INTERNATIONAL STANDARD

ISO/IEC 24570

First edition
2005-02-15

Software engineering — NESMA functional size measurement method version 2.1 — Definitions and counting guidelines for the application of Function Point Analysis

Ingénierie du logiciel — Méthode de mesure de la taille fonctionnelle NESMA, version 2.1 — Définitions et manuel des pratiques de comptage pour l'application de l'analyse des points fonctionnels

Reference number
ISO/IEC 24570:2005(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword ... v

Introduction ... vi

1 Scope ... 1

2 Overview ... 1
2.1 Objective of this International Standard ... 1
2.2 Focus of this International Standard ... 1
2.3 Organization of this International Standard ... 2

3 Introduction to FPA ... 3
3.1 Brief description of FPA ... 3
3.2 Use of FPA: application function point count versus project function point count ... 4
3.3 The types of function point counts ... 5
3.4 Function point counts during a project ... 5
3.5 Scope of the count and boundary of the application to be counted ... 5
3.6 Users ... 5
3.7 Functions and function types ... 6
3.8 The complexity of a function ... 6
3.9 The valuing of function types ... 7
3.10 The function point count ... 7

4 Guidelines to carry out an FPA ... 7
4.1 Step-by-step plan for carrying out an FPA ... 8
4.2 Types of function point counts and their accuracy ... 8
4.3 The role of the quality of the specifications ... 10
4.4 FPA during a project ... 11
4.5 Determining the application function point count ... 11
4.6 Determining the project function point count ... 13
4.7 FPA in specific situations ... 16
4.8 Illustration: FPA and the system life cycle ... 20

5 General counting guidelines ... 25
5.1 Counting from a logical perspective ... 25
5.2 Applying the rules ... 25
5.3 Built functionality, not requested functionality ... 25
5.4 Double counting ... 25
5.5 Production of re-usable code ... 26
5.6 Re-use of existing code ... 26
5.7 Screens and reports ... 26
5.8 Input and output records ... 26
5.9 Security and authorization ... 26
5.10 Operating systems and utilities ... 26
5.11 Report generators and query facilities ... 27
5.12 Graphs ... 27
5.13 Help facilities ... 27
5.14 Error messages and other messages ... 27
5.15 Menu structures ... 28
5.16 List functions ... 28
5.17 Browse and scroll functions ... 28
5.18 Cleaning functions ... 28
5.19 Completeness check on the function point count ... 29
5.20 FPA tables ... 29

5.21 Deriving logical files (data functions) from a normalized data model 30
5.22 Shared use of data 34

6 Internal Logical files 35
6.1 Definition of an internal logical file 35
6.2 Counting internal logical files 36
6.3 Determining the complexity of internal logical files 37

7 External Interface Files 38
7.1 Definition of an external interface file 38
7.2 Counting external interface files 38
7.3 Determining the complexity of external interface files 40

8 External inputs 40
8.1 Definition of an external input 41
8.2 Counting external inputs 42
8.3 Determining the complexity of external inputs 44

9 External Outputs 45
9.1 Definition of an external output 45
9.2 Counting external outputs 47
9.3 Determining the complexity of external outputs 50

10 External inquiries 51
10.1 Definition of an external inquiry 51
10.2 Counting external inquiries 52
10.3 Determining the complexity of external inquiries 53

11 Practical Situations and their solutions 54
11.1 Standard authorization functions 55
11.2 Specific authorization functions 55
11.3 Report generator and query facility 56
11.4 Help functions 56
11.5 Error messages 57
11.6 Menu structures 57
11.7 FPA tables 58
11.8 Denormalization 59
11.9 Counting logical files (data functions) 61
11.10 Combined external inputs 65
11.11 Counting a transaction file 66
11.12 Reports on different media 67
11.13 Daily and weekly processing 69
11.14 Conversion 69
11.15 External outputs with summary information 70
11.16 The number of data elements on a report 72
11.17 Combined external outputs 72
11.18 Combination effects with functions 77
11.19 Querying with several search keys 79
11.20 Screens with list function 81
11.21 Browse and scroll functions 82
11.22 Selection screens and changing data with a search key 85
11.23 Direct and delayed processing 89
11.24 Case study of a customer application 91

Annex A (normative) **The most important features and tables for valuing function types** 95

Annex B (normative) **Function point analysis glossary** 101

Annex C (informative) **Application of function point analysis including general system characteristics** 106

Annex D (informative) **General system characteristics** 113

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 24570 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and system engineering*.

