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/IEC 2008

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2008

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

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

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

© ISO/IEC 2008 – All rights reserved

Not for Resale

Contents

Page

Foreword iv

Introduction v

1 Scope 1
1.1 Field of application for FiSMA 1.1 1
1.2 Limitations of FiSMA 1.1 1
1.3 Scope of FSM for FiSMA 1.1 1

2 Normative references 2

3 Terms and definitions 2

4 BFC Classes and BFC Types of FiSMA 1.1 3
4.1 Interactive end-user navigation and query services (q) 4
4.2 Interactive end-user input services (i) 5
4.3 Non-interactive end-user output services (o) 6
4.4 Interface services to other applications (t) 6
4.5 Interface services from other applications (f) 6
4.6 Data storage services (d) 7
4.7 Algorithmic and manipulation services (a) 8

5 FiSMA 1.1 Measurement process 8

6 Counting rules for each BFC type class 10
6.1 Interactive end-user navigation and query BFC's (q) 10
6.2 Interactive end-user input BFC's (i) 10
6.3 Non-interactive end-user output BFC's (o) 10
6.4 Interface BFC's to other applications (t) 11
6.5 Interface BFC's from other applications (f) 11
6.6 Data storage services (d) 11
6.7 Algorithmic and manipulation services (a) 12

7 Functional size measurement unit 12

8 Calculation of the FiSMA 1.1 functional size of a piece of software 12

9 Measurement reporting 12

10 Convertibility from FiSMA 1.1 to other FSM Methods 13

Annex A Glossary of terms relevant to FiSMA 1.1 14

© ISO/IEC 2008 – All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

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.

--`,,```,,,,````-`-`,,`,,`,`,,`---
© ISO/IEC 2008 – All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

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.

© ISO/IEC 2008 – All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

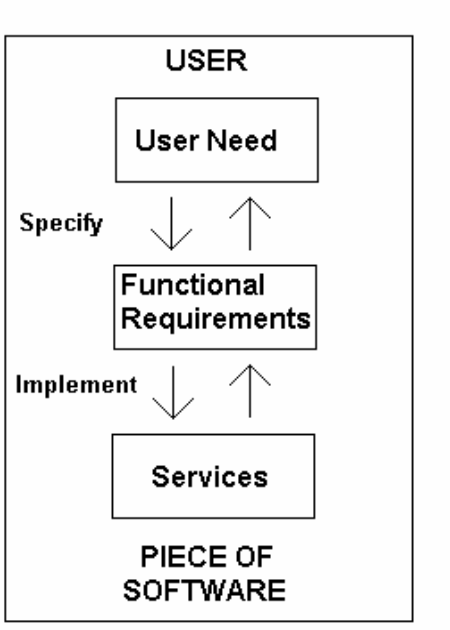


Figure 1 — Links between user and a piece of software

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.

