
**Information technology — Software
measurement — Functional size
measurement —**

**Part 1:
Definition of concepts**

iTeh STANDARD PREVIEW
(standards.iteh.ai)

*Technologies de l'information — Mesurage du logiciel — Mesurage de la
taille fonctionnelle —*

Partie 1: Définition des concepts

ISO/IEC 14143-1:1998

[https://standards.iteh.ai/catalog/standards/sist/c8cb79ed-6240-412a-9f36-
d05be3df4d0f/iso-iec-14143-1-1998](https://standards.iteh.ai/catalog/standards/sist/c8cb79ed-6240-412a-9f36-d05be3df4d0f/iso-iec-14143-1-1998)

Contents

	Page
Foreword	iii
Introduction	iv
1 Scope	1
2 Normative Reference	1
3 Definitions	1
4 Abbreviations	2
5 Characteristics and requirements	2
5.1 Characteristics	2
5.1.1 FSM Method characteristics	2
5.1.2 Base Functional Component characteristics	2
5.1.3 Functional Size characteristics	2
5.2 Requirements	3
5.2.1 FSM Method requirements	3
5.2.2 Base Functional Component assessment requirements	3
5.2.3 Designation of Functional Size	3
6 Process for applying an FSM Method	3
7 FSM Method labeling conventions	4
Annex A (informative) Uses of FSM	5

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 organizations 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. 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.

International Standard ISO/IEC 14143-1 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software engineering*.

ISO/IEC 14143 consists of the following part, under the general title *Information technology - Software measurement - Functional size measurement*:

Part 1: Definition of concepts

Further parts may follow.

Annex A of this part of ISO/IEC 14143 is for information only.

Introduction

Organizations engaged in software engineering have struggled for years in search of acceptable quantitative methods for measuring process efficiency and effectiveness, and for managing software costs, for the systems they acquire, develop, enhance or maintain. One critical, and particularly elusive, aspect of this measurement requirement has been the need to determine software size. Numerous software sizing methods have been proposed in the past. These included numbers of source lines of program code and various measures derived from the technical characteristics of the software.

These methods can have limitations in that they;

- cannot always be applied early in the software development process,
- cannot always be applied uniformly throughout the software's life time, or
- cannot always be meaningfully understood by users of the software.

The concepts of *Functional Size Measurement* (FSM) are designed to overcome these limitations by shifting the focus away from measuring how the software is implemented, to measuring size in terms of the functions required by the user.

Since the public release of Function Point Analysis, many sizing methods have been developed based on Albrecht's and other concepts. As these various sizing methods were developed without common agreement of the fundamental concepts of FSM, it was natural that inconsistencies amongst the methods would develop. These inconsistencies lessen the ability and attractiveness of any of these methods to be used as a standard method for the functional sizing of software.

This part of ISO/IEC 14143 defines the fundamental concepts of FSM, thereby promoting the consistent interpretation of FSM principles.

The text in this part of ISO/IEC 14143 has been formatted in order to facilitate the checking of a candidate software sizing method for conformance to this part of ISO/IEC 14143.

Information technology — Software measurement — Functional size measurement —

Part 1: Definition of concepts

1 Scope

This part of ISO/IEC 14143 defines the fundamental concepts of Functional Size Measurement (FSM) and describes the general principles for applying an FSM Method. This part of ISO/IEC 14143 does NOT provide detailed rules on how to:

- measure Functional Size of software using a particular method,
- use the results obtained from a particular method, or
- select a particular method.

NOTE - Guidelines on the uses of FSM are in Annex A.

This part of ISO/IEC 14143 is applicable when determining if a method for sizing software is an FSM Method. It does not prevent the development of various methods, but rather provides a basis for assessing whether a particular method conforms to FSM.

This part of ISO/IEC 14143 is intended for use by those persons associated with the acquisition, development, use, support, maintenance and audit of software.

2 Normative Reference

The following standard contains provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 14143. At the time of publication, the edition indicated was valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 14143 are encouraged to investigate the possibility of applying the most recent edition of the standard indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 9126:1991¹⁾, *Information technology - Software product evaluation - Quality characteristics and guidelines for their use.*

3 Definitions

For the purposes of this part of ISO/IEC 14143, the following definitions apply.

3.1

Base Functional Component (BFC)

an elementary unit of Functional User Requirements defined by and used by an FSM Method for measurement purposes

NOTE - Example, a Functional User Requirement could be "Maintain Customers" which may consist of the following BFCs: "Add a new customer", "Report Customer Purchases", and "Change Customer Details". Another example might include a collection of logically related business data maintained by the software under study such as "Customer Details". There are many other examples.

3.2

BFC Type

a defined category of BFCs

NOTE 1 - Examples of BFC Types are 'External Inputs', 'External Outputs' and 'Logical Transactions', and data stores such as 'Internal Logical Files'.