Introduction

Version 1.0 (November 1990)

The NESMA Board set up a counting guidelines committee devoted to the standardization of counting guidelines/definitions in September of 1989. The committee's task was and (still) is to draw up and maintain a NESMA FPA manual.

Version 1.1 (May 1991)

Version 1.1 is a reprint of version 1.0. Except for the improvement of some minor errors, the two versions are the same.

Addendum (May 1994)

The manual *Definitions And Counting Guidelines For The Application Of Function Point Analysis* satisfies a great need and has become a standard in the Netherlands within a short time.

In May of 1991 the Board of the NESMA set up the work group "FPA Case Study" and gave it the task of developing a case study that would present the application of FPA and counting guidelines within a context.

While developing the case study, the work group felt that a number of definitions of counting guidelines needed to be more precise:

- The derivation of logical files from a data model in third normal-form (the so-called denormalization rules)
- A more concrete definition of the concept of FPA table
- Uniform treatment of selection screens
- Dealing with combination effects of functions

The Counting Guidelines Committee established additional counting guidelines for these topics after extensive discussion took place both within the committee itself and within the work group FPA Case Study.

You will find the additional counting guidelines necessary and/or helpful when working out the case. In view of the issue date of the case (mid 1994), the NESMA Board decided to issue these additional counting guidelines as an Addendum to version 1.1 of the Counting Guidelines Manual.

Version 2.0 (April 1996)

This new version of the manual *Definitions And Counting Guidelines For The Application Of Function Point Analysis* incorporates the following improvements:

- The guidelines recorded in the addendum have now been integrated into the manual
- A large number of points in the guidelines have been further clarified
- The results of extensive consultation with the IFPUG have been processed
- The manual's accessibility has been increased further as a result of editorial improvements
- Many examples and illustrations have been added

The committee is of the opinion that the changes made are chiefly an elaboration and further illustration of the guidelines drawn up earlier. In modifying the manual, the committee has worked in such a way that the changes made have as little effect as possible on the results of a function point analysis. Appendix D goes further into this.

The guidelines published in this manual have been applied to a rather large case study with the title, *FPA Case Study "Hotel" For The Application of Function Point Analysis*. Applying the guidelines in practice is explained in this document in detail.

The publication of this version takes precedence over versions 1.0 and 1.1, as well as the Addendum.

English translation of version 2.0 (November 1997)

This English version of the manual is an accurate translation of the Dutch version

Version 2.1 Unadjusted (February 2002)

This version has been developed for the manual to be an ISO recognized standard. The main adaptation is the exclusion of the General System Characteristics. This exclusion conforms to the ISO standard 14143-1 Functional Size Measurement.

Reason for this International Standard

The NESMA was set up in the spring of 1989. (At that time it was called the NEFPUG.) During its first meeting in June, it carried out a study among its participants in order to survey which subjects they were interested in. The standardization of counting guidelines/definitions was high on the list. In reaction to this, the NESMA Board decided to set up a committee devoted to this topic. This committee set itself the task of putting together a International Standard for the theoretical application and the practical use of function point analysis (FPA)[1].

Over the years a number of "dialects" have arisen for FPA. These dialects complicate the goal of determining the number of function points and make it almost impossible for organizations to compare results. One insufficiently acknowledged reason for this is that different interpretations of the "Albrecht" method have arisen.

This International Standard hopes to provide clarity by formulating standards for the definitions and counting guidelines that pertain to FPA.

