BS ISO/IEC 24570:2018

BSI Standards Publication

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

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 24570:2018

National foreword

This British Standard is the UK implementation of ISO/IEC 24570:2018. It supersedes BS ISO/IEC 24570:2005, which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee IST/-/99, Participating IST members.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2018
Published by BSI Standards Limited 2018

ISBN 978 0 580 96247 9

ICS 35.080

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 28 February 2018.

Amendments/corrigenda issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 24570:2018

INTERNATIONAL STANDARD

ISO/IEC 24570

Second edition
2018-02-15

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)

Documents
Document Sharing Hub
© ISO/IEC 2018

Full document details: BS ISO/IEC 24570:2018

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2018, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

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

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 24570:2018

Contents

Page

Foreword ..... **v**

Introduction to this Standard ..... **vi**

1 Scope **1**
1.1 Purpose 1
1.2 Conformity 1
1.3 Applicability 1
1.4 Focus 1

2 Introduction to FPA **2**
2.1 Brief description of FPA 2
2.1.1 Background, purpose and application of FPA 2
2.1.2 Rationale behind FPA 2
2.2 Use of FPA: *application* versus *project* functional size 3
2.3 Types of function point analyses 3
2.4 Function point analyses during a project 3
2.5 Scope of the analysis and boundary of the application to be analyzed 4
2.6 Users 4
2.7 Functions and function types 4
2.8 The complexity of a function 5
2.9 The valuing of functions 6
2.10 The functional size 6

3 Guidelines to perform an FPA **7**
3.1 Step-by-step plan to perform an FPA 7
3.2 Types of function point analyses and their accuracy 7
3.2.1 Indicative function point analysis 8
3.2.2 High level function point analysis 9
3.2.3 Detailed function point analysis 9
3.3 Role of the quality of the specifications 10
3.4 FPA during a project 10
3.5 Determining the functional size of an application 11
3.5.1 Determining the application boundary 11
3.5.2 Functional size of new applications 12
3.5.3 Functional size of enhanced applications 12
3.5.4 Functional size of re-built applications 12
3.6 Determining the functional size of a project 13
3.6.1 Determining the scope of a project function point analysis 13
3.6.2 Functional size of development projects 14
3.6.3 Functional size of enhancement projects 15
3.6.4 The project function point analysis during the replacement of an application 15
3.7 Definition of functional change 16
3.7.1 General 16
3.7.2 Modification of a transactional function 16
3.7.3 Modification of a data function 16
3.7.4 Modification of a DET 16
3.8 FPA in specific situations 17
3.8.1 Analyzing on the basis of traditional design 17
3.8.2 Analyzing packaged software 17
3.8.3 Analyzing screens or windows 19
3.8.4 Analyzing when prototyping 20
3.9 Illustration: FPA and the application life cycle 21
3.9.1 FPA during the requirements phase 21
3.9.2 FPA during the analysis phase 22
3.9.3 FPA during the functional design phase 23
3.9.4 FPA during the construction phase 24

AVA Documents
Document Sharing Hub

© ISO/IEC 2018 – All rights reserved

Full document details: BS ISO/IEC 24570:2018

3.9.5 FPA during the implementation phase ... 24
3.9.6 FPA during the operation and maintenance phase ... 24

4 General FPA guidelines ... 25
4.1 Analyzing from a logical perspective ... 25
4.2 Applying the rules ... 25
4.3 No double counting ... 25
4.4 Built functionality, non-requested functionality ... 25
4.5 Production of re-usable code ... 26
4.6 Re-use of existing code ... 26
4.7 Screens, windows and reports ... 26
4.8 Input and output records ... 26
4.9 Security and authorization ... 26
4.10 Operating systems and utilities ... 27
4.11 Report generators and query facilities ... 27
4.12 Graphs ... 27
4.13 Help facilities ... 27
4.14 Messages ... 28
4.15 Menu structures ... 28
4.16 List functions ... 28
4.17 Browse and scroll functions ... 28
4.18 Cleanup functions ... 29
4.19 Completeness check on the function point analysis ... 29
4.20 FPA tables ... 29
4.21 Deriving logical files (data functions) from a normalized data model ... 30
4.21.1 Introduction ... 30
4.21.2 Denormalization rules ... 30
4.21.3 The nature of the relationship (cardinality and optionality) ... 31
4.21.4 Independence or dependence of an entity type ... 31
4.21.5 Conversion table: from normalized entity types to logical files ... 33
4.22 Shared use of data ... 33
4.23 Generic rule for counting data element types ... 36