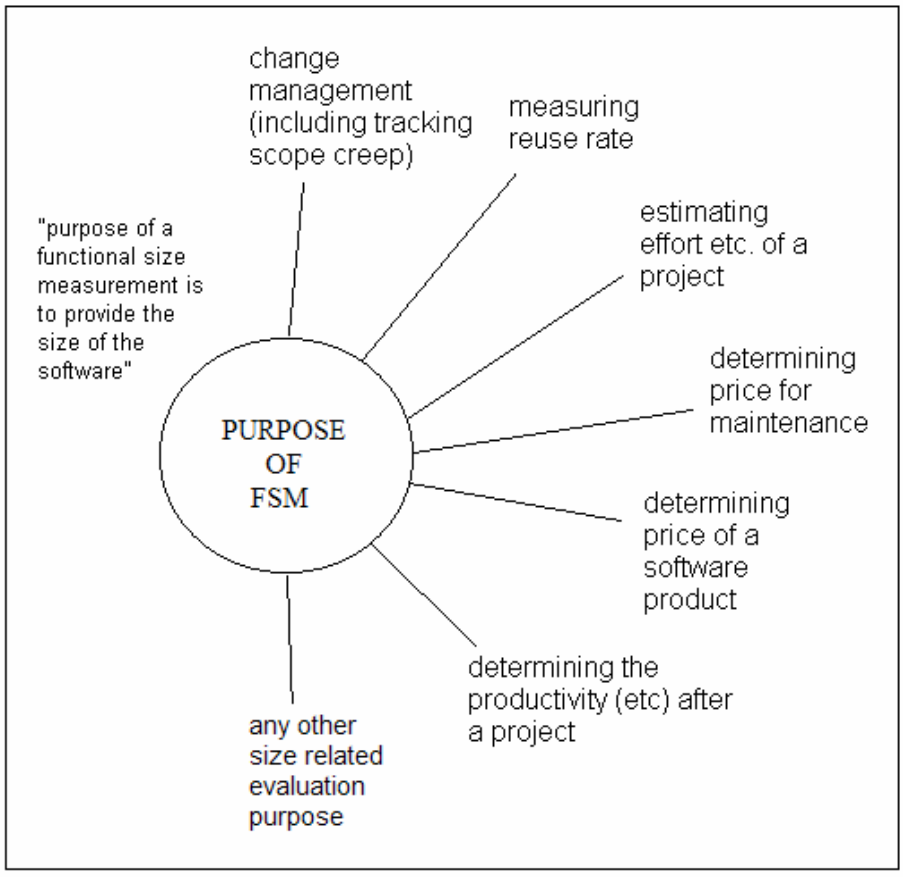


Figure 2 — Common Purposes of Functional Size Measurement

© ISO/IEC 2008 – All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

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.

© ISO/IEC 2008 – All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

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.

© ISO/IEC 2008 – All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

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.

© ISO/IEC 2008 – All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

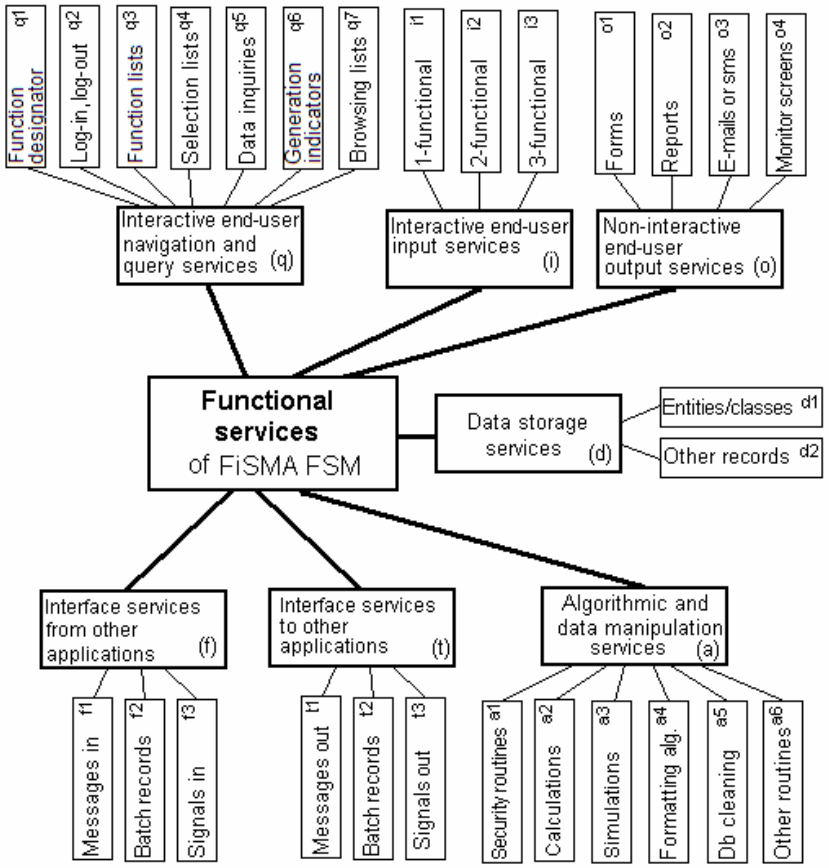


Figure 3 — FiSMA 1.1 BFC classes and BFC types

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.

© ISO/IEC 2008 – All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

- **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.

--`,,```,,,,````-`-`,,`,,`,`,,`---

© ISO/IEC 2008 – All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

4.3 Non-interactive end-user output services (o)

This class of BFC involves data and/or services crossing the boundary out of the software. Non-interactive end-user output services specify all parts of the user interface which are non-interactive and do not maintain data store(s) of the software.

