INTERNATIONAL STANDARD

ISO/IEC 29881

First edition
2008-05-15

Information technology — Software and systems engineering — FiSMA 1.1 functional size measurement method

Technologies de l'information — Logiciel et systèmes d'ingénierie — Méthode de mesure de la taille fonctionnelle FiSMA 1.1

Reference number
ISO/IEC 29881:2008(E)

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

Published in Switzerland

Contents

Page

Foreword

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

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

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

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

ISO/IEC 29881 was prepared by the Finnish Software Measurement Association (FiSMA) 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.

Introduction

Functional size is an essential measure for comparisons of software development activities and development alternatives. Beside its uses in estimating and productivity analysis, functional size has proven to be useful in project planning, tracking, control and contracting. Because Functional Size Measurement (FSM) works best when there is a complete list of functional user requirements and services, it makes scope management and change management effective, reliable and relatively easy to understand even to the end-user.

The correctness of counting parameters and thus the usefulness of an FSM method can be evaluated based on the correlation between functional size and effort under similar environmental and technical circumstances and quality requirements. This kind of evaluation may indicate a need to justify the counting parameters used to derive functional size. FiSMA Functional Size Measurement Method Version 1.1 (referred to throughout this International Standard as simply FiSMA 1.1) is a general, parameterised functional size measurement method for all types of software. It was developed by a working group of Finnish Software Measurement Association (FiSMA), to replace the previous FSM method Experience 2.0 Function Point Analysis (FPA), which has been applied largely in Finland since 1997. More than 600 software development projects were measured using that method between 1997 and 2003.

The current values of constraints used in FiSMA 1.1 are derived from its predecessor Experience 2.0 FPA, and were confirmed statistically to be correct. They may be updated in future releases of the FiSMA FSM Method if the data collection and analysis demonstrate the need to do so.

For readers who are unfamiliar with Functional Size Measurement terminology, a review of terms is provided in Annex A, together with definitions and explanations of the most important terms.

Results from FiSMA 1.1 and Experience 2.0 FPA are largely convertible with each other, if the source data has been collected at the recommended detail level.

FiSMA 1.1 is based purely on Functional User Requirements (FUR). User requirements can be thought of as functional – what the software does, and non-functional – how the software must perform (including quality requirements). For FiSMA 1.1, the Functional User Requirements are the object of measurement. While some FSM methods are process oriented, FiSMA 1.1 is service oriented. Process oriented methods require the identification of all functional processes supported by the piece of software. In contrast, service oriented methods, such as FiSMA 1.1, require identification of all different *services* provided by the piece of software.

The FiSMA 1.1 relationship chain between users and the developed piece of software involves user needs and services as presented in Figure 1.

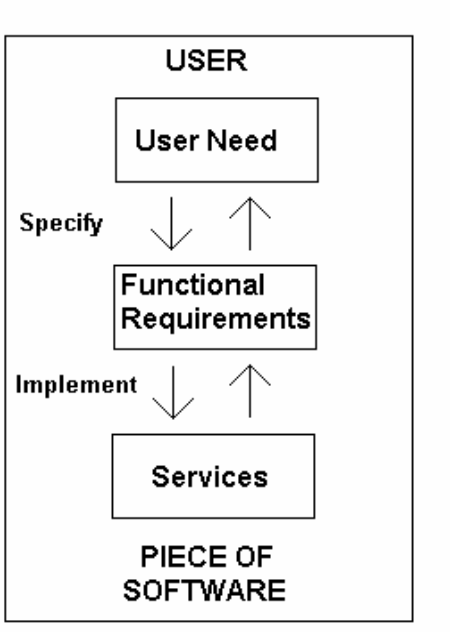

Figure 1 — Links between user and a piece of software

While each audience may have their own reasons for size measurement, the typical user viewpoint is to estimate the effort for a software project. Other important industry uses of FSM are presented in Figure 2.