5 Internal Logical Files ... 37
5.1 Definition of an internal logical file ... 37
5.2 Identifying internal logical files ... 38
5.3 Determining the complexity of internal logical files ... 39

6 External Logical Files ... 40
6.1 Definition of an external logical file ... 40
6.2 Identifying external logical files ... 41
6.3 Determining the complexity of external logical files ... 43

7 External Inputs ... 43
7.1 Definition of an external input ... 44
7.2 Identifying external inputs ... 45
7.3 Determining the complexity of external inputs ... 46

8 External Outputs ... 48
8.1 Definition of an external output ... 48
8.2 Identifying external outputs ... 50
8.3 Determining the complexity of external outputs ... 52

9 External Inquiries ... 53
9.1 Definition of an external inquiry ... 54
9.2 Identifying external inquiries ... 55
9.3 Determining the complexity of external inquiries ... 56

Annex A (normative) **Summary features for valuing function types ... 58**

Annex B (normative) **Function Point Analysis glossary ... 63**

Annex C (informative) **Increase in Functional Size ... 68**

AVA Documents
Document Sharing Hub

© ISO/IEC 2018 – All rights reserved

Full document details: BS ISO/IEC 24570:2018

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.

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 24570:2018

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.

© ISO/IEC 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 24570:2018

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.

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 24570:2018

This page deliberately left blank

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 24570:2018

INTERNATIONAL STANDARD ISO/IEC 24570:2018(E)

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.

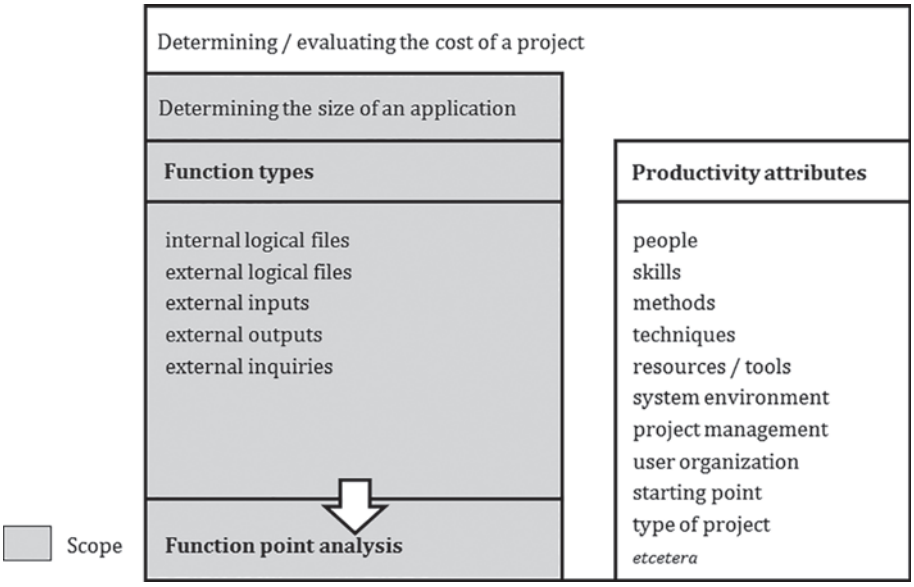

Figure 1 — Scope of the International Standard

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 24570:2018

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).

© ISO/IEC 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 24570:2018

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.

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 24570:2018

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.

