IEEE Std 14143.1-2000
(Adoption of
ISO/IEC 14143-1:1998)

Implementation Note for IEEE Adoption of ISO/IEC 14143-1:1998 Information Technology—Software Measurement—Functional Size Measurement—Part 1: Definition of Concepts

Sponsor

Software Engineering Standards Committee of the IEEE Computer Society

Approved 30 January 2000

IEEE-SA Standards Board

Abstract: Implementation notes that relate to the IEEE interpretation of ISO/IEC 14143-1:1998 are described.
Keywords: function point, Functional Size Method, measurement, metrics

---

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Print: ISBN 0-7381-2165-7 SH94840
PDF: ISBN 0-7381-2166-5 SS94840

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Introduction

(This introduction is not part of IEEE Std 14143.1-2000, Implementation Note for IEEE Adoption of ISO/IEC 14143-1:1998 Information Technology—Software Measurement—Functional Size Measurement—Part 1: Definition of Concepts.)

This standard is an adoption of ISO/IEC 14143-1:1998 and contains implementation notes based on the IEEE interpretation of ISO/IEC 14143-1:1998.

Participants

The following members of the balloting committee voted on this standard:

Syed Ali
Baku Banerjee
Edward E. Bartlett
Barbara K. Beauchamp
Leo Beltracchi
Richard E. Biehl
Juris Borzovs
Greg Brown
Enrico A. Carrara
Keith Chan
Antonio M. Cicu
Theo Clarke
Francois Coallier
Rosemary Coleman
Paul R. Croll
Martin D'Souza
Christopher Dale
Geoffrey Darnton
Taz Daughtrey
Carol Dekkers
Bostjan K. Derganc
Perry R. DeWeese
Edward J. Dudash
George Duffield
John Durbin
Christof Ebert
Leo G. Egan
Franz D. Engelmann
William Eventoff
Jonathan H. Fairclough
Richard E. Fairley
John W. Fendrich
Neal Fishman
John H. Fowler
David Garmus
Marilyn Ginsberg-Finner
Shirley A. Gloss-Soler
John Garth Glynn
Charles Gold
Julio Gonzalez-Sanz
L. M. Gunther
John Harauz
Rob Harker
Rick Hefner
James H. Heil
Debra Herrmann
Stan Hopkins
John W. Horch
David Hsiao
Frank V. Jorgensen
Vladan V. Jovanovic
William S. Junk
Chris F. Kemerer
Ron S. Kenett
Judith S. Kerner
Robert J. Kierzyk
Shaye Koenig
Magnus Kueper
Thomas M. Kurihara
Fung-chun Leonard Lam
John B. Lane
J. Dennis Lawrence
Karl Leung
Jim J. Logan
John Lord
Hal Lowe
Alex Lubashevsky
Stan Magee
Henry A. Malec
Tomoo Matsubara
Frank Mazzucco
Ian R. McChesney
Patrick D. McCray
Denis C. Meredith
Jair Merlo
Jerome W. Mersky
Kathryn J. Moland
James W. Moore
Francisco Navas Plano
Pavol Navrat
Susumu Ohno
Gerald L. Ourada
Mark Paulk
A. Pedar
Peter T. Poon
Kenneth R. Ptack
John Purssey
Edwin Quinones
John Reddan
Ann E. Reedy
Donald J. Reifer
Andrew P. Sage
Helmut Sandmayr
Frederico Sousa Santos
Robert J. Schaaf
Stephen R. Schach
Norman Schneidewind
David J. Schultz
David Seaver
Carl A. Singer
Linda Smith
Jag Sodhi
Alfred R. Sorkowitz
Mark Spillers
Julia Stesney
Fred J. Strauss
Robert N. Sulgrove
P. Suresh
Toru Takeshita
Richard H. Thayer
Booker Thomas
Leonard L. Tripp
Glenn D. Venables
Jose Carrillo Verdun
Scott A. Whitmire
Paul A. T. Wolfgang
Paul R. Work
Janusz Zalewski
Geraldine Zimmerman

When the IEEE-SA Standards Board approved this standard on 30 January 2000, it had the following membership:

Richard J. Holleman, *Chair*
Donald N. Heirman, *Vice Chair*
Judith Gorman, *Secretary*

Satish K. Aggarwal
Dennis Bodson
Mark D. Bowman
James T. Carlo
Gary R. Engmann
Harold E. Epstein
Jay Forster*
Ruben D. Garzon
James H. Gurney
Lowell G. Johnson
Robert J. Kennelly
E. G. "Al" Kiener
Joseph L. Koepfinger*
L. Bruce McClung
Daleep C. Mohla
Robert F. Munzner
Louis-François Pau
Ronald C. Petersen
Gerald H. Peterson
John B. Posey
Gary S. Robinson
Akio Tojo
Hans E. Weinrich
Donald W. Zipse