The number of functional size units of output functions depends on the number of different data elements of the BFC and the number of needed reading references to entities.

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

FiSMA 1.1 output services are divided into four BFC types :

- **Output forms** (o1) are services resulting in printed or displayed documents, which always present the same layout (e.g a receipt).
- **Reports** (o2) are services resulting in printed or displayed documents, whose layout may vary within the specified framework according to the presented data (e.g. product list or sales report).
- **E-mails and text messages** (o3) are services resulting in electronically transmitted output documents, which have a standardised structure. The structure often contains title fields, data fields and optional attachments.
- **Monitor screen output** (o4) service involves continuously displayed documents, which are updated regularly in consequence of data changes (e.g. measurement display of a process).

4.4 Interface services to other applications (t)

This class of BFC involves data and/or services crossing the boundary out of the software. Interface services to other applications specify all automatic data transfers that move data from the measured piece of software to another application or any device.

The number of functional size units of outbound interface functions depends on the number of different data elements of the BFC measured (i.e. the number of attributes) and the number of needed reading references to entities.

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

FiSMA 1.1 outbound interface functions are divided into three BFC types:

- **Messages to other applications** (t1) are services where data groups are sent on-line, usually in real-time, to any other application.
- **Batch records to other applications** (t2) are services where data groups are written to a temporary file for transfer to another application.
- **Signals to devices or other applications** (t3) are services where data strings or single pieces of information are sent to any other application or device (e.g., a LED).

4.5 Interface services from other applications (f)

This class of BFC involves data and/or services crossing the boundary into the software. Interface services from other applications specify all automatic data transfers that receive data groups that are provided and sent by another application or any device.

© ISO/IEC 2008 – All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

The number of functional size units of inbound interface services from other applications depends on the number of different data elements of the BFC measured, and the number of reading and writing references to entities.

(There is an indirect relationship between the entities identified in this step as being referenced 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.)

FiSMA 1.1 divides this BFC class into three BFC types:

- **Messages from other applications** (f1) are services where data are received on-line, usually in real-time from any other application.
- **Batch records from other applications** (f2) are services where data are received in groups or "batches" from any other application.
- **Signals from devices or other applications** (f3) are services where data strings or single pieces of information are received from any other application or device (e.g., a sensor).

4.6 Data storage services (d)

This class of BFC involves data storage associated with data crossing the boundary by means of another BFC class into the software.

Data storage services specify a group or collection of related and self-contained data in the real world, about which the user requires the software to provide one or more data stores. Data storage services are functional services provided by the piece of software to satisfy these data storage requirements. These "groups or collections of related and self-contained data" are often called entities, data groups, data classes or objects of interest, depending on the terminology used in the development environment.

Data storage services result in data stores and make data available for maintenance, inquiry, or output.

NOTE Data storage services are typically implemented as tables in relational databases, or as records in data files in general.

The number of functional size units of data storage services depends on the number of different data elements (i.e. the number of attributes related together) in the self-contained group or collection.

In this FSM method data storage services are divided into two BFC types:

- **Entities or classes** (d1) are data storage services resulting in one or more unique data stores representing fundamental things of relevance to the user, and about which persistent information is stored.

  NOTE 1 Entities are typically defined during the early phases of the application development, feasibility study or requirements specification phase. They are like the common vocabulary of the system representing things that are known by the users and which are important to them (for example customer, order, supplier).

  NOTE 2 You should look for entities in ER-models, which have been transformed into the third normal form (3NF), i.e. for example many-to-many relationships have been removed from the model. In object-oriented models you describe classes and you can find them from the class diagrams.

- **Other record types** (d2) are the other type of data storage services and result in one or more unique data stores besides that which is counted as entities or classes.

  NOTE Other record types services result in the tables and logical records, which were not counted as entities or classes. These may include many kinds of records and data storage (e.g.,legally required log records and tax tables). The functional user requirements for these services are often discovered or articulated later in the software development work.