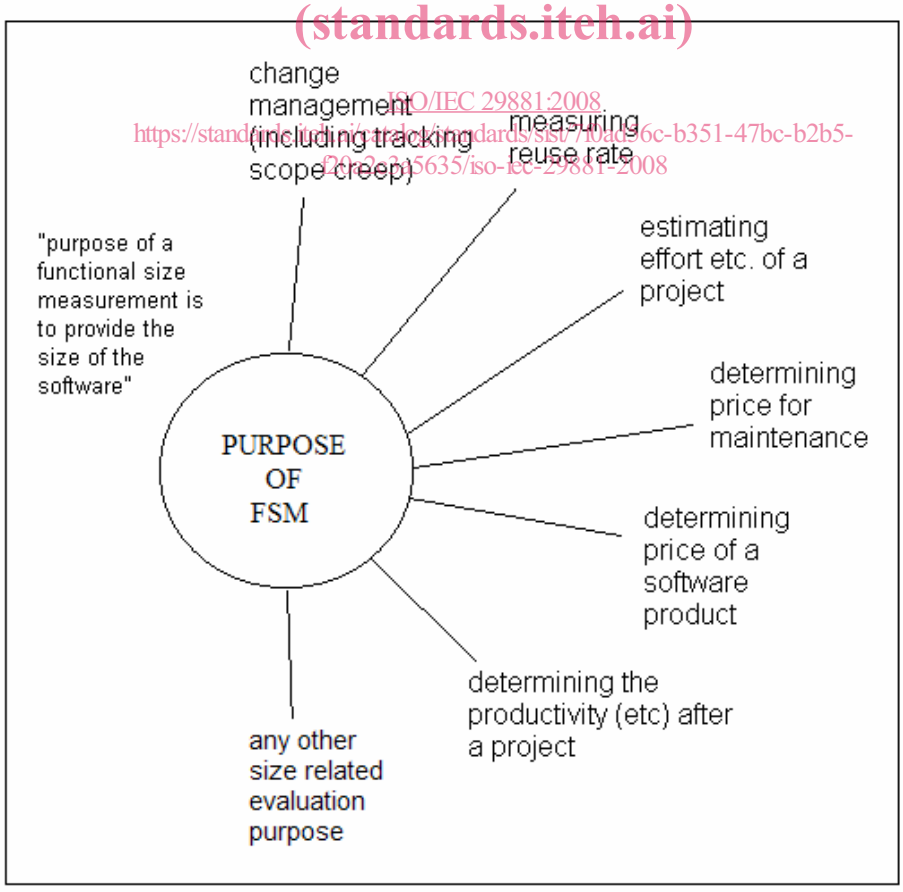

Figure 2 — Common Purposes of Functional Size Measurement

Information technology — Software and systems engineering — FiSMA 1.1 functional size measurement method

1 Scope

This International Standard specifies the set of definitions, conventions and activities of FiSMA 1.1.

The target audience of this International Standard includes anyone who applies FiSMA 1.1 to measure the functional size of a piece of software. FiSMA 1.1 is intended for use by those persons associated with the acquisition, development, use, support, maintenance, and audit of software. FiSMA 1.1 is based on an assessment of the Functional User Requirements. It measures the functional size of a piece of software from the perspective of the users.

1.1 Field of application for FiSMA 1.1

FiSMA 1.1 is applicable to measure all software in any functional domain.

1.2 Limitations of FiSMA 1.1

FiSMA 1.1 has no limitations related to the type or quality of software to be measured.

1.3 Scope of FSM for FiSMA 1.1

The scope of the Functional Size Measurement for FiSMA 1.1 is determined by the purpose for measuring the software. When using FiSMA 1.1, the set of FUR to be included depends on the purpose of the count and thus, may include the FUR for one piece of software or a set of pieces of software. Each piece of software within the scope is measured separately and if more than one piece of software is included within a project, all of the functional sizes may be added together. The scope of the FSM instance is always a subset of the overall user requirements and includes purely the Functional User Requirements, in other words, “what” in terms of services and tasks that the software must perform. The purpose of the FSM determines which FUR will be included in the FSM instance.

NOTE 1 For example if the purpose for the FSM is to determine the size of the first release of a piece of software, then the size using FiSMA 1.1 will include only the FUR for the first release of the software.

NOTE 2 As another example,if the purpose for the FSM is to determine the supported size of an installed package, only those functional user requirements in the package that are used by the organization will be included in the instance of the FSM.

NOTE 3 FiSMA 1.1 only measures the size of the Functional User Requirements included within the scope as outlined above.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 14143-1:2007, *Information technology — Software measurement — Functional size measurement — Part 1: Definition of concepts*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply. Whenever a term is already defined by ISO/IEC, such as “Functional Size Measurement”, the ISO definition has been adopted for this method.

3.1
BFC class
defined group of BFC types

3.2
boundary
conceptual interface between the software under study and its users

[ISO/IEC 14143-1:2007, definition 3.3]

NOTE The boundary of a piece of software to be sized using FiSMA 1.1 conceptually separates the piece and the environment in which it operates, perceived from the external user perspective. The boundary provides the measurement analyst(s) with a solid delimiter to distinguish, without ambiguity, what is included inside the measured software from what is part of the measured software’s operating environment.