Intended audience

This International Standard is meant for everyone who performs function point counts; i.e., both for people who count according to the NESMA rules and for those who use the IFPUG rules. For those using the IFPUG rules, the NESMA International Standard can be a valuable supplement to the IFPUG International Standard if the differences stated on the website “WWW.NESMA.ORG” are taken into account. The NESMA International Standard, after all, contains many hints, guidelines, and examples that can be of value to every FPA counter. It is assumed that the reader has some knowledge of FPA. Nevertheless, we have also attempted to produce as complete a International Standard as possible that includes sufficient introductory material and explanation for the new FPA user. For both the maintenance of the IFPUG International Standard and the NESMA International Standard there is a co-operation between the IFPUG CPC and the NESMA CPC.

Departure points of this International Standard

The NESMA FPA method is in principle applicable to all Functional domains.

The following documentation has served as the foundation for this International Standard:

- IBM CIS & A Guideline 313, AD/M Productivity Measurement and Estimate Validation, November 1, 1984.

---

[1] The abbreviation FPA is used for the term Function Point Analysis.

This is an internal IBM publication. The method described in it is usually referred to as *Albrecht '84.*

Future versions

When changes and supplements to this International Standard prove necessary in the future, an entire new version will be produced

Software engineering — NESMA functional size measurement methode version 2.1 — Definitions and counting guidelines for the application of Function Point Analysis

1 Scope

This International Standard:

a) specifies a method to measure the functional size of software,

b) gives guidelines on how to determine the components of functional size of software,

c) specifies how to calculate the functional size as a result of the method, and

d) gives guidelines for the application of the method.

2 Overview

This clause provides an overview to the International Standard "Definitions and counting guidelines for the application of function point analysis". The following questions are answered: What is its aim (subclause 2.1)? What is its focus (subclause 2.2)? How is it laid out (subclause 2.3)?

2.1 Objective of this International Standard

The International Standard attempts to provide a theoretical framework by presenting definitions and standard guidelines. It also tries to illustrate the counting guidelines as concretely as possible by using several practical situations.

2.2 Focus of this International Standard

The International Standard focuses on how the functional size of an application is determined. The International Standard does not go into any of the aspects that play a role when project budgeting is drawn up on the basis of this functional size; e.g., productivity standards and productivity attributes. This particular topic has been described in the manual "*Budgeting on the basis of logical design using function point analysis",* also by the NESMA.

Figure 1 indicates what this International Standard *will* and *will not* cover.