--`,,```,,,,````-`-`,,`,,`,`,,`---

© ISO/IEC 2008 – All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

4.7 Algorithmic and manipulation services (a)

This class of BFC involves data and/or services performed by the software to independently transform data that may or may not cross the boundary Algorithmic and manipulation services are user-defined, independent data manipulation functions. While these functions are often associated with another type of BFC, they are separate services and require their own specification from the user. Independence means that the functionality of the service is extraneous to the service provided by any other BFC type. An algorithmic and manipulation service may consist of arithmetic and/or logical operations. Simple operations that are associated with another type of BFC, but do not require any additional specification from the user, shall not be considered as algorithmic or manipulation services.

The number of functional size units of algorithmic and manipulation services depends on the number of different operations performed and the number of different variables needed to perform the service.

In this FSM method algorithmic and manipulation services are divided into six BFC types:

- **Security routines** (a1) are algorithmic services providing security features such as encryption, decryption, advanced authorization, etc.
- **Calculation routines** (a2) are algorithmic services providing arithmetic or logical counting services.
- **Simulation routines** (a3) are algorithmic services providing simulative calculating services.
- **Formatting routines** (a4) are manipulation services providing special format conversion services (i.e. beyond typical, simple editing).

  NOTE An example of a formatting routine could be changing table rows into graphics.

- **Database cleaning routines** (a5) are manipulation services supporting data storage maintenance, such as removing unnecessary records and combining or cumulating data elements based on user-defined rules.

  NOTE These routines are often scheduled and performed in batch mode.

- **Other manipulation routines** (a6) include all independent user-defined algorithmic and manipulation services, which are not counted as any other algorithmic and manipulation BFC type functions.

5 FiSMA 1.1 Measurement process

The FiSMA 1.1 measurement process consists of the following steps:

1. Gather documentation and software development artifacts to describe the functional user requirements for the software (to be or already)_ developed. These include any items such as use cases, preliminary user requirements, use manuals, entity relationship diagrams, screen, report or database mockups, data flow diagrams, etc. – anything that describe what the software will do in terms of tasks or services, independently of any quality or technical requirements.
2. Determine the Scope of the FSM: The Scope of FiSMA 1.1 is determined by the purpose for doing the FSM and includes the FUR to be developed or enhanced in the project or application to be counted.
3. Determine which are the Functional User Requirements to be measured by FiSMA 1.1 by determining the Scope as outlined in 1 and include only those user requirements that describe what the software is to do in terms of tasks and services.
4. Identify the BFCs within the Functional User Requirements from 2. above in two main parts: 1. measuring the end-user interface services, and 2. measuring indirect services. If one of these two parts does not exist for the piece of software, then the process consists only of measuring the services that are present.
5. Classify the BFC's into the appropriate BFC type by mapping each BFC identified to the descriptions of the BFC types in Clause 4. Be cautious to identify duplicate logical functionality so that it is counted only

--`,,```,,,,````-`-`,,`,,`,`,,`---

© ISO/IEC 2008 – All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

once per instrance of the FSM. Two BFC types are considered to be duplicate if they have the same characteristics (i.e., identical BFC types with the same values for each of the component parts for the BFC type i.e., identical data elements, reading references, and/or writing references as appropriate for the BFC type.)

6. Assign the appropriate numeric value to each BFC using the calculations outlined for each BFC type in Clause 7.

7. Calculate the Functional Size as outlined in Clause 8.

Figure 4 presents the process overview for steps 3-6. There is one cell for counting the functional size units for each of the 28 BFC types. To complete any cell follows the same procedure of asking three questions:

1. How many of these BFC types are present in this piece of software?
2. What are they? Identify them all.
3. What are they like? Give the numbers of details for each BFC.

Detailed counting rules for each BFC type are defined in Clause 6. The process for counting the total functional size is explained in section 8.

8. Document the instance of the FiSMA 1.1 count details using a spreadsheet or other software tool that clearly identifies: a description or identifier of the function counted, the BFC class and type, the specific components counted for each BFC type (e.g., reading and writing references, etc.), and the results of the BFC class calculations. The total FiSMA 1.1 function point count should also be documented as the sum total of all the included BFC types counted. Each FiSMA FSM instance should be clearly designated with the FiSMA version number used in the count, along with an indicator and a description of any local customizations that were used.

