INTERNATIONAL STANDARD

ISO/IEC 24570

Second edition
2018-02

Software engineering — NESMA functional size measurement method — Definitions and counting guidelines for the application of function point analysis

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

Reference number
ISO/IEC 24570:2018(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Published in Switzerland

Contents

Page

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

This document was prepared by NESMA and was adopted, under the PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information Technology*, in parallel with its approval by national bodies of ISO and IEC.

This International Standard is the latest release in the continually improving Nesma method. This method is a consistent interpretation of functional size measurement in conformance with ISO/IEC 14143-1:2007. The Nesma functional size measurement method is known as Function Point Analysis (FPA)[1] and the unit of functional size is called Function Point.

This second edition cancels and replaces the first edition (ISO/IEC 24570:2005), which is now obsolete. Functional size measurements as determined based on this new edition of the standard are identical to those based on the previous edition of the standard. Results obtained in the past do not need to be updated.

[1] In this document the abbreviation FPA is used for the term Function Point Analysis.

Introduction to this Standard

Reason for this International Standard

Over the years a number of "dialects" have arisen for function point analysis. 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 provides 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 analyses. It is assumed that the reader has some knowledge of function point analysis. Nevertheless, we have attempted to produce an International Standard that is both complete and includes sufficient introductory material and explanation for the new user.

Application of this standard in practice

This International Standard is one component in the Nesma publications. It is recommended that it be read in conjunction with the other Nesma publications. These provide guidance to application of the rules specified within this International Standard and background information to aid in understanding the use and applicability of the resulting functional size. Supporting Nesma publications include the following:

- Examples to illustrate the use of the Nesma method in specific situations and a fully documented Hotel case.
- Nesma website at nesma.org which contains a number of documents that can be used in a specific context, for example guidelines how FPA can be used in a Data Warehouse environment, with UML documentation, or different aspects in contracts.

Organization of this International Standard

Clause 1 describes the scope of this International Standard.

Clause 2 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 analysis and a project function point analysis are examined, just as are other various types of function point analyses, the role of FPA during a project, users, and function point analysis.

Clause 3 provides an overview of the position of FPA in a project and explains the types of function point analyses that can be carried out during the life cycle of an application. In other words, the clause explains when FPA can be applied and what information is needed minimally in order to count. The clause will also give a step-by-step plan for performing a function point analysis and indicates how projects, applications, and packaged software should be counted. Each of these requires their approach.

Clause 4 states general counting guidelines for a function point analysis.

Clauses 5, 6, 7, 8 and 9 successively give the definitions and guidelines used to identify function types and to determine the complexity of function types for internal logical files, external logical 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.

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 in this International Standard.

Annex C describes the mechanisms behind the increase in functional size.

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, the reader can look up what is important to him. For one reader, 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.

Software engineering — NESMA functional size measurement method — Definitions and counting guidelines for the application of function point analysis

1 Scope

1.1 Purpose

This International Standard specifies the set of definitions, rules and guidelines for applying the Nesma Function Point Analysis (FPA) method.

1.2 Conformity

This International Standard is conformant with all mandatory provisions of ISO/IEC 14143-1:2007.

1.3 Applicability

This International Standard can be applied to all functional domains.

1.4 Focus

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 budgets are established on the basis of this functional size (e.g. productivity standards and productivity attributes).

The figure below indicates what this International Standard *will* and *will not* cover.