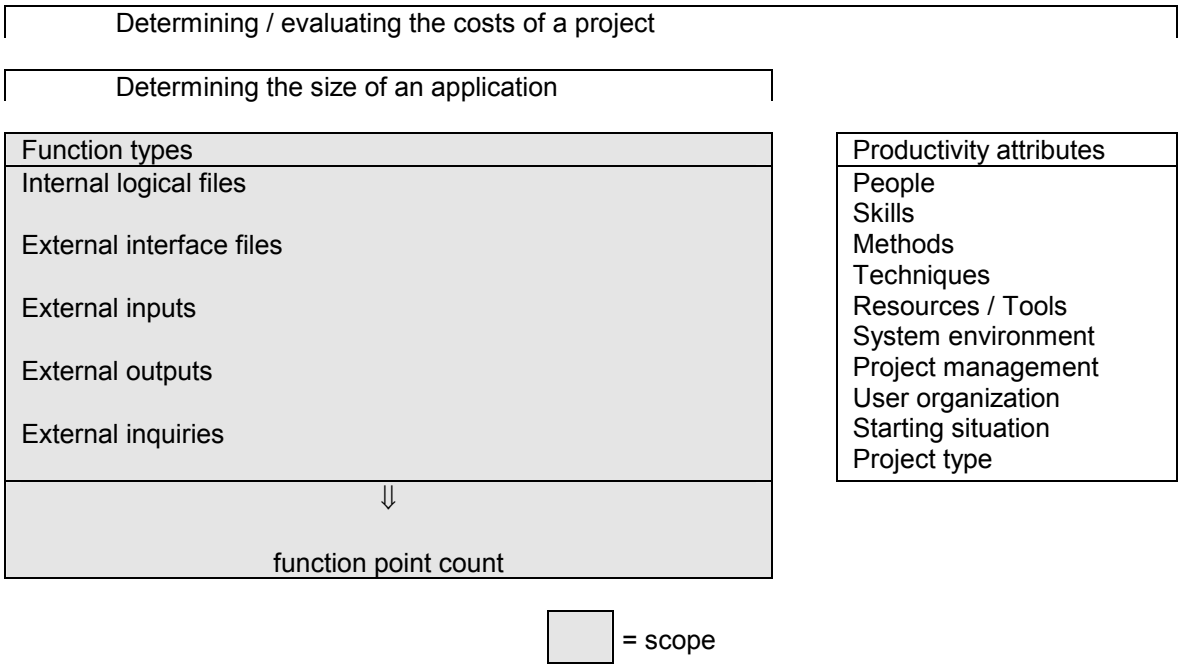

Figure 1 — Scope of the International Standard

2.3 Organization of this International Standard

The terms and concepts used in the International Standard are explained in Clause 2 and defined in Annex B.

Clause 3 provides an introduction to FPA and in which the functional aspect of FPA is emphasized. It will also spell out briefly what FPA is and explains the terms that form the basis for the concept of FPA. Matters such as distinguishing between an application function point count and a project function point count are examined, just as are other various types of function point counts, the role of FPA during a project, users, and function point count.

Clause 4 provides you with an overview of the position of FPA in a project and with the types of function point counts that can be carried out during the life cycle of an application. In other words, the chapter will explain when points can be counted and what information is needed minimally in order to count. The chapter will also give a step-by-step plan for carrying out a function point analysis and indicates how projects, applications, and packages should be counted. Each of these requires their approach.

Clause 5 states general counting guidelines for a function point count.

Clauses 6, 7, 8, 9 and 10 give successively the definitions and guidelines used to identify function types and to determine the complexity of function types for internal logical files, external interface files, external inputs, external outputs, and external inquiries. The guidelines are broken down per function type for identifying the function type concerned, for determining the number of data element types, and for determining the number of record types or referenced logical files.

Clause 11 provides several practical situations and their solutions. The counting guidelines are not repeated explicitly here, but reference is made to the relevant guideline(s) or section(s) on which a solution is based.

Annex A is meant to be a short summary of the guidelines and contains the most important features of each function type, as well as the tables for valuing the function types.

Annex B contains the definitions of the terms that this International Standard uses.

For backward compatibility with previous adjusted function point counts, Annex C describes the application of FPA with the general system characteristics that lead to a value adjustment factor with which the adjusted function point count can be determined from the function point count.

Annex D describes the general system characteristics.

The Alphabetic index of keywords is given after the table of contents.

This International Standard has been set up in such a way that the reader does not necessarily have to start at Clause 1 before continuing on to Clause 2, then 3 and 4 etc. Instead, he can look up what he thinks is important to him. For one person, specific counting guidelines for a particular function type may be important, while someone else may want a more general frame of reference for an initial introduction to FPA.

3 Introduction to FPA

This chapter gives a short description of FPA and explains a number of important concepts related to it. More specifically, section 3.1 provides a brief synopsis of FPA. Sections 3.2 through 3.4 distinguish between the different kinds of function point counts. Sections 3.5 through 3.9 discuss each of the following successively within the context of FPA:

- The boundaries in which counting takes place
- Users
- Function types
- The complexity of a function type
- The valuing of function types

Section 3.10 defines the term " function point count" and describes how it is determined.

3.1 Brief description of FPA

3.1.1 Background, purpose and application of FPA

FPA was developed by A.J. Albrecht at IBM between 1974-1979 as a result of productivity research into a large number of projects. The first version of FPA was introduced in 1979, followed by adaptations based on practical experiences in 1983 and 1984.

FPA is currently applied in countless organizations throughout the entire world. Experiences with the technique are exchanged in user groups: the International Function Point Users Group (IFPUG), the Australian Software Metrics Association (ASMA), the United Kingdom Software Metrics Association (UKSMA), the NESMA, and various other national user groups.