*Member Emeritus

Also included is the following nonvoting IEEE-SA Standards Board liaison:

Robert E. Hebner

Catherine Berger
IEEE Standards Project Editor

Contents

Implementation Note for IEEE Adoption of ISO/IEC 14143-1:1998 Information Technology—Software Measurement—Functional Size Measurement—Part 1: Definition of Concepts

1. Implementation Note for IEEE Std 14143.1-2000

IEEE Std 14143.1-2000 is an adoption of ISO/IEC 14143-1:1998.[1] The following implementation notes relate to the IEEE interpretation of ISO/IEC 14143-1:1998. Two differences that affect conformance are noted in Clause 6 of this document.

2. Terminology

Some differences exist between the terminology used in IEEE Std 14143.1-2000 and that in IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Technology, because the terminology of the International Standard is specialized to its particular purpose. The reader should use the local definitions in IEEE Std 14143.1-2000 where there is a difference with those of IEEE Std 610.12-1990. In some cases, the definition in IEEE Std 610.12-1990 may provide further clarification. It is important to keep in mind, however, that full compatibility of definitions has yet to be achieved in the evolving field of software engineering.

The terms *functional user requirement, quality requirement,* and *technical requirement* are used in this implementation note to classify requirements into categories that are deemed either to contribute or not to contribute to a measurement of Functional Size. For this specific purpose, the terms are used in a manner that may not be consistent with usage in other IEEE Software Engineering standards.

The definition of *user* encompasses, not merely humans who interact with the system, but also *things,* such as hardware and software components of interacting systems.

---

[1] ISO and ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse. ISO publications are also available in the United States from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.

The Note to Annex A.3.4 draws a distinction between *software enhancement* and *software maintenance*. The definition of *maintenance* in IEEE Std 610.12-1990 includes the concept of enhancement.

None of these differences cause a contradiction between this standard and other IEEE standards. They merely indicate that users of the standard should pay careful attention to the definitions within the particular context of the standard.

3. Related IEEE standard

IEEE Std 1045-1992, IEEE Standard for Software Productivity Metrics

> *This standard provides some information regarding the utility of "function points," an example of Functional Size Measurement.*

4. The role of the standard

IEEE Std 14143.1-2000 deals with the fundamental concepts that apply to the general class of Functional Size Measurement (FSM) methods. An important principle of FSM methods is that they are independent of quality requirements (including performance requirements) and technical requirements. This does not diminish the importance of such requirements in the project planning, estimating, and measurement processes. However, the Functional Size itself is independent of such requirements; those requirements are typically introduced as project characteristics that are used in estimating and analysis of results.

This standard does not provide detailed rules for selecting a FSM method, using a method to measure the Functional Size of software, or using Functional Size for project planning, estimating, and measurement. An example of a method for measuring Functional Size is provided by *IFPUG Function Points, Release 4.1*.[2]

5. Errata

The editor of the working group that developed the International Standard noted the following errors in the published standard:

a) **Introduction, second paragraph:** There should be a second sentence that reads, "In 1979, Allan Albrecht of IBM was the first to publicly release a method based on such concepts, known as Function Point Analysis."

b) **Clause 5.2.2, Note 2:** Change "if more than two files are accessed" to "if two or more files are accessed."

6. Exceptions

The following items reference normative material in ISO/IEC 14143-1:1998. In this regard, conformance with IEEE Std 14143.1-2000 differs from conformance with ISO/IEC 14143-1:1998. The changes are minor and probably have no impact on conformance in actual situations.

[2] IFPUG publications are available from the International Function Point Users Group, 5008-28 Pine Creek Drive, Westerville, OH 43081, USA, 614-895-7130.

a) **Clause 5.1.1.1a:** This subclause requires that an FSM method shall be "based on" a representation, perhaps a particular representation, of Functional User Requirements. Practice indicates, though, that FSM is successfully performed using a variety of different representations of Functional User Requirements. Therefore, the term "based on" seems too strong. Substitute the following text:

it can be used with a representation of the Functional User Requirements from the perspective of the users.

b) **Clause 5.2.1.1d:** This subclause is a requirement that an FSM method shall "describe the Functional Domain(s) to which the FSM method can be applied." Because the International Standard proposed to enumerate the Functional Domains has been reclassified as a Technical Report that remains incomplete, this provision is premature. Replace with the following text:

either state that it may be applied to all classes of software, or describe those classes of software to which it may be applied.

7. Clarifications

The following items reference material in ISO/IEC 14143-1:1998 that are recommendations or information and that are not related to conformance.

a) **Note to Clause 3.11:** The note is an overly simplified treatment of a complex subject. Delete the note.

b) **Clause 5.1.3e and 5.1.3f:** These subclauses reference independence from physical components and technological components of the software being measured. Unfortunately, these terms are not defined in the standard. While they do not appear to be related to quality requirements, it appears that they may be subtypes of technical requirements as defined in 3.12. For clarification, the following examples are offered:

— Physical components
Examples include: source language and version, bytes of source code, code complexity

— Technological components
Examples include: development platform (hardware and software), target execution platform (hardware and software)

c) **Clause 5.2.1.1a:** This subclause requires that an FSM method shall "define the attributes of the BFCs" but does not explain what an attribute is. For the purpose of this adoption, the following Note should be inserted:
Note—The "attributes" of a BFC are features that may be counted.

d) **Clause 5.2.1.2d:** This clause is a recommendation that an FSM method should "state its degree of interoperability to other sizing methods." The recommendation seems overly general because "other sizing methods" is not specific to FSM or even to software. At the current time, there is no catalog or register of FSM methods to support implementation of the recommendation. The list item is deleted.

e) **Annex A.1, second paragraph, last sentence:** Change "may contribute to the practical use" to "should be considered to allow the practical use."

f) **Annex A.3.2, second sentence:** Change "Productivity indicators (Functional Size divided by effort, schedule or cost)" to "Productivity indicators (measured in part by Functional Size divided by effort, schedule, or cost)."

g) **Annex A.3.3:** This example deals with "defect density," a measurement commensurate with implementation rather than with functional user requirements. Delete it.

8. One Functional Size versus many Functional Sizes

There is some controversy in the FSM community regarding the nature of Functional Size. There are three major alternatives:

1) Should Functional Size be regarded as a property inherent to the software being measured and which is approximated by different methods providing differing results?
2) Should Functional Size be regarded as a property of both the software and the method, meaning that a piece of software has several functional sizes?
3) Should Functional Size be regarded as a property of the software, the method, and the intended use of the method (e.g., estimating, productivity measurement, etc.)?

The wording of the standard leaves the situation unclear. A consistent interpretation of the standard can be attained by assuming the second position. In this case, one would interpret the definition in 3.6 to refer to the unique size obtained by applying a specific FSM method to a specific set of software. This is consistent with the references in 5.1.3a through 5.1.3f, which refer to Functional Size as “it.” With this understanding, the phrase “a functional size” is correct when making a generalization, and the phrases “the functional size” and “it” are correct when dealing with the result of a specific instance of FSM.

INTERNATIONAL STANDARD

ISO/IEC 14143-1

First edition
1998-06-15

Information technology — Software measurement — Functional size measurement —

Part 1: Definition of concepts

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

Partie 1: Définition des concepts

Reference number
ISO/IEC 14143-1:1998(E)

Contents

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 9126:1991[1)], *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 - 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.

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

characteristics which could influence the software development, enhancement, or maintenance processes. Examples are staff experience, tools usage, user relationships, working conditions, staff business knowledge and development language. Productivity could be managed by manipulating those characteristics and monitoring the productivity trends of future software projects to see if the desired effect has been achieved.

A.3.3 Quality management

An FSM Method can assist with managing the number of defects. Defect density (the ratio of the number of defects identified within a period of time to Functional Size) could be analyzed to determine which demographic characteristics have the greatest impact on defect density. Defect density could be managed by manipulating those characteristics and monitoring the defect density trends of future software projects to see if the desired effect has been achieved.

A.3.4 Project resource forecasting

For new development and enhancement projects, an algorithmic forecasting model can be constructed from various types of data collected from a sample of completed projects, for example, Functional Size, Quality Requirements, Technical Requirements, the resources consumed (expressed as cost, effort, or schedule consumed), and the demographic characteristics expected to have had an influence on the amount of resources consumed. Once a model has been constructed, a forecast of resources can be generated early in the lifecycle of future software projects by entering, for example, the following information into the model:

- the Functional Size of the software;

- the expected influence of Quality Requirements, Technical Requirements and demographic characteristics.

NOTE - Software enhancement is the process of modifying software to add, change and delete user functionality in comparison to software maintenance, which is the process of modifying software to correct defects, improve performance, and support computing environment changes (such as the addition of new types of data storage devices).

A.3.5 Budgeting for maintenance

An FSM Method can assist with budgeting for the maintenance of an organization's software portfolio. Functional Size of the portfolio, as well as the maintenance cost or effort compared to Functional Size, could be monitored. This information could be used to forecast maintenance budgets.

A.3.6 Contract management

An FSM Method can assist in managing the cost and schedule of software development by software suppliers. The contract with a vendor can be based on the functional requirements, Functional Size, an expected productivity rate and a cost per Functional Size unit.

ICS 35.080

Descriptors: data processing, information interchange, computer software, measurement, functional size, concepts, definitions.

Price based on 6 pages