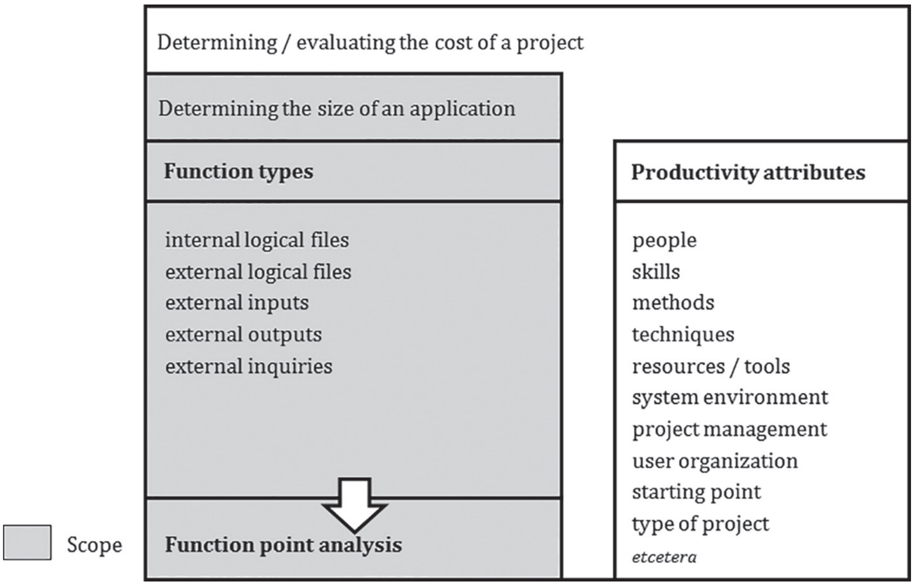

Figure 1 — Scope of the International Standard

2 Introduction to FPA

This clause gives a short description of FPA and explains a number of important concepts related to it. More specifically, subclause 2.1 provides a brief synopsis of FPA. Subclauses 2.2 through 2.4 distinguish between the different types of function point analyses. Subclauses 2.5 through 2.9 discuss each of the following successively within the context of FPA:

- The boundaries for an analysis
- Users
- Function types
- The complexity of a function type
- The valuing of function types

Subclause 2.10 defines the term *functional size* and describes how it is determined.

2.1 Brief description of FPA

2.1.1 Background, purpose and application of FPA

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

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 to 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.

2.1.2 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 to a *user* (see subclause 2.6). Because users see only the "outside" or the *boundary* (see subclause 2.5) 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 identifying data functions and transactional functions (see subclause 2.7).

Point

The *complexity* of a function type is determined according to certain standard guidelines (see subclause 2.8). Each function is worth a number of points, depending on its complexity (subclause 2.9). The sum of these points yields the functional size (see subclause 2.10).

Analysis

FPA is the analysis of an application or the analysis of the description/specification of an application in order to establish its functional size. The act of establishing the functional size of an application or project is therefore called *function point analysis*.

In order to be able to perform a function point analysis the following must first be determined:

- purpose of the function point analysis (subclause 2.2);
- scope of the analysis and boundaries of the application or project to be analyzed (subclause 2.5).

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

2.2 Use of FPA: *application* versus *project* functional size

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

- Determining the functional size of an application.
- Determining the functional size of a project.

Application functional size

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

Project functional size

is the number of function points that is a measure for 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 functional size is an essential parameter when determining the effort and schedule required for a project.

Determining the application functional size is elaborated on in subclause 3.5. Subclause 3.6 discusses the project functional size further.

2.3 Types of function point analyses

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

1 Indicative function point analysis

2 High level function point analysis (previously known as *Estimated*)

3 Detailed function point analysis

These function point analyses are explained further in subclause 3.2.

2.4 Function point analyses during a project

Function point analyses 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 analyses arises: the *initial function point analysis*, the *interim function point analysis*, and the *final function point analysis*. These analyses are discussed further in subclause 3.4.

2.5 Scope of the analysis and boundary of the application to be analyzed

The scope of the analysis is the set of functional requirements/specifications to be included in the function point analysis. When the scope has been determined, the boundary can be defined, the conceptual interface between the application and its users and/or other applications.

As indicated earlier in subclause 2.1, the scope of the analysis 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 perform a function point analysis.

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 subclause 2.2, a distinction is made between a function point analysis for an application and a function point analysis for a project. Subclause 3.5.1 provides guidelines for determining the application functional size and subclause 3.6.1 gives guidelines for determining the project functional size.

2.6 Users

FPA acknowledges three types of users:

- The people and/or organizations that use or are going to use the application to be measured. This category includes, amongst others, 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 analyzed.

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

2.7 Functions and function types

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

FPA defines *function types* 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.

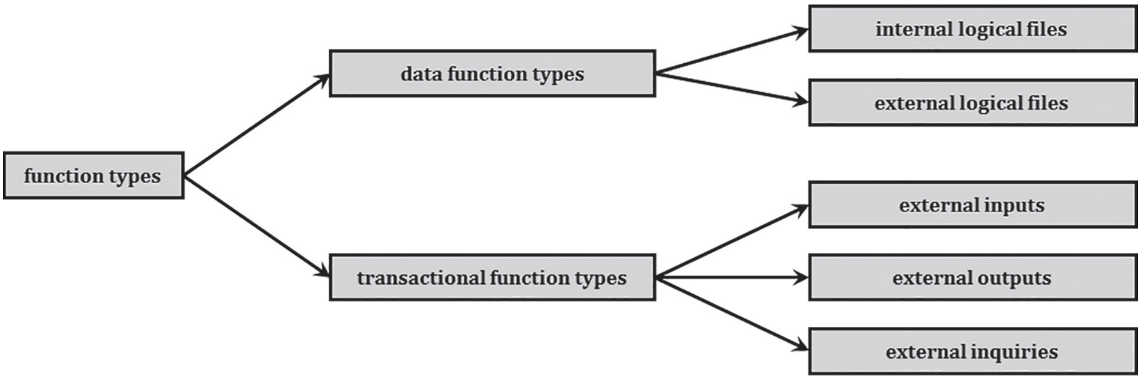

Figure 2 — Functions and function types

Function types are categorized into two main groups:

- Data function types
- Transactional function types

A *data function* is:

a logical group of data seen from the perspective of the user. FPA distinguishes between the following data function types:

- Internal logical files
- External logical files

A *transactional function* is:

an elementary process that meets the following criteria:

- the function has an autonomous meaning to the user and fully executes one complete processing of information, and
- after the function has been executed, the application is in a consistent state.