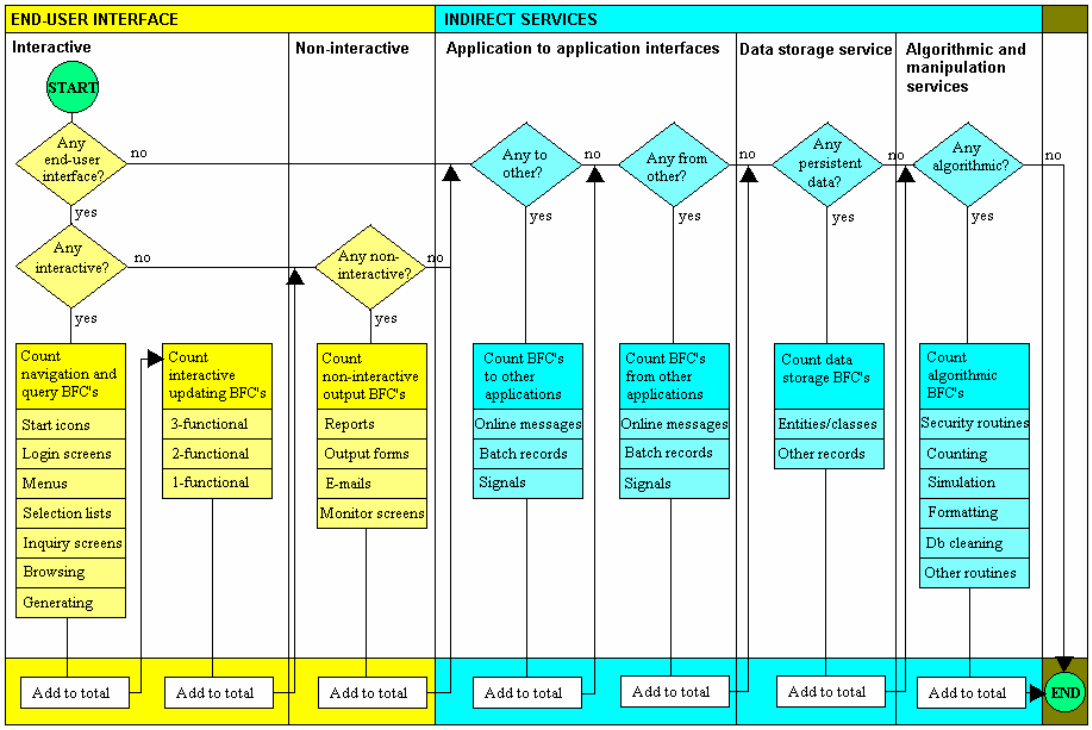


Figure 4 — FiSMA 1.1 process

NOTE While this diagram implies some sequence in the counting process, this is not necessarily true when performing the functional size measurement. Figure 4 is simply arranged to make it easy for measurement practitioners to methodologically work through all of the BFC types and not skip any types inadvertently in the process. Any of the BFC types can be counted as they are encountered in the documentation of the functional user requirements.

© ISO/IEC 2008 All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6 Counting rules for each BFC type class

6.1 Interactive end-user navigation and query BFC's (q)

$S_q = a_q + n/ d_q + r_r/ c_q$

where

S_q = size of query (dialog, menu etc)

n = number of data elements, fields

r_r = number of reading references to entities

d_q = BFC class specific number of data elements giving 1 Ffp

c_q = BFC class specific number of reading references giving 1 Ffp

a_q = establishment constant, 0,2 Ffp

Ffp = functional sizing unit FiSMA function points, value of d_q is 7,00 and value of c_q is 2,00

6.2 Interactive end-user input BFC's (i)

$S_i = m * (a_i + n/ d_i + r_w/ c_i + r_r/ b_i)$

where

S_i = size of input

m = functionality multiplier, value 1, 2 or 3, depending on how many of functions create, update and delete the BFC incorporates

n = number of data elements, fields

r_w = number of writing references to entities

r_r = number of only reading references to entities

d_i = BFC class specific number of data elements giving 1 Ffp

c_i = BFC class specific number of writing references giving 1 Ffp

b_i = BFC class specific number of reading references giving 1 Ffp

a_i = establishment constant, 0,2 Ffp

Ffp = functional sizing unit FiSMA function points, value of d_i is 5,00, value of c_i is 1,50 and value of b_i is 2,00

