA (Australia) and FESMA (Europe).

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.

The working group consisted of five members:
- 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.

Zeist, Netherlands, June 1998

FOREWORD TO VERSION 1 OF 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

FOREWORD TO VERSION 2.2.1

At the beginning of 2008 work was started to update the documentation to the new situation that was a result of the migration of the Nesma counting practises from version 2.0 to version 2.2. This migration was necessary because of the ISO certification of the Nesma FPA method (ISO 24570).

The changes made to this guideline were carried out by members of the FPA core group at Getronics PinkRoccade:

- Ton de Groot,
- Rini Scholten, and
- Theo Thijssen.

In July of 2008 the updated document was extensively reviewed by the following members of the Nesma executive board:

- Jabob Brunekreef,
- Jolijn Onvlee,
- Adri Timp,
- Wim Visser, and
- Hans Vonk.

Editor-in-Chief was:

- Ton Dekkers.

This version has been translated by Robert Louwers (ABN/AMRO) and was reviewed by Ton Dekkers (Galorath International Limited) and Adri Timp (Equens SE).
We especially call your attention to the disclaimer in section 1.6.

Netherlands, September 2009

FOREWORD TO VERSION 2.3

The 2019 update is initiated after the ISO certification of the Nesma FPA method (ISO 24570) as Nesma counting practises version 2.3. We are also glad we received two investigations done by parties using this guideline, therefore we could rephrase the disclaimer that was mentioned in section 1.6 of previous version to remarks.

The changes made to this guideline were carried out by:

- Alexander Vermeulen;
- Ton Dekkers (Editor-in-Chief).

The updated document was extensively reviewed by the following members of Nesma:

- Wim Visser,
- Hans Bernink,
- Jolijn Onvlee,
- Jacques van der Knaap,
- Martin Jacobs.

Netherlands, April 2019

1 INTRODUCTION

1.1 Purpose of this Document

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 define how FPA should be used to determine the functional size of an information system. Both guidelines follow the “Albrecht” method in maintenance and enhancement situations because a more detailed instruction is covered by the guideline.

Now that the methodology of analysing new software projects has been researched and described, it is interesting to know whether FPA can be applied to maintenance and enhancement situations, 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.

Several organisations started using the method in actual practice. Their experiences resulted in further refinement of the guidelines and the inclusion of examples to illustrate the application of the method.

In 2019 an updated version of this guideline was published in order to conform to Nesma 2.3.

1.2 Objectives of the Guidelines

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

The goal is to provide an objective and replicable method for assessing the scope and size of enhancements. The method is objective in that the results obtained are independent of the person applying the method: two different people using the same guideline will 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 Nesma publication “Definitions and Counting Guidelines for the Application of Function Point Analysis” [Nesma, 1], and the IFPUG "Function Point Counting Practices Manual" [IFPUG, 1].

1.4 Intended Audience

These guidelines are intended for anyone performing Function Point Analysis and wanting to measure the size of enhancement projects more precisely. It is assumed that the reader is familiar with the standard ISO certified FPA methods as documented in [Nesma, 1] and [IFPUG, 1], (see section 1.3) and can be used in combination with both methods.

1.5 Scope of the professional guide

Nesma considers 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 Remarks

Nesma is the custodian of Function Point Analysis according to the "Definitions and Counting Guidelines for the Application of Function Point Analysis" [Nesma, 1]. That version (in English) is certified by ISO under number 24570. The IFPUG "Function Point Counting Practices Manual" [IFPUG, 1] is certified by ISO under number 20926. In both these versions no mention is made of the method as set forth in this guideline. As such, this method is therefore neither a part of the Nesma ISO certified method nor part of the IFPUG ISO certified method.

The method described in this guideline is an alternative one and has been found in practice to provide an indication of the size of enhancements. The large interest in this document illustrates its need in the enhancement domain. Over the years various examples of its application have been published. In the bibliography (Chapter 8) some references to documents are added.

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.

Comments and suggestions for further improvement of this method may be sent to office@nesma.nl. This will help Nesma to improve and refine this method to measure the size of enhancement projects.

1.7 Future versions

When changes 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 https://nesma.org/.