© ISO/IEC 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 24570:2018

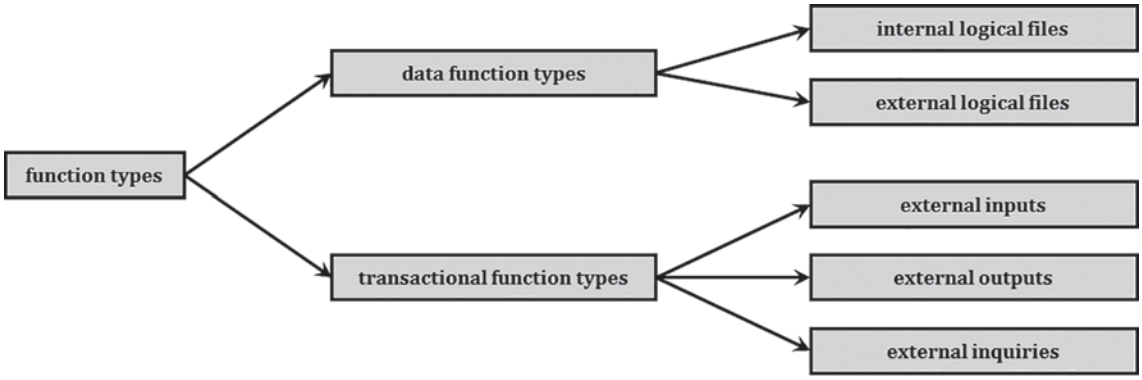

Figure 2 — Functions and function types

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.

© ISO/IEC 2018 – All rights reserved

AVA Documents Document Sharing Hub
Full document details: BS ISO/IEC 24570:2018

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
ELF = External logical file
bold = value for high level FPA
EI = External input
EO = External output
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)

© ISO/IEC 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS 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.

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 24570:2018

3.2.1 Indicative function point analysis

Definition

An indicative function point analysis indicates the size of an application or a project based on either a conceptual data model or a normalized data model. Caution is advised when using this indicative functional size, as deviations of up to 50% are possible.

When a conceptual model, or a model with a comparable level of detail, is available, the following formula indicates the functional size:

The number of internal logical files in the conceptual data model * 35
+
the number of external logical files in the conceptual data model * 15

Notice that the entities that are an FPA table (see subclause 4.20) *and* that are maintained by the application to be analyzed are counted together as one internal logical file. The same applies to the entities that are an FPA table type but which are maintained by another application, they are counted together as one external logical file. So, for all FPA tables together, at most two logical files are counted.

The factor 35 assumes that for each internal logical file three external inputs (add, change, delete), two external outputs, one external inquiry and some generic functionality will be present. Herewith a low complexity of the internal logical file is supposed (7 fp), and an average complexity of the transactions (3x4 + 2x5 + 4 = 26 fp) and 2 fp for the generic functionality.

The factor 15 is based on a low complexity of the external logical file (5 fp), an external output (5 fp) and an external inquiry (4 fp) and some generic functionality (1 fp) for each external logical file.

When a data model in third normal form, or a model with a comparable level of detail, is available, the following formula indicates the functional size:

The number of entity types maintained in the normalized data model * 25
+
the number of entity types referenced in the normalized data model * 10

Here, too, the entity types that are from the FPA table type (see subclause 4.20) *and* that are maintained by the application to be counted, are counted together as one entity type. The same applies to the entity types that are from the FPA table type but which are maintained by another application. For FPA tables, therefore, at most two entity types are counted.

Applicability

An indicative function point analysis can be carried out when a data model is available, from which data functions can be derived (see subclause 3.7). A data model can be presented in many ways, including Bachmann diagram, description in textual form, Entity Relationship Diagram (ERD) or UML class model.

This data model can be at a high level (e.g. at the end of the requirements phase in a waterfall approach or completion of the product backlog in an agile approach), or at a detailed level, later on in the application life cycle. High level here means that not all details are known, so the logical files are known, with the applications that maintain them, but without complete lists of attributes and without full details on their relations and validations.