3.3

boundary

a conceptual interface between the software under study and its users

3.4

FSM Method

a specific implementation of FSM defined by a set of rules, which conforms to the mandatory features of this part of ISO/IEC 14143

3.5

Functional Domain

a class of software based on the characteristics of Functional User Requirements which are pertinent to FSM

3.6

Functional Size

a size of the software derived by quantifying the Functional User Requirements

¹⁾ Currently under revision.

3.7**Functional Size Measurement (FSM)**

the process of measuring Functional Size

3.8**Functional User Requirements**

a sub-set of the user requirements. The Functional User Requirements represent the user practices and procedures that the software must perform to fulfil the users' needs. They exclude Quality Requirements and any Technical Requirements

3.9**local customization**

an FSM Method that has been modified for local use, such that it might produce different Functional Sizes from those obtained prior to modification

3.10**Quality Requirements**

any requirements relating to software quality as defined in ISO 9126:1991

3.11**Scope of the FSM**

the set of Functional User Requirements to be included in a specific FSM instance

NOTE - The Scope of the FSM is determined by the purpose for measuring the software. For example, if an organization needs to know the size of its software portfolio, then the Scope of the FSM will include all the Functional User Requirements currently utilized. However, if a project manager is seeking to determine the size of a particular release of software, then the scope will include only those Functional User Requirements impacted by the project.

3.12**Technical Requirements**

requirements relating to the technology and environment, for the development, maintenance, support and execution of the software

NOTE - Examples of Technical Requirements include programming language, testing tools, operating systems, database technology and user interface technologies.

3.13**user**

any person that specifies Functional User Requirements and/or any person or thing that communicates or interacts with the software at any time

NOTE - Examples of 'thing' include, but are not limited to, software applications, animals, sensors, or other hardware.

4 Abbreviations

FSM Functional Size Measurement

BFC Base Functional Component

5 Characteristics and requirements**5.1 Characteristics****5.1.1 FSM Method characteristics**

5.1.1.1 An FSM Method shall have the following characteristics:

- a) it is based on a representation of the Functional User Requirements from the perspective of the users;
- b) it can be applied as soon as any Functional User Requirements have been defined and while they are available;
- c) it derives a Functional Size through the assessment (refer to 5.2.2) of Base Functional Components.

5.1.1.2 An FSM Method should be as independent as possible of particular software development methods or technologies.

NOTE - This will facilitate a broader use of the FSM Method.

5.1.2 Base Functional Component characteristics

A BFC shall have the following characteristics:

- a) it expresses only Functional User Requirements;
- b) it does not express Technical Requirements;
- c) it does not express Quality Requirements;
- d) it is classified as one, and only one BFC Type.

5.1.3 Functional Size characteristics

Functional Size shall have the following characteristics:

- a) it is not derived from the effort required to develop the software being measured;
- b) it is not derived from the effort required to support the software being measured;
- c) it is independent of the methods used to develop the software being measured;
- d) it is independent of the methods used to support the software being measured;
- e) it is independent of the physical components of the software being measured;
- f) it is independent of the technological components of the software being measured.

5.2 Requirements

5.2.1 FSM Method requirements

5.2.1.1 An FSM Method shall

- a) define the attributes of the BFCs,
- b) define the rules used to assess the BFCs,
- c) define the units in which Functional Size is expressed, and

NOTE - For example 'Function Points'

- d) describe the Functional Domain(s) to which the FSM Method can be applied.

5.2.1.2 An FSM Method should

- a) describe the kind of information necessary to enable the FSM Method to be applied,
- b) provide guidelines on how to document a specific FSM instance,
- c) describe the purposes for which the FSM Method can best be used such that the users of the FSM Method can judge its suitability for their purpose, and
- d) state its degree of convertibility to other sizing methods.

NOTE - Examples of «degree of convertibility» include, but are not limited to, the following:

- full convertibility - The Functional Size can be transformed to another software size measure using an algorithm or a mathematical model, under all conditions;
- restricted convertibility - The Functional Size can be transformed to another software size measure using an algorithm or mathematical model under some conditions. For example, for a limited range of sizes or within a specified degree of accuracy;
- no convertibility - The Functional Size cannot be transformed to another software size measure.

5.2.2 Base Functional Component assessment requirements

NOTE 1 - The order of presentation of the following BFC assessment requirements is at the discretion of the FSM Method.

An FSM Method shall

- a) define the BFC Types,
- b) describe how to identify which Functional User Requirements will be included within the Scope of the FSM,
- c) describe how to identify BFCs within the Functional

User Requirements,

- d) define how to classify BFCs into BFC Types, if there is more than one BFC Type,
- e) define how to assign a numeric value to a BFC according to its BFC Type,

NOTE 2 - An example of such a definition is: «an Output is a BFC Type which is assigned the value 5 if the BFC accesses less than two files and the value of 8 if more than two files are accessed».