1.8 Organisation of the Guide

After 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

In this paragraph we start by looking more closely at the terms which define the scope of the guideline.

2.1.1 Maintenance

Maintenance includes all those activities necessary to sustain the operation of a system without modifying its functionality. Maintenance is carried out in the environment in which the system operates but is also applied to the technical infrastructure that can vary widely among organisations.

In this guideline we have refrained from offering a budgeting method or model for maintenance. A separate publication "Budgeting the Operational Costs of Information Systems" [Nesma, 2] is available for this.

2.1.2 Conversion

Conversion activities are not included in this guideline because they can be manifested in so many different forms. Conversion can be, for example, 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. It is also often not evident which forms of conversion should be considered maintenance.

When enhancements are done, it is sometimes necessary to develop a specific conversion system. This conversion system can be considered to be new development and its functional scope and size can be determined using the standard FPA method.

2.1.3 Enhancement

In “Definitions and counting guidelines for the application of function point analysis” [Nesma, 1] a functional change is defined:

Functional specifications of one or more functions may be adjusted on the basis of a change request. If these adjustments lead to activities (adjustment of the design, software code, testing, etc.) intended to bring the application in accordance with the changed specifications, then these functions should be considered as functions within the scope of the enhancement release.

To determine the size of a function after the change, the same counting guidelines are used as for development.

All functions mentioned in the change request, and all functions that should be changed as a result of the proposed changes, have to be analyzed in the function point analysis of the enhancement release.

After the enhancement, the transactional function should be functionally changed from the user's point of view. The user's primary intent of this transactional function has not changed.

A transactional function referred to in the change request, is to be considered as functionally changed if it meets at least one of the following criteria:

- one or more DETs are added to the transactional function, and/or one or more DETs of the transactional function are changed (see subclause 3.7.4), and/or one or more DETs are removed from the transactional function;
- one or more logical files are added to the transactional function and/or one or more logical files are removed from the transactional function;
- in the enhancement release a data function is changed and at least one of the modified DETs of this data function is part of the transactional function;
- the logical way of processing of the transactional function is changed in the enhancement release (for example as a result of added, modified and/or deleted validations or calculations).

A data function referred to in the change request, is to be considered as functionally modified if it meets at least one of the following criteria:

- the structure of the data function has changed because in the enhancement project one or more DETs are added to the data function, and/or one or more DETs of the data function are changed (see subclause 3.7.4) and/or one or more DETs are removed from the data function;
- the nature of the data function has changed in the enhancement project. This is the case where as a result of functional changes to the transactional function (see subclause 3.7.2) the data function changes from ELF to ILF or vice versa.

A DET changes if it meets at least one of the following criteria:

- the length (number of positions) changes;
- the data type changes (for example from alphanumeric to numeric);
- the number of decimal places changes.

In the context of this guideline the definition a functional change is adopted. The detailed part of the description refers to the complexity of the function and the related size.

However, the phrase "If these adjustments lead to activities (adjustment of the design, software code, testing, etc.) intended to bring the application in accordance with the changed specifications, then these functions should be considered as functions within

the scope of the enhancement release." This means that an Enhancement release also covers cosmetic changes when they are required by the user. All implicated changed functions are included in the analysis of the Enhancement Release

Other types of maintenance, such as corrective, preventive or perfective maintenance are not included in this guideline because they do typically not lead to functional changes to the information system. Because Function Point Analysis expresses the functionality of an information system in terms of transactional functions and data functions, these principles have also been applied to the FPA method for software enhancement.

A prerequisite condition must be that the maintenance is applied to an existing and operational information system.

Enhancements resulting only in the addition of new functions, with no changes being made to existing functions, should be treated as "added 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 as set forth in [Nesma, 1] and [IFPUG, 1].

An information system becomes operational only after it has been formally accepted by the customer.

2.2 Prerequisites

The following are required to carry out a function point analysis of an enhancement:

- a high level or detailed function point analysis of the current system affected by the enhancement (current FPA functional size);
- documentation describing the affected part of the current system in order to judge the enhancement proposal;
- an enhancement proposal that describes the modifications to be made;
- a test plan for the affected changes.