FPA introduces a unit, the function point, to help measure the size of an application that is to be developed or maintained. The word "application" within the framework of FPA means "an automated information system". The function point expresses the quantity of information processing that an application provides a *user*. This unit of measurement is separate from the way in which the information processing is realized in a technological sense. A function point is an abstract term and can be compared somewhat to so-called "rental points". Rental points are based on the number of rooms in a house, the surface area of these rooms, the number of facilities the house has, and the location of the house. This then serves as a measurement for a residence offered to a potential tenant.

FPA was first used to *measure the productivity* of system development and system maintenance after an application was built. It soon became clear that the technique could also be used to support *project budgeting* because the data needed for an FPA can be made available early on in a project.

This FPA method may be applied to all functional domains.

3.1.2 Rationale behind FPA

The three separate words that make up the term "Function Point Analysis" can be used to explain the way of thinking behind FPA.

Function

As mentioned earlier, FPA bases itself on the functionality that an application provides a *user* (see section 3.6). Because users see only the "outside" or the *boundary* (see section 3.5) of an application, FPA examines the specifications that describe the application's exchange of information with its environment. Functionality is derived from incoming and outgoing information flows (these can be both data or control information), as well as from the logical files that an application contains or uses. The functionality of an application is measured by determining the *function types*. (See section 3.7.)

Point

The *complexity* of each function type is determined according to certain standard guidelines. (See section 3.8.) Each function is worth a number of points, depending on its complexity (section 3.9). The sum of these points yields the number of function points (section 3.10. This is the basis for project budgeting.

Analysis

FPA is the analysis of an application or the analysis of the description/specification of an application in order to establish its number of function points. The act of establishing the quantity of an application's function points is also called *function point counting*.

In order to be able to carry out a function point count the following must first be determined:

- Purpose of the function point count (section 2.2)
- Scope and boundaries of the application or project to be counted (section 2.5)

This concludes a summary of the methodology and a brief description of FPA. The sections that follow explain the various terms used in FPA.

3.2 Use of FPA: application function point count versus project function point count

Function point counts can be linked to applications or to projects. This means that a distinction is made between the following two objectives:

- Determining the application function point count
- Determining the project function point count

Application function point count:

> The number of function points that measures the amount of functionality that an application is to supply or has already supplied to a user. It also measures the size of an application that must be maintained.

Project function point count:

> The number of function points that measures the amount of functionality of a new application or of changes to an existing application. Changes to an existing application pertain to adding, changing, and deleting functions. The project function point count is an essential parameter when determining the effort and lead-time required for a project.

Determining the application function point count is elaborated on in section 4.5. Section 4.6 discusses the project function point count further.

3.3 The types of function point counts

One of three kinds of function point counts can be chosen, depending on the degree of detail of the specifications available. The following represent the different types of function point counts. Note that they are listed by degree of detail, number one having the least detail and number three the most:

1. Indicative function point count

2. Estimated function point count

3. Detailed function point count

These function point counts are explained further in section 4.2.

3.4 Function point counts during a project

Function point counts can be carried out at different times during a project. They can therefore be related to the phases of a project; e.g., the planning phase, the execution phase, and the evaluation phase. As a result, the following breakdown of function point counting arises: the *initial function point count*, the *interim function point count*, and the *final function point count*. These counts are discussed further in section 4.4.

3.5 Scope of the count and boundary of the application to be counted

The scope of the count is the set of functional requirements/specifications to be included in the function point count. When you have determined the scope you can define the boundary, the conceptual interface between the application and its users.

As indicated earlier in section 3.1, the scope of the count and the boundary of an application to be counted plays an important role in FPA. Consequently, the boundaries of the application to be counted must first be determined in order to be able to carry out a function point count.

The boundary is necessary in order to be able to determine:

- The application that certain data belongs to

- Which data crosses the boundary

As mentioned in section 3.2, a distinction is made between a function point count for an application and a function point count for a project. Section 4.5.1 provides guidelines for determining the application function point count and section 4.6.1 gives guidelines for determining the project function point count.