6.3 Non-interactive end-user output BFC's (o)

$S_o = a_o + n/ d_o + r_r/ c_o$

where

S_o = size of output

n = number of data elements, fields

r_r = number of reading references to entities

© ISO/IEC 2008 – All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

d_o = BFC class specific number of data elements giving 1 Ffp

c_o = BFC class specific number of reading references giving 1 Ffp

a_o = establishment constant, 1,0 Ffp

Ffp = functional sizing unit FiSMA function points, value of d_o is 5,00 and value of c_o is 2,00

6.4 Interface BFC's to other applications (t)

$$S_t = a_t + n/d_t + r_r/ c_t$$

where

S_t = size of interface to other application

n = number of data elements (attributes)

r_r = number of reading references to entities

d_t = BFC class specific number of data elements giving 1 Ffp

c_t = BFC class specific number of reading references giving 1 Ffp

a_t = establishment constant, 0,5 Ffp

Ffp = functional sizing unit FiSMA function points, value of d_t is 7,00 and value of c_t is 2,00

6.5 Interface BFC's from other applications (f)

$$S_f = a_f + n/ d_f + r_w/ c_f + r_r / b_f$$

where

S_f = size of interface from other application

n = number of data elements, fields

r_w = number of writing references to entities

r_r = number of only reading references to entities

d_f = BFC class specific number of data elements giving 1 Ffp

c_f = BFC class specific number of writing references giving 1 Ffp

b_f = BFC class specific number of reading references giving 1 Ffp

a_f = establishment constant, 0,2 Ffp

Ffp = functional sizing unit FiSMA function points, value of d_f is 5,00, value of c_f is 1,50 and value of b_f is 2,00

6.6 Data storage services (d)

$$S_d = a_d + n/d_d$$

where

S_d = size of entity or record

© ISO/IEC 2008 – All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

n = number of data elements (attributes)

dd = BFC class specific number of data elements giving 1 Ffp

a_d = establishment constant, 1,5 Ffp

Ffp = functional sizing unit FiSMA function points, value of d_d is 4,00

6.7 Algorithmic and manipulation services (a)

$$S_a = a_a + n/ d_a + r_c/ c_a$$

where

S_a = size of algorithm

n = number of data elements (variables, operands)

r_c = number of operations

d_a = BFC class specific number of data elements giving 1 Ffp

c_a = BFC class specific number of calculation rules giving 1 Ffp

a_a = establishment constant, 0,1 Ffp

Ffp = functional sizing unit FiSMA function points, value of d_a is 5,00 and value of c_a is 3,00

7 Functional size measurement unit

FiSMA 1.1 functional size unit is called a FiSMA function point (Ffp).

NOTE In general practice, when it is obvious that the functional size measurement method is FiSMA 1.1, the shortened version without the preceding 'F' can be used (e.g., fp).

8 Calculation of the FiSMA 1.1 functional size of a piece of software

The scope of Functional Size Measurement for FiSMA 1.1 is one piece of software or a set of pieces of software. Each piece of software shall be measured separately and all functional sizes may be added together.

The total functional size of the measured software system is the sum of functional sizes of its components, i.e. the pieces of software included in the system.

The functional size (S) of a piece of software is the sum of the sizes (S_x) of BFC's by class. The size of a BFC depends on its type and the amount of class specific elements defined in Clause 6.

$$S = S_q + S_i + S_o + S_f + S_t + S_d + S_a$$

The functional size of a multi-component software is the sum of the functional sizes of the components.

9 Measurement reporting

When reporting the FiSMA 1.1 functional size, the measurement unit and the method version shall be presented.

--`,,```,,,,````-`-`,,`,,`,`,,`---

© ISO/IEC 2008 – All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

FiSMA 1.1 functional size is typically reported as the sum total as outlined in section 8. However, there are management needs for reporting at a more granular level (e.g., reporting each BFC class or BFC type within the functional size). These needs come from the following reasons for reporting functional size:

- contract authorisation or progress reporting;
- traceability with original functional user requirements;
- verification and scope management of functional user requirements;
- certification of software content or quality;
- maintainability of the functional size measurement results.