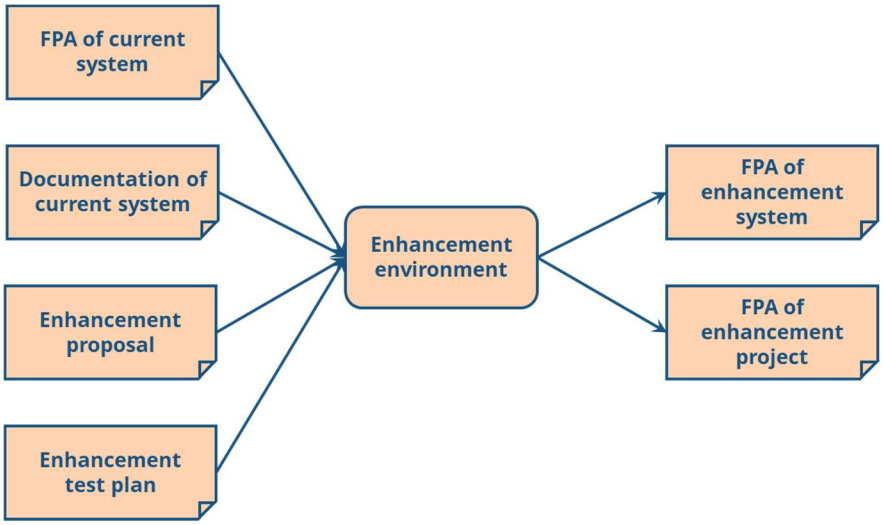


Figure 1 - Determining size of enhancement

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.

In addition to the principles stated by "enhancement" in paragraph 2.1 "Limiting conditions", the following considerations must be taken into account when determining the functional size of the enhancement project:

- *A Function Point functional size is available for the affected parts of the existing system.*
  The method presented in this guideline takes as its starting point the function point analysis results obtained for the existing system. A high level or detailed function point analysis must exist or must be made of the current system, or at the very least, of the part that will be modified. This Function Point Analysis should at least summarize the individual functions, and for each function, the size in function points, the number of associated Data Element Types (DET's) and logical data functions (FTR's).

- *The current system is well documented.*
  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 ELFs) and the user interface. Good functional system documentation (logical datamodel, functional design) is crucial to identify the scope and size of the enhancement.
  The analyst needs to consider:
    - the extent of changes to individual transactional and data functions;
    - the wider implications of the specified changes on other transactional and data functions.

- *An enhancement proposal is available*
  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.

If a function point analysis of the total system is unavailable, or if only partial function point analyses are available of earlier enhancements, but all other criteria have been met, then the scope of the enhancement can be determined only after the missing functions and their size have been determined.

- *A test plan is available*
  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

Six steps are necessary for determining the size of the enhancement expressed in enhancement function points:

1. Identify the transactional and data functions within the scope of the enhancement project and determine their functional size.
2. Determine which transactional and data functions are to be added.
3. Determine which transactional and data functions are to be deleted.
4. Determine which data functions are to be changed and determine the impact factor.
5. Determine which transactional functions are to be changed and determine the impact factor.
6. Calculate the number of Enhancement Function Points.

The analysis is primarily concerned with determining the FPA functions which are added, modified or deleted. For this part of the analysis Function Point Analysis (FPA) is used. The result is a summary of the impacted FPA functions with their functional size ($\sum FP_{BASE}$).

During enhancement, transactional functions and data functions can be added, modified or deleted. With regard to deleted transactional functions and data functions, the number of function points before deletion is decisive; for added and modified transactional functions and data functions, the number of function points after modification is decisive. The impact of an enhancement may go beyond what is initially apparent from the

enhancement proposal. For example, the change of a logical file or transaction may impact other transactions or logical files.

After this, every impacted function needs to be carefully assessed to identify the extent of the impact of enhancement on the function. The impact factor (I) reflects the degree of change of each identified (data and transactional) function.

Finally, the enhancement size of each affected transactional and data function is calculated by multiplying its base size (FP_{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).

In chapter 3, "Methodology", the relationship between the original and the new unit of measure will be described.