3.6 Users

FPA acknowledges three kinds of users:

- The people and/or organizations that use or are going to use the application to be measured. This category includes the following: end-users, functional managers, and operators.

- The owner and/or employee(s) who determine(s) the requirements and wishes included in the specifications. These requirements and wishes are recorded on the basis of the demands of the end-user(s) for example, but also on the basis of requirements that a government or its legislation can impose on the application.

- Other applications that use the data or the functions of the application to be counted.

Because the function point count takes place from the perspective of the user, it is always necessary to have it done in cooperation with the user or, at the very least, to have the result of the count verified by the user. The user, after all, is the only one who can determine whether a certain function is being requested.

3.7 Functions and function types

The function point count measures the size of the functions of an application or a part of an application. The count revolves around the *what* and not the *how* of the application furnished. Only those components that the user requests, that he can recognize, and that he considers significant are assessed. These components are called functions or base functional components. A function belongs to a function type.

FPA defines *functions* as follows:

> The five types of components of which an application exists, as seen from the perspective of FPA. These components determine the amount of functionality an application provides to a user.

Function types are categorized into two main groups:

- Data function types
- Transactional function types

A *data function type* is:

> A logical group of data seen from the perspective of the user.
>
> FPA distinguishes between:

- Internal logical files
- External interface files

A *transactional function type* is:

> A succession of actions that the user sees as a single work unit. FPA distinguishes between:

- External inputs
- External outputs
- External inquiries

Each function type is discussed further in chapters 5 through 9.

3.8 The complexity of a function

The *complexity of a function* is defined as follows:

> The weight of a function on the basis of which a number of function points are allocated to the function.

The complexity of a function is determined by using the appropriate complexity matrix. A separate table has been defined for each function type. Complexity depends on the number of data element types and the number of referenced logical files connected to a given function. Three levels of complexity are distinguished:

Low: Few data element types and/or referenced logical files are involved with the function

Average: The function is neither low nor high as regards complexity

High: Many data element types and/or referenced logical files are involved with the function

3.9 The valuing of function types

After the complexity of a function has been determined as described in chapters 5 through 9, the number of function points can be allocated to the function. This should be done according to the rating illustrated in the table below.

Table 1 — Function point table for determining the value of function types

	ILF	EIF	EI	EO	EQ
Low	7	5	3	4	3
Average	10	7	4	5	4
High	15	10	6	7	6

ILF = Internal logical file

EIF = External interface file

EI = External input

EO = External output

EQ = External inquiry

Specifications are enough to identify functions and their type when carrying out an *estimated* function point count (see section 4.2.2), but it will be difficult to determine the complexity of these functions. In such a case, a data function is rated as *low*, while the rating *average* is used for a transactional function.

3.10 The function point count

The function point count is the sum of the number of function points assigned to each of the functions (in the way described above) that lie within the boundaries of the object to be counted; i.e., the application or the project.

A NESMA-FSM-method measurement result on the FUR/specifications for a piece of software shall be labeled according to the following convention:

F(unction) P(oint) (ISO/IEC 24570:2003, NESMA FSM method V2.1).

The values for the function types are defined in Annex A.

4 Guidelines to carry out an FPA

This chapter indicates how function point analysis should be carried out. To this end, section 4.1 will first present a generally applicable step-by-step plan. Section 4.2 will then indicate how you should act when dealing with an indicative, estimated, and detailed function point count. Section 4.3 goes into the role of the quality of specifications, while section 4.4 explains the use of FPA during a project. Sections 4.5 and 4.6 show how an application and a project function point count are determined in the event of development and in the event of enhancement, respectively. Section 4.7 states what you must take into consideration when dealing with the different ways of recording specifications. Section 4.8 concludes the chapter with an illustration of how the different types of function point counts can be applied during the life cycle of an application. For illustration purposes, this section will assume a general system life cycle as a phasing method for the life cycle of an application.