3.3
data element
unique, user recognizable, non-repeated field in a BFC

NOTE 1 A data element can be a character string, or a digital or graphical element in a BFC.

NOTE 2 When “data elements” are indicated for a BFC, the number of data elements is always greater than 0.

3.4
data store
organized and persistent collection of data and information that allows for its retrieval

[ISO/IEC 15939:2002]

3.5
end-user
any person that communicates or interacts with the software at any time

3.6
Functional Services
base functional components (BFC) defined by FiSMA 1.1

3.7
operation
arithmetic or logical operation performed in an algorithmic and manipulation BFC

NOTE The number of operations is always greater than 0.

3.8
reading reference
data storage entity or record, or interface record from another software or system containing data retrieved in a BFC

NOTE The number of reading references is equal to 0 for all BFC types where it is applicable.

3.9
user
any person or thing that communicates or interacts with the software at any time

NOTE For readers who are unfamiliar with Functional Size Measurement terminology, and to increase the readability of this International Standard, a review of terms is provided in Annex A, together with definitions and explanations of the most important terms.

3.10
writing reference
data storage entity or other record, or interface record to another software or system to which data is written in a BFC

NOTE The number of writing references is greater than 0 with all BFC types where it is applicable.

4 BFC Classes and BFC Types of FiSMA 1.1

FiSMA 1.1 identifies seven distinct BFC classes:

- Interactive end-user navigation and query services (q)
- Interactive end-user input services (i)
- Non-interactive end-user output services (o)
- Interface services to other application (t)
- Interface services from other applications (f)
- Data storage services (d)
- Algorithmic and manipulation services (a)

Each BFC class of FiSMA 1.1 further decomposes into several BFC types. All together there are 28 BFC types. Figure 3 shows the relationships between the BFC classes and their component BFC types. Each BFC Class is explained in the clauses that follow.

NOTE For ease of presentation, the following short form conventions have been used:

- Each of the seven BFC classes is denoted by a single alphabetic character as shown in Figure 3;
- Each BFC type is prefixed by its BFC class alphabetic character and an integer number that has been assigned to it as denoted in Figure 3.

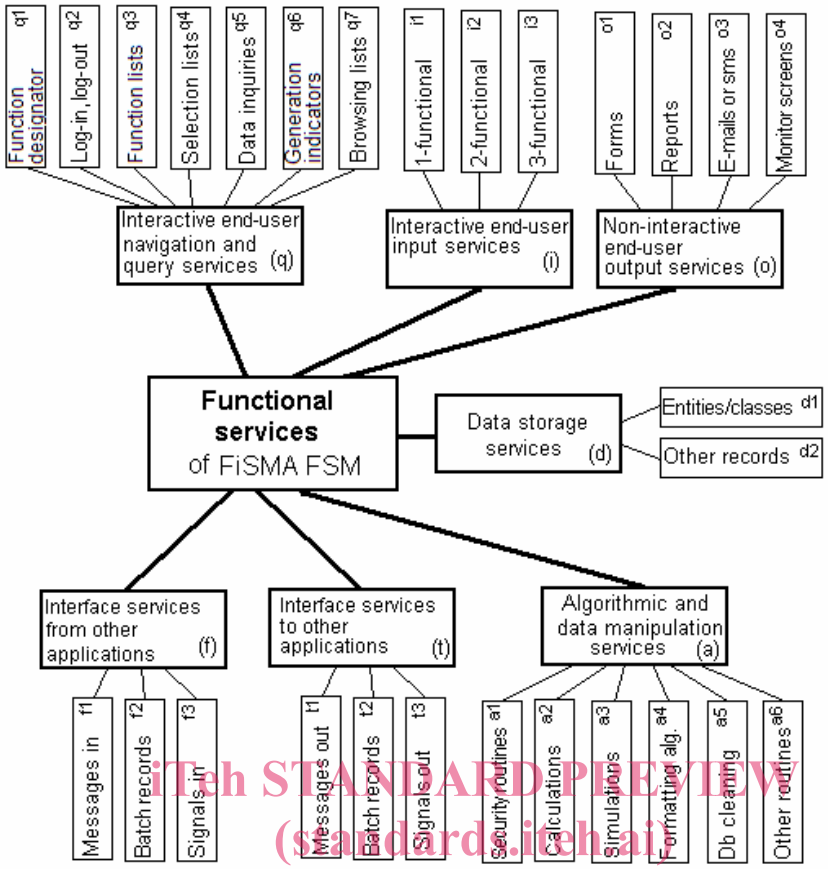

Figure 3 — FiSMA 1.1 BFC classes and BFC types

4.1 Interactive end-user navigation and query services (q)

This class of BFC involves data and/or services crossing the boundary into or out of the software. Interactive end-user navigation and query services specify all parts of the interactive user interface where there is no maintenance of persistent data stored in the system. Maintenance refers to any service where data is changed as a result of the service and includes, for example, creating, updating or deleting.

The number of functional size units for each navigation and query service depends on the number of data elements of the BFC and the number of unique entities that need to be referenced. (There is an indirect relationship between the entities identified in this step as being referenced and the BFC types identified within the BFC Class called data storage services. Each independent entity identified as a reference in this BFC type must also be explicitly counted once in the software application stored data.)

In FiSMA 1.1, the BFC class “Navigation and query services” is divided into seven BFC types:

- **Function designator** (q1) is an object that inititates a service or the piece of software. They are made for starting the application or opening a new functional part of it. Usually there is not very much information in an icon. With a function designator the end-user will not update or change contents of stored data.

  NOTE End users may refer to function designators as “Icons” however, this does not imply any particular design. Function designators are an important part of interactive end-user navigation and query services, especially in the case of graphical user interface (GUI).

- **Log-in and log-out functions** (q2) are parts of interactive end-user navigation and query services. Usually this BFC type does not update persistent data. They control users access and prevent illegal use. The end-user may change temporary user settings from the log-in window. With a log-in or log-out service, the end-user will not update or change contents of stored data.

- **Function list** (q3) are common parts of interactive end-user navigation and query services in user interfaces. They are made for selecting the next operation. With a function list the end-user will not update or change the content of stored data.

  NOTE End users may refer to these as “menus”, however, this does not imply any particular design.

- **Selection lists** (q4) are parts of interactive end-user navigation and query services. They show a list of acceptable parameter values to the end-user. Often they are very simple, showing values of one single data item, but they may be more complicated. With a selection list the end-user will not directly update or change contents of stored data, though selection lists may be used in connection with interactive input services.

  NOTE There are many different ways to implement selection lists in practice, but there is no design implied. In practice end users will refer to these functions as “drop-down lists”, “pop-up windows”, “combo boxes”, “list boxes”, etc.

- **Data inquiries** (q5) are parts of interactive end-user navigation and query services. They show the content of stored data to the end-user. With an inquiry, the end-user will not update or change contents of stored data .

  NOTE Inquiries are also called enquiries or queries.

- **Generation indicators** (q6) are parts of interactive end-user navigation and query services. Very often they are connected to some other type of functional services, such as a report or manipulation routine. With a generation indicator, the end-user may initiate the production of a report without any automatic timing controls. With a generation indicator the end-user will not directly update or change contents of stored data, although the connected routine may do it.

  NOTE End users may refer to generation indicators as “generation dialogs”, however, this does not imply any particular design.

- **Browsing lists** (q7) are parts of interactive end-user navigation and query services. They show a list of similar data elements, typically the most important details to help filter the entities for further operations. With a browsing list, the end-user will not directly update or change contents of data storage.

4.2 Interactive end-user input services (i)

This class of BFC involves data and/or services crossing the boundary into the software. Interactive end-user input services specify all parts of the interactive user interface where there is maintenance of data store(s) of the software. Data storage consists of logical entities (data records). Maintenance refers to any service where data is changed as a result of the service, and includes, for example, creating, updating and deleting.

From a user’s point of view, interactive end-user services perform those business tasks which change the data contents of the software. From the information system point of view end-users manipulate system data using interactive end-user services.

The number of functional size units of input functions depends on the number of different data elements of the BFC measured, and the number of needed reading and writing references to unique entities.

(There is an direct relationship between the entities identified in this step as writing references and the BFC types identified within the BFC Class: data storage services. Each independent entity identified as a writing reference in this BFC type must also be explicitly counted once as stored data.)

In FiSMA 1.1, end-user input services are divided into three BFC types:

- **1-functional input dialogs** (i1) support only one of the three maintenance types create, update or delete.
- **2-functional input dialogs** (i2) support two of the three maintenance types create, update and/or delete.
- **3-functional input dialogs** (i3) support all three maintenance types create, update and delete.