3 METHODOLOGY

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

1. Identify the transactional and data functions within the scope of the enhancement project and determine their functional size.
2. Determine which transactional and data functions are to be added.
3. Determine which transactional and data functions are to be deleted.
4. Determine which data functions are to be changed and determine the impact factor.
5. Determine which transactional functions are to be changed and determine the impact factor.
6. Calculate the number of Enhancement Function Points.

In this chapter these steps will be further refined.

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 function points, $\sum FP_{BASE}$.

3.2 Determine which transactional and data functions will be 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 size of the added functionality is expressed as $\sum FP_{ADDED}$. The impact factor for added functions is 1.00.

Hence, the number of enhancement function points for a single added function is determined as follows:

$$\mathbf{EFP_{ADDED} = FP_{ADDED}}$$

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 which transactional and data functions will be 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 total size of the deleted functions is expressed as $\sum FP_{DELETED}$. For deleted functions an impact factor of 0.40 is used. The number of enhancement function points for a single deleted function is determined as follows:

$$\mathbf{EFP_{DELETED} = FP_{DELETED} \times 0.40}$$

This means, for example, that 6 function points deleted will result in 6 x 0.40 = 2.4 enhancement function points.

3.4 Determine which data functions will be modified and determine the impact factor

A data function can be either an internal logical file (ILF) or an external logical file (ELF). 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 ELF is changed into an ILF or vice versa).

Determine which data functions will change and how many function points each data function represents **after** the change, applying the standard FPA rules. The function point size of the changed data function is expressed as $FP_{CHANGED}$.

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).

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

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

Percentage DETs	≤ 33%	≤ 66%	≤ 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 logical file becomes an internal logical file), a value of 0.40 is used for the impact factor. However, in case of a change of type one needs to check if there is also an internal change of the Logical File (change 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 ($I_{CHANGED}$) (see example 3 in Chapter 6).

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

$$\mathbf{EFP_{CHANGED} = FP_{CHANGED} \times I_{CHANGED}}$$

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 ELF or an ILF is *split* 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 ELF and an ILF are *combined*, two deleted data functions and one added data function are counted.

3.5 Determine which transactional functions will be modified and determine the impact factor

The transactional functions that change are identified and the size of each transaction **after the change** (the enhancement) 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. To determine the functional size of a transactional function after the change the same counting guidelines are used as for new-built systems, applying the standard FPA rules. The number of function points after the change for each transactional function is expressed as $FP_{CHANGED}$.

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 at least 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.

In general, 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 a Logical Data File (ILF or ELF) that is added, changed or deleted;
- the user interface is functionally 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:
  - static data is changed or moved in a report or other media,
  - a heading is replaced or changed in a report or on a screen,
  - see also example 9 in Chapter 6.

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.

These steps are explained below.

3.5.1 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 will be included in the scope of the enhancement project.

3.5.2 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).

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

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

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

3.5.3 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	**≤ 66%**	**≤ 100%**	**> 100%**
≤ 33%	0.25	0.50	0.75
≤ 66%	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, you should consider whether enhancing the transaction is still meaningful.

3.5.4 Calculate the enhancement function point size

The enhancement function point size of a single transactional function is calculated as follows (see also example 11 in Chapter 6):

$$EFP_{CHANGED} = FP_{CHANGED} \times I_{CHANGED}$$

3.6 Calculate the size of the enhancement project

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

$$EFP_{TOTAL} = \sum EFP_{ADDED} + \sum EFP_{DELETED} + \sum EFP_{CHANGED}$$

3.7 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 function point size of the application prior to the change (FP_{BASE}) using the standard FPA method.
2. Identify the transactional and data functions deleted from the existing application and determine their function point size ($\sum FP_{DELETED}$).
3. Determine the transactional and data functions changed. Calculate the number of function points these represent before and after the enhancement ($\sum FP_{AFTER}$ and $\sum FP_{BEFORE}$), using the standard FPA method.
4. Determine the transactional and data functions added to the system and calculate how many function points these represent ($\sum FP_{ADDED}$).
5. Calculate the size of the system after enhancement (FP_{NEW}).

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

$$FP_{NEW} = FP_{BASE} + (\sum FP_{ADDED}) + (\sum FP_{AFTER\text{-}CHANGE} - \sum FP_{BEFORE\text{-}CHANGE}) - (\sum FP_{DELETED})$$

Note: The Impact Factor does not play when determining the size of the system after enhancement.

3.8 High level Function Point Analysis for Software Enhancements

In practise there is a tendency in FPA to speed up the sizing process and to apply High level Function Point Analysis. This will slightly change the approach of FPA for Software Enhancement. You still can apply the detailed approach at the base size obtained by the High level Function Point Analysis. A practical approach might be to define a more high level way of identifying the impact factor. The challenge is to keep it objective.

Suggestion for default impact factors to be used for high level EFPA:

High Level EFPA defaults	Impact Factor
New functionality	1.00
Changed functionality	0.50
Removed functionality	0.40

Table 4 – High Level EFP Impact Factor defaults

In contracts other default impact factors can be agreed upon, based on the specific situation.

4 TESTING IN ENHANCEMENT PROJECTS

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. Not only do the functions directly impacted by the enhancement need to be tested, but also all other affected functions.

The size of the functions to be tested is measured in test function points (TFPs). When determining the number of TFPs the impact factor per function is not taken into account. Also, no account is taken of whether a function was added or changed. Deleted functions are excluded from the analysis of test function point size (TFP).

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 size of each function is derived using the standard FPA counting practices for new software development: the number of function points (FP_{TEST}) is the sum of the sizes of the transactional and data functions involved in the tests. In general, testing is done on *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*. Deleted functions are therefore not included in the scope and consequently do not add to the total number of TFPs.

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

$$\mathbf{TFP = \sum FP_{TEST}}$$

Note: The abbreviation TFP must not be confused with the term TP (Test Points) in Test Point Analysis (TPA). The approach used in this guideline differs from TPA. TPA is a methodology for measuring the functional size (based on FP) of structured testing and expressing this in test points, as defined by Tmap®.

5 BUDGETING ENHANCEMENT

The technique described in this guideline is based on the standard FPA method. When estimating small enhancement projects, on average, the estimations will be correct, but large deviations above and below may occur. With larger projects, there will likely also be deviations above and below, but when regarding the total enhancement project, these differences will most likely cancel each other out.

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 development measured using the standard FPA method.

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

Total enhancement effort =
(number of EFPs x hours per EFP) + (number of TFPs x hours per TFP)

6 EXAMPLES

6.1 Example 1 - Expanding an ILF

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 (x100%) = 0,378378 (x100%), which is between 1/3 (x100%) and 2/3 (x100%). From table 1, this change gives in an impact factor of 0.50. The enhancement function point size is 10 x 0.50 = 5 EFP.

6.2 Example 2 – ELF becomes ILF

Situation
Information system A uses an external logical 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 logical file (ELF) is imported into system A and converted to an internal logical file (FTR). An impact factor of 0.40 is used when an ELF 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 0.40, therefore the change results in 7 x 0.40 = 2.8 EFP.

6.3 Example 3 – ELF becomes ILF with modifications

Situation
Information system A uses an external logical 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 0.40 is used when an ELF is changed to an ILF or vice versa. However in this case the effect of the structural change to the data function must also be taken into account. The percentage change is:

25/45 (x100%) = 0,5555... (x100%).

From Table 1, this change (between 1/3 and 2/3) gives an impact factor of 0.50. This is greater than the impact factor for a change of type (from an ELF to an ILF), so the higher value of 0.50 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 0.50 and so the change gives rise to 7 x 0.50 = 3.5 EFP.

6.4 Example 4 – Splitting an ELF

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

Question
How do we account for this change?

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

6.5 Example 5 – Adding an ELF

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

Question
How do we account for this change?

Answer
Count one added external logical 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.

6.6 Example 6 – Modifying an ILF, 1

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.

6.7 Example 7 – Changing the name of a DET

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.

6.8 Example 8 – Modifying a heading in a report

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 considered cosmetic.

Question
Do we need to account for this change?

Answer
The function will change as a result of the cosmetic change, this on request of the user to meet requirements. As no DETs are changed the impact factor = 0.25.