10 Convertibility from FiSMA 1.1 to other FSM Methods

There is no known convertibility between FiSMA 1.1 function point results and equivalent values in the units of measure for other FSM Methods. Additionally, there is no established convertibility between FiSMA 1.1 and other FSM Methods.

--`,,```,,,,````-`-`,,`,,`,`,,`---

© ISO/IEC 2008 – All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex A

Glossary of terms relevant to FiSMA 1.1

A.1 Boundary

A conceptual interface between the software under study and its users.

A.2 Boundary of a piece of software

A conceptual frontier between a piece of software and the environment in which it operates, as it is perceived from the perspective of its users. The boundary allows the measurer to distinguish, without ambiguity, what is included inside the measured software from what is part of the measured software's operating environment.

A.3 Boundary of a system

A conceptual interface between the system and its external users.

A.4 Component of the software

A set of functional services in the software, which, when implemented represents a well-defined set of functions and is distinguishable by a unique name. A component may or may not be independently managed from the end-user or administrator's point of view. A component may also include other components and/or be a part of other components.

A.5 Data base

A collection of data describing a specific target area that is used and updated by one or more applications.

A.6 Data group

A distinct, non-empty, non-ordered and non-redundant set of data elements. Each included data element describes a complementary aspect of the same object of interest. A data group is characterized by its persistence.

A.7 Data element

The smallest parcel of information, within an identified data group, carrying a meaning from the perspective of the software's Functional User Requirements.

NOTE Data element is often used interchangeably with Data item.

--`,,```,,,,````-`-`,,`,,`,`,,`---

© ISO/IEC 2008 – All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

A.8 Entity

Logical components of the data store, representing fundamental things of relevance to the user, and about which persistent information is stored.

A.9 Experience database

The FiSMA maintained database containing measurement data from completed software development projects.

NOTE Project and maintenance data added to the experience data base must fulfill FiSMA's requirements for completeness and consistency.

A.10 FiSMA

Finnish Software Measurement Association. It is a network of Finnish companies, which share interest to develop software measurement and or software processes. FiSMA is the developer of FiSMA 1.1.

NOTE FiSMA was previously known as Finnish Software Metrics Association (1997-2002).

A.11 Functional process (transactional process)

An elementary component of a set of Functional User Requirements, comprising a unique, cohesive and independently executable set of data or data movements (functional services). It is triggered by one or more triggering event (-types), either directly, or indirectly via an user. A functional process is complete when it has executed all steps required to be done in response to the triggering event.

A.12 Functional service

A service that must be implemented in the piece of software in order to fulfill functional user requirements.

A.13 Multi-component system

Measured system consisting of more than one piece of software.

A.14 Piece of software

Software being measured. It can be a new software application or component to be developed or an existing one.

NOTE Measurement can also be targeted to reflect added, modified and/or removed parts of an existing software.

A.15 Reading reference

A piece of software may refer to one or more entities without updating the contents of the entity, in order to provide any kind of functional service. The number of reading references has an impact on the functional size of most of the BFC types.

© ISO/IEC 2008 – All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

A.16 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 organisation needs to know the size of its software portfolio, then the Scope of the FSM will include all the Functional User Requirements currently utilised. 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.

A.17 User need

The set of functional user requirements and non-functional user requirements that the users need the system to fulfill. All functional services provided by a system or a piece of software are derived from the functional user requirements.

A.18 Viewpoint of the measurement

Measurements are usually made in order to assist management of software development or maintenance. Viewpoint explains the particular way of thinking about the measurement. The viewpoint should be decided in advance, because different viewpoints may necessitate different levels of granularity for reporting the functional size measurement results.

A.19 Writing reference

A piece of software may refer to one or more entities in order to update the contents of the entity. Writing references may be related to interactive end-user input services or interface services from other pieces of software. The number of writing references has an impact on the functional size of these types of BFCs.

© ISO/IEC 2008 – All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ICS 35.080

Price based on 16 pages

--`,,```,,,,````-`-`,,`,,`,`,,`---

© ISO/IEC 2008 – All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale