INTERNATIONAL STANDARD

ISO/IEC 20926

Second edition
2009-12-01

Software and systems engineering — Software measurement — IFPUG functional size measurement method 2009

Ingénierie du logiciel et des systèmes — Mesurage du logiciel — Méthode IFPUG 2009 de mesurage de la taille fonctionnelle

Reference number
ISO/IEC 20926:2009(E)

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 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

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 20926 was prepared by the International Function Point Users Group (IFPUG) and was adopted, under the PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

This second edition cancels and replaces the first edition (ISO/IEC 20926:2003), which has been technically revised.

Introduction

The use of function points, as a measure of the functional size of software, has grown since the mid 1970s from a few interested organizations to an impressive list of organizations worldwide. Allan Albrecht was the first to publicly release a method for functionally sizing software called function point analysis. With the growth in the use of function points, there has been wider application and use of the measure. Since its formation in 1986 the International Function Point Users Group (IFPUG) has continuously enhanced the original Albrecht method for functionally sizing software. This International Standard is the latest release in the continually improving IFPUG method that promotes the consistent interpretation of functional size measurement in conformance with ISO/IEC 14143-1:2007. The IFPUG functional size measurement method is known as function point analysis and its units of functional size are called Function Points.

Organizations can apply this International Standard to measure the size of a software product to:

- support quality and productivity analysis;
- estimate cost and resources required for software development, enhancement and maintenance;
- provide a normalization factor for software comparison;
- determine the size of a purchased application package by functionally sizing all the functions included in the package;
- assist users in determining the benefit of an application package to their organization by functionally sizing functions that specifically match their requirements.

Function point analysis measures software by quantifying the tasks and services (i.e., functionality) that the software provides to the user based primarily on logical design. The objectives of function point analysis are to measure:

- functionality implemented in software, that the user requests and receives;
- functionality impacted by software development, enhancement and maintenance independently of technology used for implementation.

The process of function point analysis is:

- simple enough to minimize the overhead of the measurement process;
- a consistent measure among various projects and organizations.

In order to effectively apply this International Standard, persons can be formally trained in the method using IFPUG certified course materials.

This International Standard is one component in the IFPUG publications. It is recommended that it be read in conjunction with the other IFPUG publications. These provide guidance to application of the rules specified within this International Standard and background information to aid in understanding the use and applicability of the resulting functional size. Supporting IFPUG publications include the following:

- the current version of the IFPUG Counting Practices Manual, which incorporates this International Standard supplemented with counting practices and examples that support its implementation;

- “Framework for Functional Sizing”, 2005, which discusses the contribution of both functional size and non-functional size to the overall software product size; the IFPUG FSM method is a method for measuring the functional size;
- IFPUG website at www.ifpug.org.

Software and systems engineering — Software measurement — IFPUG functional size measurement method 2009

1 Scope

1.1 Purpose

This International Standard specifies the set of definitions, rules and steps for applying the IFPUG functional size measurement (FSM) method.

1.2 Conformity

This International Standard is conformant with all mandatory provisions of ISO/IEC 14143-1:2007.

1.3 Applicability

This International Standard can be applied to all functional domains.

NOTE IFPUG continues to publish white papers providing guidelines for use in evolving environments and domains.

This International Standard is fully convertible to prior editions of IFPUG sizing methods.

IFPUG function point analysts have identified different delivery rates (hours to deliver a function point) related to building applications in different functional domains calibrated for varying project sizes and software complexities.

1.4 Audience

This International Standard can be applied by anyone requiring a measurement of functional size. Persons experienced with the method will find this International Standard to be a useful reference.

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.

3.1
adaptive maintenance
modification of a software product, performed after delivery, to keep a software product usable in a changed or changing environment

NOTE Adaptive maintenance provides enhancements necessary to accommodate changes in the environment in which a software product must operate. These changes are those that must be made to keep pace with the changing environment. For example, the operating system might be upgraded and some changes may be made to accommodate the new operating system.

[ISO/IEC 14764:2006, 3.1]

3.2
application
cohesive collection of automated procedures and data supporting a business objective, consisting of one or more components, modules, or subsystems

EXAMPLES Accounts payable, accounts receivable, payroll, procurement, shop production, assembly line control, air search radar, target tracking, weapons firing, flight line scheduling and passenger reservations.

3.3
application functional size
measure of the functionality that an application provides to the user, determined by the application function point count

3.4
application function point count
activity of applying this International Standard to measure the functional size of an application

3.5
arranging
activity of sequencing attributes in a transactional function

3.6
associative entity type
entity type that contains attributes which further describe a many-to-many relationship between two other entity types

3.7
attributive entity type
entity type that further describes one or more attributes of another entity type

3.8
base functional component
BFC
elementary unit of Functional User Requirements defined by and used by an FSM Method for measurement purposes

EXAMPLE A Functional User Requirement could be “Maintain Customers”, which might 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.

[ISO/IEC 14143-1:2007, 3.1]

3.9
boundary
conceptual interface between the software under study and its users

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

NOTE ISO/IEC 20926:2003 used the term “application boundary”.

3.10
consistent state
point at which processing has been fully executed, the Functional User Requirement has been satisfied and there is nothing more to be done

EXAMPLE 1 The Functional User Requirement is to print a check and mark the appropriate account as paid. If only part of the Functional User Requirement is completed (e.g., only printing the check or only marking it as paid), the application is not in a consistent state. The printing of a check without marking the account as paid causes an inconsistency in the application as does marking it as paid without printing it.

EXAMPLE 2 The Functional User Requirement is to have a batch process that accepts an input file to update a data store, produce a production control report and return an error report back to the sending application. The process is not in a consistent state unless all of these parts are completed.