Specifications required

The above shows that the following should be at your disposal for an indicative function point analysis:

— a conceptual or normalized data model of the application to be counted;

© ISO/IEC 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 24570:2018

— an indication as to how the logical files distinguished are maintained, either by the application to be counted or by another application.

3.2.2 High level function point analysis

In previous releases of this guideline, this type of analysis was named as *estimated.* Since this term could be associated with less accuracy, this term has been changed to high level, indicating that it can be used when not all details of the specification are known.

Definition

A high level function point analysis determines the number of functions for each function type (transactional function types and data function types) and uses a standard value for complexity: *Average* for transactional function types and *Low* for data function types.

These complexity values overrule all other complexity value rules for detailed function point analyses.

Applicability

A high level function point analysis can be carried out when a data model (see subclause 3.2.1) is available and an insight in the transactional functions (see subclause 2.7) that use it.

In general, in a waterfall approach this information becomes available at the end of the analysis phase and in an agile approach it is available when user stories can be related to transactional functions and data functions can be identified. This is when a sprint backlog (or sometimes, the product backlog) is ready for use.

Specifications required

The following specifications must be available for a high level analysis:

— a model that shows the logical files involved and their relations;

— an indication as to how the logical files distinguished are maintained: by the application to be counted or by another application;

— a model that shows the application functions with their incoming and outgoing information flows;

— the information flows between the functions of the application to be counted and its environment.

3.2.3 Detailed function point analysis

Definition

The detailed function point analysis is the most accurate analysis in which all the specifications needed for FPA are known in detail. This means that transactional functions have been specified up to the level of referenced logical files and data element types, and that logical files have been specified up to the level of record types and data element types. As a result, the complexity for each function identified can be established.

Applicability

A detailed function point analysis can be carried out when a detailed data model (see subclause 4.2.1) is available together with a detailed insight in the transactional functions that use it. All details on how the logical files within the data model are used by the transactional functions and all further functional details about them need to be known.

In general, in a waterfall approach this information becomes available during or at the end of functional design. In an agile approach this would be within a sprint, when corresponding user stories reach the definition of done.

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 24570:2018

Specifications required

The following specifications must be available for a detailed analysis:

— a model with all logical files and their relations (e.g. a Bachman diagram or an Entity Relationship Diagram);

— the record types and the data element types of the logical files;

— an indication as to how the logical files distinguished are maintained: by the application to be counted or by another application;

— a model that shows the application functions, their incoming and outgoing information flows, the logical files involved with the functions, and the supporting functions (help functions and so on);

— a detailed specification of the incoming and outgoing information flows of the application up to the level of data element types.

3.3 Role of the quality of the specifications

Regardless of the type of function point analysis (as described in subclause 3.2), sound functional specifications of the application are needed in order to perform a function point analysis. Depending on the method used, the form of the specifications produced can differ[2)]. The function point analysis, however, should continuously render the same result for the same application.

The reliability of a function point analysis depends directly on the quality of the specifications provided. Good-quality specifications will ensure that little effort is needed to translate specifications into function points and will result in a reliable count. Flawed or incomplete specifications can make it impossible to perform an FPA, or the result of the analysis may differ considerably from person to person as a result of various interpretations of the specifications.

3.4 FPA during a project

FPA plays a role during the entire course of a system development project. For instance, during the planning phase of a project, an *initial function point analysis* is carried out. As soon as change requests pertaining to specifications already recorded have been submitted during the project, *interim function point analyses* are carried out. When a project has been fully completed, the *final function point analysis* determines how big the supplied product is and what the functional size of the project has been. It does not matter which phases of an application's life cycle are being carried out in the project. Given a system life cycle as a phasing method, for example, the project can pertain to analysis, functional design, and/or construction. Regardless of the project's content, there will always be an initial and a final function point analysis, and sometimes one or more interim function point analyses.

An *initial function point analysis* is:

an analysis at the beginning of a project for developing an application or for enhancing an application (adding, changing, and deleting functionality) in which the functional size of an application (see subclause 3.5) and a project (see subclause 3.6) is recorded.

