



**DEFINITIONS AND COUNTING GUIDELINES  
FOR THE APPLICATION  
OF FUNCTION POINT ANALYSIS**

**Release notes**

**Version 2.3**

[nesma.org](http://nesma.org)



## 1 FOREWORD

In 2005 the Nesma FPA counting guidelines became the international standard ISO/IEC 24570:2005. In the 2010 and 2015 systematic reviews the standard was prolonged as International Standard. The Nesma Counting Practices Committee has made an inventory of questions and issues that occurred in practice with the use of the standard. This inventory has resulted in a list of proposed changes to the standard.

Based on this list the Counting Practices Committee has produced a new version of the Nesma FPA counting guidelines. All changes have the primary intent to improve the unambiguity of the standard, while leaving the essence of the standard unchanged. This means that functional size measurements based on the new version of the standard will give the same results as functional size measurements based on the earlier versions 2.1 (English) and 2.2 (Dutch).

In this version not only the items of the list of changes have been implemented, but the whole text of the standard has been modernized to reflect the common practices in software engineering in 2018. This means that it is impractical to describe all the changes from the previous version to the current version. In chapter 2 we describe the major changes to the standard. In chapter 3 we describe changes to each chapter in more detail.

We hope that you will appreciate the improvements to the standard as much as we do.

The Counting Practices Committee:

- Adri Timp
- Frank Vogelesang
- Jacqueline Eshuis
- Jacques van der Knaap
- Jolijn Onvlee
- Marinus Spaan
- Martin Jacobs
- Wim Visser

## 2 MAJOR CHANGES

### 2.1 Changed terminology

Some terms from the previous version of the guidelines have been changed:

- **Count** was used in three different meanings:
  - 1 As the process to perform a function point analysis. Wherever this meaning is applicable, the word **analyze** or **analysis** is used.
  - 2 As the result of a function point analysis. Wherever this meaning is applicable, the expression **functional size** is used.
  - 3 Wherever count is used as an instruction the word **count** is used.
- The term **estimated function point count** has been abandoned, and replaced by **high level function point analysis**. This has been done, since the term estimated could be associated with less accuracy.
- The term **External Interface File** has been abandoned, and replaced by **External Logical File**. External Interface File could be associated with some form of technical implementation. The new term expresses that it is a similar concept as an Internal Logical File, but maintained by another application.
- The term **chapter** has been abandoned in favour of **clause**, to make it easier to conform to ISO rules on the way standard documents should be composed.
- For the same reason the term **section** has been abandoned in favour of **subclause**.

These changes have resulted in a large number of changes in the text of the guidelines. All these changes will not be documented in the release notes.

### 2.2 Removed parts of the guidelines

- **Chapter 11** with examples has been removed. The need for examples for certain situations has changed more quickly than the need to make changes to the guidelines. To facilitate this examples will be published separately and can be extended as the need arises, without having to change the guidelines. Examples have never been intended to be a normative part of the guidelines, but are meant to clarify the guidelines. To make it clear that the examples are not a normative part of the guidelines was an additional argument to remove them from the guidelines.
- **Annex C** on the application of FPA including general system characteristics has been removed since this approach has been abandoned in 2005 when the Nesma FPA method became an official ISO/IEC standard.
- **Annex D** on the description of the 14 general system characteristics has been removed since the use of these characteristics has been abandoned in 2005 when the Nesma FPA method became an official ISO/IEC standard.

## 2.3 New parts of the guidelines

- **Subclause 3.7** on the definition of a functional change. In previous releases of the guidelines this has never been explicitly defined, which led to different interpretations of functional change. Therefore we added a normative definition of a functional change to make this unambiguous.  
The new definition is based on the definition that was introduced in the guide "Function point analysis for software enhancement".
- **Subclause 4.23** describes a generic rule for counting data element types. This generic rule has been introduced in the Dutch version of the guidelines and is now included in the ISO/IEC standard.
- **Subclause 9.3** on the complexity of External Inquiries has been rewritten. In previous releases of the guidelines the complexity of an External Inquiry was based on either the complexity of the input part or of the output part. This has been simplified to a single way of determining the complexity, based on both the input and the output part. This means that subclause 9.3 now has a self-contained set of guidelines to determine the complexity of an External Inquiry. The impact on function point analyses based on previous versions of the guidelines is negligible.
- **Annex C (new)** on the increase of functional size during the software development lifecycle. This non-normative annex describes the mechanisms of autonomous growth and scope creep and explains why they cause an increase of the functional size during the software development lifecycle.  
This annex is based on the guide "The application of function point analysis in the early phases of the application life cycle".

## 3 DETAILED CHANGES PER CLAUSE

Next to the major changes that have been described in the previous chapter, a number of smaller changes to the guidelines is described here. Due to the restructuring of the former clauses 1 and 2 the numbering of the remaining clauses has decreased by one. This will not be explained in the detailed changes per clause.

### 3.1 Changes to clause 1

Clause 1 is now a combination of the content of the clauses 1 and 2 and a small part of clause 3 to make it a concise description of the scope of the guidelines.