6.9 Example 9 – Adding / modifying / deleting DET's

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.

6.10 Example 10 – Determining the % of modified DET's and FTR's

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 external output 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 x 0.75 = 3.8 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, requested by the user, 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). Cosmetic changes are not considered enhancement in the formal Nesma counting guidelines. Because these changes are requested by the user to fulfil the requirements, EFPA considers them as part of the enhancement.

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

DET
Data element type.

EFP
Enhancement function point.

ELF
External logical 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.

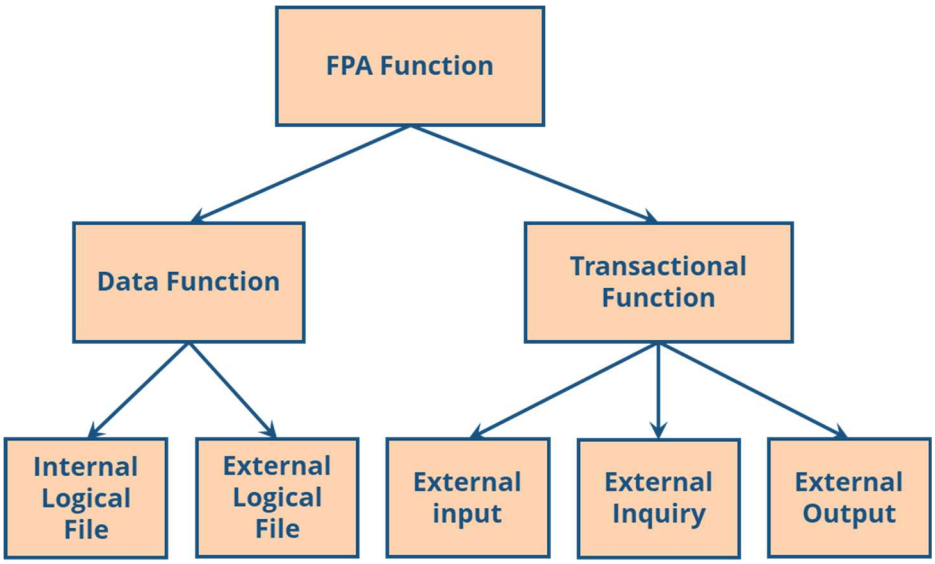
Enhancement proposal
A formal request for enhancement of an operational information system. The proposal must be sufficiently comprehensive to enable the scope and impact of the enhancement to be determined.

FTR (File Type Referenced)

- An internal logical file (ILF) read or maintained by a transactional function.
- An external logical file (ELF) read by a transactional function.

Function

An external input (EI), external output (EO), external inquiry (EQ), internal logical file (ILF) or external logical file (ELF) 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 logical file (ELF); 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 Analysis
A function point analysis 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.

8 BIBLIOGRAPHY

[IFPUG, 1] **IFPUG Function Point Counting Practices Manual**, release 4.3.1
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 2010

[Nesma, 1] **Definitions and counting guidelines for the application of function point analysis.**
A practical manual, version 2.3. (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. It is identical to the version 2.2 in Dutch.
For more information, see the Nesma web site www.nesma.org (English).

Date of issue: 2018
Size: 112 pages A4-format
ISBN 978-90-76258-45-4

[Nesma, 2] **Begrotingsmodel voor de exploitatielasten van informatiesystemen**

(Budgeting the Operational Costs of Information Systems; Only available in Dutch)

Date of issue: November 1998
Size: 117 pages A4-format
ISBN 90-76258-11-2

[Nesma, 3] **Onderhoud en Functiepuntanalyse**

Version 2.2

(Function Point Analysis for Software Enhancement)

Date of issue: August 2008.
Size: 21 pages A4-format
ISBN 90-76258-13-9

[Publication, 1] **The Added Value of Enhancement Function Points**
An Empirical Evaluation
Frank Vogelezang, Jelle de Vries
IWSM 2014

[Publication, 2] **Statistical Study of the Relationship of the FPA / EFP (Nesma) with the Effort Estimation of a Project**
LEDA Mc, 2018