An initial function point analysis should be performed at the beginning of a project. Depending on the specifications available, this analysis is either an indicative, high level, or detailed function point analysis. The objective of the initial function point analysis is to draw up a budget for the project in terms of effort and schedule.

An *interim function point analysis* is:

an analysis during a new development project or an enhancement project in which the size of an interim enhancement (adding, changing, or deleting functionality) is determined. This means that the

2) Nesma has published a guide "FPA applied to UML / Use Case documentation" that gives guidance how the rules in this guideline can be applied from this type of documentation.

© ISO/IEC 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 24570:2018

effect that a change will have on both an application functional size (see subclause 3.5) and a project functional size (see subclause 3.6) is recorded.

An interim function point analysis should be done when functional specifications are changed. Depending on the specifications available, this analysis is either an indicative, high level, or detailed function point analysis. The objective of the interim function point analysis is to determine the effect of change requests on the price and delivery date agreed upon with the customer.

A *final function point analysis* is:

an analysis at the end of a new development project or an enhancement project (adding, changing, or deleting functionality) in which the final functional size of an application (see subclause 3.5) or the project (see subclause 3.6) are recorded.

A final function point analysis takes place at the end of a project. The project can pertain to the development of an application or to the realization of enhancements during an application's operation and maintenance phase. One objective of the final function point analysis is to determine the functional size of the *application* in function points. Another objective is to determine the size of the *project* so that productivity can be determined and, for example, a final monetary settlement for the project can take place, when a fixed price per function point has been agreed upon.

3.5 Determining the functional size of an application

As stated in subclause 2.2, FPA can be used to determine the application functional size or the project functional size. This subclause explains the use of FPA when determining the functional size of an *application*.

The purpose of performing analyses on applications is to determine the amount of functionality that is to be furnished to a user or that has already been furnished to him. This means that the functional specifications of an application must serve as the starting point and *not* the physical components such as programs or physical files.

Determining the functional size of an application can be carried out differently when developing an application than when enhancing an application. Still, in both cases, the application boundary must first be determined. This and the analyzing method employed for both development and enhancement are covered in greater depth in the subclauses below.

3.5.1 Determining the application boundary

An *application boundary* is:

the boundary between the (developing or developed) application and its environment (users and/or other applications).

An *application* in this International Standard refers to:

an automated information system. This is an application that collects, saves, processes, and presents data by means of a computer.

The following guidelines can help in determining the application boundary:

— The application demarcated by the application boundary should make up an independent whole that can function separately from other applications to a large degree.

— Establish whoever the owner or main user is. If there are several owners or key users, it often means that there are several applications.

— Look at the application through the eyes of the user, so only use the part of the application the user can actually observe. Use the specifications that describe or define the outside of the application, seen from the user perspective. This is called the application context, and it can be represented in a context diagram, among other ways. Determine what is located inside and outside the application.

© ISO/IEC 2018 – All rights reserved

AVA Documents
Document Sharing Hub
Full document details: BS ISO/IEC 24570:2018

Only those things that the user requests and that are relevant to him are significant to the function point analysis.

— Think of an application as a group of programs maintained as a whole. The application boundary encloses this group of programs. All functions within this boundary are identified.

3.5.2 Functional size of new applications

This pertains to the functional size of applications in the process of being built or that have already been built at the request of a user or user organization, and that provide a solution to the needs or wishes of the user or user organization.

Determining the functional size during the development of an application occurs as indicated in subclause 3.1. If the application in a development project is realized in a single project, determining the functional size of an *application* does not occur differently than when determining the functional size of a *project* (see subclause 3.6.2). Notice, however, that the size of any conversion software shall not be counted when the functional size of an application is determined.

If the application is being realized in the form of a number of sub-projects carried out in parallel, then the total functionality furnished by all the sub-projects will have to be examined in order to determine the functional size of an application. When examining these sub-projects, make sure that functionality appearing in more than one sub-project (such as a logical file) is not counted twice.