### 3.2 Changes to clause 2

In subclause 2.7 the definition of a *transactional function* has been rewritten to make it more unambiguous what the criteria are for such a type of function.

In subclause 2.10 a paragraph has been added to make clear that the functional size is only a measure for size and that the translation of that size to a project budget is beyond the scope of this standard.

### 3.3 Changes to clause 3

In subclause 3.1 two steps from the step-by-step plan to perform an FPA that had been combined in the previous version of the guidelines are now shown as separate steps. Visually the plan appears to have been extended with an extra step, but in substance, the content has not changed.

In subclause 3.2.1 the formulae for calculating the indicative functional size have been simplified to make them better understandable.

In subclause 3.2.2 the new term **high level function point analysis** is introduced. Also a sentence has been added to make it explicit that the complexity values for high level FPA overrule all other complexity value rules. In the section on applicability a distinction has been made between the use in a waterfall approach and in agile approaches.

In subclause 3.2.3 the section on applicability has been extended to distinguish between the use of detailed FPA in a waterfall approach and in agile approaches.

In subclause 3.3 a note has been added to refer to the guide "FPA applied to UML / Use Case documentation".

In subclause 3.5.1 the definition of application boundary has been simplified to make it better understandable.

In subclause 3.5.3 the order of the steps to determine the functional size of enhanced applications has been changed to match the order of the formula.

In subclause 3.6.1 the definition of an *enhancement* has been abandoned, and replaced by a reference to the new subclause 3.7.

In subclause 3.6.3 the order of the steps and the order of the formula to determine the functional size of an enhancement project have been changed to match the order of the formula in subclause 3.5.3.

In subclause 3.6.3 a note has been added to refer to the guide on "FPA for software enhancement".

Subclause 3.8 and its underlying subclauses on how to apply FPA in specific situations has been modernized to reflect the current practices in software development. This means that the subclause on GUI environments has been merged into the subclause on screens or windows, since GUI environments are no longer an exceptional practice. In subclause 3.8.2 a note has been added to refer to the guide on "Estimating Packaged Software" that deals with the cost aspects of packaged software.

In subclause 3.9.1 a note has been added to refer to the guide on "The application of Function Point Analysis in the early phases of the application life cycle".

In subclause 3.9.2 a paragraph has been added on autonomous growth and scope creep that can influence the functional size as it appears in earlier phases with respect to the final functional size that will be released. A note to the new (non normative) annex C has been added in which both phenomena are explained in more detail.

### **3.4 Changes to clause 4**

In subclause 4.13 two additional examples of help functionality have been added to reflect current software development practices. The text has also been modified to make it clear that these are examples and not an exhaustive list of allowed help facilities. In addition to subclause 3.2.2 it has been made explicit that the counting rule for a detailed FPA will be overruled by the general counting rule of a high level function point analysis.

In subclause 4.20 the example for situation 3 has been replaced by a more appropriate example.

In subclause 4.21.2 it has now been made explicit that the order of the steps to carry out denormalization from a data model in third normal form is a mandatory order.

In subclause 4.22 a graphical representation has been added to the different situations that can occur on shared use of data.

Subclause 4.23 has been added to offer a generic rule for counting data element types. This generic rule is translated to concrete expressions for the various transactional function types in chapters 7, 8 and 9.

### **3.5 Changes to clause 5**

Subclause 5.2.c describes the three possible reasons when there are no maintenance functions for a candidate internal logical file has been clarified. The text from earlier versions of the guidelines could be misinterpreted.

Subclause 5.3.c describes how to act when several entity types in a normalized data model form together an internal logical file. The text has been simplified.

### **3.6 Changes to clause 6**

Subclause 6.2.h has been extended to bring it in line with subclause 5.2.l that describes a similar situation for an Internal Logical File.

Subclause 6.3.c describes how to act when several entity types in a normalized data model form together an internal logical file. The text has been simplified.

Subclause 6.3.d describes how to determine the number of record types for an External Logical File. The text has been simplified.

### **3.7 Changes to clause 7**

In clause 7 only editorial improvements have been conducted.

### **3.8 Changes to clause 8**

In subclause 8.1 the term *operator* has been explicitly defined and examples have been added.

Subclause 8.2.a has been rearranged to bring it in line with subclause 7.2.a that describes a similar situation for an External Input.

Subclause 8.2.g has been rephrased to make it less prone to misinterpretation.

Subclause 8.3.h has been rephrased to bring it in line with subclause 7.3.e that describes a similar situation for an External Input.

### **3.9 Changes to clause 9**

Subclause 9.2.a has been rearranged to bring it in line with subclause 7.2.a that describes a similar situation for an External Input.

Subclause 9.2.e describes that the display of data before editing is not counted when not explicitly specified as such. The text has been clarified to make it less prone to misinterpretation.

All the subclauses in subclause 9.3 are new to make them a self-contained set of rules for determining the complexity of External Inquiries. This approach is in line with the IFPUG approach.



### **3.10 Changes to Annex A**

In Annex A only editorial improvements have been conducted.

### **3.11 Changes to Annex B**

In Annex B a number of editorial improvements have been conducted to modernize the glossary definitions and to bring it in line with the new version of the counting guidelines and definitions from other ISO-standards.