- f) define the relationship, if any, between the BFC Type and the boundary, and

NOTE 3 - An example of a relationship of a BFC Type with the boundary is: "an Internal Logical File must reside on the software side of the boundary".

- g) define the relationships, if any between the BFC Types.

NOTE 4 - An example of a relationship between BFC Types is: "an Internal Logical File must be maintained by one or more External Inputs".

NOTE 5 - An example of a relationship between BFC Types is "each Logical Transaction comprises an Input, Process and Output component".

5.2.3 Designation of Functional Size

The FSM Method shall state the conventions to be adopted when reporting Functional Size such that it is qualified with

- a) the units of the FSM Method,
- b) the name of the FSM Method, and

NOTE 1 - Example: Functional Size = 300 Function Points (XYZ v2.0).

- c) an indicator that a local customization of a particular FSM Method has been used, where applicable.

NOTE 2 - Example: Local customization of version 2.0 of XYZ Method = XYZ v2.0c.

6 Process for applying an FSM Method

An FSM Method shall include the following activities in order to derive Functional Size:

- a) determine the Scope of the FSM;
- b) identify the Functional User Requirements within the Scope of the FSM;
- e) identify the BFCs within the Functional User Requirements;

- d) classify BFCs into BFC Types, if applicable;
- e) assign the appropriate numeric value to each BFC;
- f) calculate Functional Size.

7 FSM Method labeling conventions

An FSM Method shall

- a) use a name which will distinguish it from all other existing FSM Methods, and

NOTE 1 - Example: Method name = XYZ.

- b) where applicable, include a version number appended to the Method name, which will distinguish it from all other versions of the Method.

NOTE 2 - Example: Version 2.0 of XYZ Method = XYZ v2.0.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC 14143-1:1998
<https://standards.iteh.ai/catalog/standards/sist/c8cb79ed-6240-412a-9f36-d05be3df4d0f/iso-iec-14143-1-1998>

Annex A

(informative)

Uses of FSM

A.1 Introduction

The purpose of this annex is to introduce the reader to some uses of FSM Methods. It is neither intended to be a manual for the use of FSM Methods nor is it intended to be exhaustive.

The user requirements for software may include Quality Requirements and Technical Requirements (refer to section 3. Definitions), in addition to the Functional User Requirements. Some methods for sizing software consider these Quality and Technical Requirements through the use of one or more steps additional to those needed for Functional Size Measurement. These additional steps are not part of the process of applying an FSM Method (refer to clause 6), but may contribute to the practical use of Functional Size as described in the sections which follow.

NOTE - The adjustment via these additional steps is outside the scope of FSM because it measures features rather than functions that the user requires.

The uses of FSM Methods are organized into two parts: uses for Project Management and uses for Forecasting and Performance Management.

A.2 Project management

This portion of the uses of FSM Methods addresses how Functional Size could be applied to the management and control of projects.

A.2.1 Tracking the progress of a project

At an early point in a software project's lifecycle, FSM Methods can create an inventory of BFCs for the development or enhancement of software. The project manager can use this inventory to track and communicate the progress of the project, firstly, by tracking changes to the target set of BFCs (that is, by identifying BFCs added and deleted from the inventory due to scope changes) and secondly, by noting BFCs which have and have not been developed. The project's progress can then be communicated as the percentage of target BFCs which have passed a milestone or have been completed.

A.2.2 Managing scope change

At an early point in a software project's lifecycle, FSM Methods can determine the scope of the software by creating an inventory of BFCs agreed to by the users and the software supplier. For each change to this set of BFCs, Functional Size could be calculated and entered into an estimating model to forecast the effort and schedule impact. The impact could be used to negotiate modifications to the software scope and project plan.

A.2.3 Package functionality fit

FSM Methods can assist in expressing the fit of the functionality provided by a package to the functional requirements. A Functional Size could be calculated for the functional requirements. A Functional Size could also be calculated for the functional requirements satisfied by the package. The degree of fit could be expressed, using another Functional Size, as the proportion of the functional requirements satisfied.

A.3 Forecasting and performance management

This portion of the uses of FSM Methods addresses how Functional Size could be applied to forecasting resource usage and the management of performance. This section typically involves the use of Functional Size as a normalizing factor and the collection of a large amount of data to create models.

A.3.1 Accounting for an organization's software asset

A Functional Size could be calculated for part or all of an organization's application portfolio and entered into an estimating model to determine the total software asset or the total cost of replacement or re-engineering.

A.3.2 Productivity management

An FSM Method can assist with managing the productivity of software development, enhancement, and maintenance processes. Productivity indicators (Functional Size divided by effort, schedule or cost) and demographic characteristics could be analyzed to determine which demographic characteristics have the greatest impact on productivity. Demographic characteristics are environmental, project, and/or staff