In the event of enhancement to an existing application (the adding, changing, or deleting of functionality), the functional size of the (changed) application should be determined as indicated in subclause 3.5.3.

3.5.3 Functional size of enhanced applications

This pertains to the functional size of an application after enhancement. In principle, enhancement can take place during each phase of an application's life cycle, but usually occurs during construction or during operation and maintenance. In the event of major enhancements, a separate project can be defined in which the functional size of the project is determined, in addition to the functional size of the application. In all cases, the functional size of the application is determined in the same way after the enhancement.

Below are the steps to be taken in order to determine the application functional size after enhancement:

Step 1: Determine the number of function points of the application before the change (*AFPB*).

Step 2: Identify which transactions and/or logical files are added to the application and establish how many function points they represent (*ADD*).

Step 3: Determine which transactions and/or logical files are deleted from the existing application and count how many function points they represent (*DEL*).

Step 4: Establish which transactions and/or logical files change. Then determine the number of function points that they represent before the change (*CHGB*) and after the change (*CHGA*).

Step 5: Determine the functional size of the application after the enhancement (*AFPA*) as follows:

$$\mathbf{AFPA = [AFPB + ADD\ DEL + (CHGA\ CHGB)]}$$

3.5.4 Functional size of re-built applications

If one application is replaced by another with the same functionality, then the functional size of the new application is equal to the old application it is replacing.

© ISO/IEC 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 24570:2018

If enhancements are the result of such a replacement, the functional size of the application can be determined in two ways:

— The replacement can be considered a new application. If this option is chosen, counting is done as described in subclause 3.5.2.

— The replacement can be considered an enhancement to the application being replaced. In this case, counting is carried out as indicated in subclause 3.5.3.

The result (the application functional size) of both methods is the same.

3.6 Determining the functional size of a project

As indicated in subclause 2.2, FPA can be used to determine an application functional size or a project functional size. This subclause explains the use of FPA to determine the functional size of a *project*.

Determining the functional size of a project (i.e. counting the number of function points of a project) differs in a number of ways from purely and simply calculating the functional size of an application, that is determining the amount of functionality provided or to be provided. This is because the effort required does not always result in an increase of functionality. Consider, for example, an effort to remove functionality, or to create a one-time functionality in order to convert data.

Using a number of project situations, the subclauses below spell out how the functional size of a project can be determined. Table 2 illustrates an example of how the functional size of both a project and an application are determined, as well as the differences between the two.

Table 2 — Functional size of a project versus an application

Size type	Lifecycle stage				
	Initial release	**Release 1**		**Release 2**	
ADD	1000	200		500	
DELETE		40		100	
CHANGE **Before**		80		220	
CHANGE **After**		100		200	
Project size	1000	ADD CHANGE After DELETE **TOTAL**	200 100 40 **340**	ADD CHANGE After DELETE **TOTAL**	500 200 100 **800**
Application size	1000	Initial release ADD CHANGE After CHANGE Before DELETE **TOTAL**	1000 +200 +100 -80 -40 **1180**	Release 1 ADD CHANGE After CHANGE Before DELETE **TOTAL**	1180 +500 +200 -220 -100 **1560**

Determining the functional size of a project when an application is developed can be carried out differently than when an application is enhanced. Still, in both cases, the application boundary must first be fixed. This and the counting method employed for both development and enhancement are covered in greater depth in the subclauses below.

3.6.1 Determining the scope of a project function point analysis

The *scope* of a project function point analysis is:

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 24570:2018

the set of functional requirements/specifications of a development project or an enhancement project to be included in a function point analysis. This may include one or more applications and therefore more than one application boundary may have to be determined as described in subclause 3.5.1.

This definition shows that two types of projects are distinguished:

— Development projects

— Enhancement projects

A *development project* is:

a project in which a completely new application is realized. In its execution, a development project can be split up into a number of sub-projects, each of which is responsible for a certain sub-system of the entire application. Each sub-project should then be considered an individual development project, if the sub-system itself is an application.