EXAMPLE 3 The Functional User Requirement is to transfer an employee to a new job and validate his/her security clearance level. To complete this, a real-time request is sent to the security application (which maintains government security clearances and not application security) and a response received before the transfer can be completed. All steps are needed to create a consistent state. The interaction with the security application is not an independent step or action. It does not happen in and of itself, and the transaction to transfer an employee is not in a consistent state without it.

3.11
control information
data that influences an elementary process by specifying what, when or how data is to be processed

3.12
conversion functionality
transactional or data functions provided to convert data and/or provide other user specified conversion requirements

NOTE Conversion functionality exists only during the development or enhancement of an application.

3.13
corrective maintenance
reactive modification of a software product performed after delivery to correct discovered problems

NOTE The modification repairs the software product to satisfy requirements.

[ISO/IEC 14764:2006, 3.2]

3.14
counting scope
set of Functional User Requirements to be included in the function point count

3.15
Data Element Type
DET
unique, user recognizable, non-repeated attribute

3.16
data function
functionality provided to the user to meet internal or external data storage requirements

NOTE A data function is either an Internal Logical File or an External Interface File.

3.17
derived data
data created as a result of processing that involves steps other than or in addition to direct retrieval and validation of information from data functions

3.18
development project
project to develop and deliver the first release of a software application

3.19
development project functional size
measure of the functionality provided to the users with the first release of the software, as measured by the development project function point count

NOTE The functional size of a development project can include the size of conversion functionality.

3.20
development project function point count
activity of applying this International Standard to measure the functional size of a development project

3.21
elementary process
smallest unit of activity that is meaningful to the user

3.22
enhancement project
project to develop and deliver adaptive maintenance

NOTE An enhancement project can also develop and deliver corrective and perfective maintenance, but these do not contribute to the enhancement project functional size.

3.23
enhancement project functional size
measure of the functionality added, changed or deleted at the completion of an enhancement project, as measured by the enhancement project function point count

NOTE The functional size of an enhancement project can include the size of conversion functionality.

3.24
enhancement project function point count
activity of applying this International Standard to measure the functional size of an enhancement project

3.25
entity dependent
⟨entity⟩ not meaningful or not significant to the business in and of itself without the presence of other entities, such that

— an occurrence of entity X must be linked to an occurrence of entity Y, and

— the deletion of an occurrence of entity Y results in the deletion of all related occurrences of entity X

3.26
entity independent
⟨entity⟩ meaningful or significant to the business in and of itself without the presence of other entities

3.27
External Input
EI
elementary process that processes data or control information sent from outside the boundary

NOTE An External Input is a type of base functional component.

3.28
External Inquiry
EQ
elementary process that sends data or control information outside the boundary

NOTE An External Inquiry is a type of base functional component.

3.29
External Interface File
EIF
user recognizable group of logically related data or control information, which is referenced by the application being measured, but which is maintained within the boundary of another application

NOTE An External Interface File is a type of base functional component.

3.30
External Output
EO
elementary process that sends data or control information outside the boundary and includes additional processing logic beyond that of an External Inquiry

NOTE An External Output is a type of base functional component.

3.31
File Type Referenced
FTR
data function read and/or maintained by a transactional function

3.32
functional complexity
specific complexity rating assigned to a function using the rules as defined within this International Standard

3.33
functional size
size of the software derived by quantifying the Functional User Requirements

[ISO/IEC 14143-1:2007, 3.6]

3.34
Functional User Requirements
sub-set of the user requirements specifying what the software shall do in terms of tasks and services

NOTE 1 Functional User Requirements include, but are not limited to, the following:

- data transfer (for example: Input customer data, Send control signal);
- data transformation (for example: Calculate bank interest, Derive average temperature);
- data storage (for example: Store customer order, Record ambient temperature over time);
- data retrieval (for example: List current employees, Retrieve aircraft position).

User requirements that are not Functional User Requirements include, but are not limited to, the following:

- quality constraints (for example: usability, reliability, efficiency and portability);
- organizational constraints (for example: locations for operation, target hardware and compliance to standards);
- environmental constraints (for example: interoperability, security, privacy and safety);
- implementation constraints (for example: development language, delivery schedule).

NOTE 2 Adapted from ISO/IEC 14143-1:2007, 3.8.

3.35
Function Point
FP
unit of measure for functional size as defined within this International Standard

3.36
Function Point Analysis
FPA
method for measuring functional size as defined within this International Standard

3.37
function point count
activity of applying the rules within this International Standard to measure the functional size of an application or project

NOTE There are three types of function point count: application, development project and enhancement project.

3.38
function type
type of base functional component identified in this International Standard

NOTE The five function types identified in this International Standard are: External Input, External Output, External Inquiry, Internal Logical File and External Interface File.

3.39
Internal Logical File
ILF
user recognizable group of logically related data or control information maintained within the boundary of the application being measured

NOTE An Internal Logical File is a type of base functional component.

3.40
maintain
add, change or delete data through an elementary process

3.41
meaningful
user recognizable and satisfies a Functional User Requirement

3.42
perfective maintenance
modification of a software product after delivery to detect and correct latent faults in the software product before they are manifested as failures

NOTE 1 Adapted from ISO/IEC 14764:2006, 3.7.

NOTE 2 Perfective maintenance provides enhancements for users, improvement of program documentation, and recoding to improve software performance, maintainability, or other software attributes.

NOTE 3 Contrast with: adaptive maintenance; corrective maintenance.

3.43
primary intent
intent that is first in importance

3.44
processing logic
any of the requirements specifically requested by the user to complete an elementary process such as validations, algorithms or calculations and reading or maintaining a data function