FPA distinguishes between the following transactional function types:

- External inputs
- External outputs
- External inquiries

Each function type is discussed in detail in clauses 5 through 9.

2.8 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 with regards to complexity.

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

The complexity tables that determine the levels of complexity are included in Annex A.

2.9 The valuing of functions

After the complexity of a function has been determined as described in clauses 5 through 9, the number of function points can be allocated to the function. This shall be done according to the rating in Table 1.

Table 1 — Function point table

Complexity	Function type				
	ILF	**ELF**	**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	EI	=	External input
ELF	=	External logical file	EO	=	External output
bold	=	value for high level FPA	EQ	=	External inquiry

High level specifications are enough to identify functions and their type when performing a *high level* function point analysis (see subclause 3.2.2), 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.

2.10 The functional size

The Number of function points: see Functional size functional size is the sum of the number of function points assigned to each of the functions (in the way described above) that lie within the scope of the object to be analyzed, that is the *application* or the *project*.

The functional size can also serve as a basis for preparing a project budget, by multiplying the number of function points with a productivity rate based on historical data (such as hours per function point). The preparation of a project budget is beyond the focus of this standard. More information on this subject can be found on the Nesma website nesma.org.

A Nesma FSM measurement result on the functional user requirements or specifications for a piece of software shall be labeled according to the following convention:

F(unction) P(oint) (ISO/IEC 24570:2018)

3 Guidelines to perform an FPA

This clause indicates how function point analysis shall be carried out. To this end, subclause 3.1 first presents a generally applicable step-by-step plan. Subclause 3.2 then indicates how to act when dealing with an indicative, high level, and detailed function point analysis. Subclause 3.3 goes into the role of the quality of specifications, while subclause 3.4 explains the use of FPA during a project. Subclauses 3.5 and 3.6 show how an application and a project functional size are determined in the event of development and in the event of enhancement, respectively. Subclause 3.7 introduces the definition of a functional change. Subclause 3.8 states what must be taken into consideration when dealing with the different ways of recording specifications. Subclause 3.9 concludes the clause with an illustration of how the different types of function point analyses can be applied during the life cycle of an application. For illustration purposes, this subclause will assume a generic application life cycle as phasing method.

3.1 Step-by-step plan to perform an FPA

Below follows a step-by-step plan to perform a function point analysis

Step 1: Collect the available documentation. The documentation that should be present for an indicative function point analysis, a high level function point analysis, and a detailed function point analysis is described in subclauses 3.2.1, 3.2.2 and 3.2.3 respectively.

Step 2: Determine the users of the application (see subclause 2.6).

Step 3: Establish whether an application function point analysis or a project function point analysis must be carried out. If an application function point analysis must be performed, follow the instructions stated in subclause 3.5. If a project function point analysis must be performed, follow the instructions in subclause 3.6.

Step 4: Determine from which other application(s) the application to be analyzed receives and/or uses data.

Step 5: Identify the functions and determine their type and complexity according to the guidelines described in clauses 5 through 9. When doing so, adhere to the sequence in which the clauses appear. Assign the number of function points using the function point table illustrated in subclause 2.9. This will result in the functional size.

Register the structure of the analysis and the number of function points. Particularly record any preconditions that have been used and assumptions that have been made.

Step 6: Together with the user(s), verify the result in relation to those aspects where specific interpretation of the available *specifications* was needed. If necessary, make any corrections as a result of that verification.

Step 7: Verify the result with an FPA expert in relation to those aspects where specific interpretation of the *counting guidelines* was needed. This may or may not be necessary. Make any corrections that are required as a result of that verification.

3.2 Types of function point analyses and their accuracy

Depending on the degree of detail of the specifications available, one of three types of function point analyses can be chosen: *an indicative, a high level, or a detailed function point analysis*. Each type of function point analysis mentioned in this subclause can be used both for the determination of the project size as for the determination of the application size. The minimum specifications required are different for each of the three types of function point analysis. In the subclauses below, the specifications required to perform each of the three types of analyses are stated. Each subclause, finally, will indicate when a particular type of analysis can be executed in the life cycle of an application.