Re-building an existing application (re-engineering) is considered as development (see subclause 3.5.4).

FPA defines an *enhancement project* as:

a project in which enhancements are carried out on an existing application. This means that functionality can be added to, changed in, or deleted from an existing application (see subclause 3.7).

Guidelines to determine the scope of the project

The following guidelines can help in determining the scope of the project:

— Look at the application to be realized through the eyes of the user. The logical structures should be taken into consideration, not the physical structures.

— An application can be developed as a number of sub-projects executed more or less in parallel, each of which realizes a sub-system. The scope of such a sub-project therefore includes a sub-system. If the sub-systems must be able to exist independently (e.g. because of a phased implementation of the application or for functional reasons), then the exchange of data between the sub-systems is included in the functional size of each sub-project. The scope of the application contains all the sub-projects. This means that the interfaces between the sub-systems lie within the entire application boundary. The project functional size is the sum of the number of function points of the sub-projects and can be higher than the number of function points of the entire application (application functional size), because in this situation, an internal logical file of a particular sub-project (for example) is also counted as an internal logical file or as an external logical file for another sub-project that makes use of the same internal logical file.

— A practical way of figuring out whether a certain function lies within a boundary of an application requested by a user is to ask whether the user really wants to pay for this function.

— If in doubt, consult with the user if possible.

3.6.2 Functional size of development projects

A development project realizes an entirely new application. When a development project is split up into a number of sub-projects, then each sub-project must be treated as an independent development project when determining the functional size of a *project*.

The steps to be taken here are as follows:

Step 1: Determine the number of function points of each (sub)system to be realized.

Step 2: Count the number of function points of the conversion.

© ISO/IEC 2018 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 24570:2018

These function points contribute to the functional size of the project only. The conversion software required does not yield any additional functionality, but is only a one-time tool and therefore not a part of the application to be implemented.

Step 3: Determine the number of function points of the changes in other applications that are being realized in the project (see steps 1 through 5 in subclause 3.1).

These function points contribute to the functional size of the *project*. (The new functional size of the *applications* affected by the project must also be determined per application.)

Step 4: The functional size of the project is the sum of the number of function points recorded as a result of steps 1, 2, and 3.

Note: *A project functional size is used often to budget. When different applications are being modified/enhanced, make certain that budgeting is done per application because, for example, there may be different development environments and, therefore, different productivity rates.*

3.6.3 Functional size of enhancement projects

An enhancement project considers enhancements to one or more existing applications. This means that functionality can be added to, changed in, and deleted from these applications.

The steps required to determine the functional size of the project in function points are as follows[3)]:

Step 1: Identify which transactions and/or logical files are going to be added to the application(s) and establish how many function points they represent (*ADD*).

Step 2: Determine which transactions and/or logical files are going to be deleted from the existing application(s) and determine how many function points they represent (*DEL*).

Step 3: Establish which transactions and/or logical files change. Then determine the number of function points they represent after the change (*CHGA*).

Step 4: Calculate the functional size for the enhancement project as follows (*EFP*):

$$\mathbf{EFP = ADD + DEL + CHGA}$$

Step 5: If conversion software has to be made as a result of changes, determine the number of function points it entails and add it to the functional size of the project determined in step 4.

Note: *Existing internal logical files and external logical files that are used by the functions to be added, changed, or deleted, but are themselves not changed during an enhancement project, are not counted as internal logical files or as external logical files when the functional size of the project is being determined.*

3.6.4 The project function point analysis during the replacement of an application

It is often necessary to replace applications that have been operational for a longer period of time with applications that are more efficient and that meet the requirements of current-day information technology. This is done with re-engineering projects.

3) Nesma has published a guide "FPA for software enhancement" that deals with the use of functional sizing during enhancement. The method for sizing enhancement projects, described in that guide, is not a part of this International Standard and it results in a different unit of measurement.

© ISO/IEC 2018 – All rights reserved

AVA Documents Document Sharing Hub
Full document details: BS ISO/IEC 24570:2018