

AS/NZS ISO/IEC 20926:2022
ISO/IEC 20926:2009



Australian/New Zealand Standard™

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



AS/NZS ISO/IEC 20926:2022

This Joint Australian/New Zealand Standard™ was prepared by Joint Technical Committee IT-015, Software and Systems Engineering. It was approved on behalf of the Council of Standards Australia on 11 May 2022 and by the New Zealand Standards Approval Board on 4 May 2022.

This Standard was published on 27 May 2022.

The following are represented on Committee IT-015:

- Australian Computer Society
- Australian Digital Health Agency
- Australian Society for Technical Communication (NSW)
- Department of Defence (Australian Government)
- Engineers Australia
- Griffith University
- Institute of IT Professionals New Zealand
- IT Service Management Forum Australia
- NSW Business Chamber
- Systems Engineering Society of Australia
- Universities New Zealand
- University of New South Wales
- University of Southern Queensland
- University of Technology Sydney

This Standard was issued in draft form for comment as DR AS/NZS ISO/IEC 20926:2022.

Keeping Standards up-to-date

Ensure you have the latest versions of our publications and keep up-to-date about Amendments, Rulings, Withdrawals, and new projects by visiting:

www.standards.org.au

www.standards.govt.nz

ISBN 978 1 76113 765 5

Australian/New Zealand Standard™

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

Originated as AS/NZS ISO/IEC 20926:2008.
Second edition 2022.

COPYRIGHT

© ISO/IEC 2022 — All rights reserved

© Standards Australia Limited/the Crown in right of New Zealand, administered by the New Zealand Standards Executive 2022

All rights are reserved. No part of this work may be reproduced or copied in any form or by any means, electronic or mechanical, including photocopying, without the written permission of the publisher, unless otherwise permitted under the Copyright Act 1968 (Cth) or the Copyright Act 1994 (New Zealand).

Preface

This Standard was prepared by the Joint Standards Australia/Standards New Zealand Committee IT-015, Software and Systems Engineering, to supersede AS/NZS ISO/IEC 20926:2008, *Software engineering—IFPUG 4.1 Unadjusted functional size measurement method—Counting practices manual*.

The objective of this document is to specify the set of definitions, rules and steps for applying the IFPUG (International Function Point Users Group) functional size measurement (FSM) method.

This document is conformant with all mandatory provisions of ISO/IEC 14143-1:2007. It can be applied to all functional domains and 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.

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

This document is identical with, and has been reproduced from, ISO/IEC 20926:2009, *Software and systems engineering — Software measurement — IFPUG functional size measurement method 2009*.

As this document has been reproduced from an International document, the following apply:

- (a) In the source text “this International Standard” should read “this document”.
- (b) A full point substitutes for a comma when referring to a decimal marker.

Australian or Australian/New Zealand Standards that are identical adoptions of international normative references may be used interchangeably. Refer to the online catalogue for information on specific Standards.

The terms “normative” and “informative” are used in Standards to define the application of the appendices or annexes to which they apply. A “normative” appendix or annex is an integral part of a Standard, whereas an “informative” appendix or annex is only for information and guidance.

Contents

Preface	ii
Foreword	iv
Introduction	v
1 Scope	1
1.1 Purpose	1
1.2 Conformity	1
1.3 Applicability	1
1.4 Audience	1
2 Normative references	1
3 Terms and definitions	1
4 Abbreviated terms	8
5 Measurement Process	8
5.1 Overview	8
5.2 Gather the available documentation	9
5.3 Determine the counting scope and boundary and identify Functional User Requirements	9
5.4 Measure data functions	10
5.4.1 Overview	10
5.4.2 Identify and group all logical data into data functions	10
5.4.3 Classify each data function as either an ILF or an EIF	11
5.4.4 Count DETs for each data function	11
5.4.5 Count RETs for each data function	11
5.4.6 Determine the functional complexity for each data function	12
5.4.7 Determine the functional size for each data function	12
5.5 Measure transactional functions	12
5.5.1 Overview	12
5.5.2 Identify elementary processes	13
5.5.3 Classify each elementary process as a transactional function	15
5.5.4 Count FTRs for each transactional function	17
5.5.5 Count DETs for each transactional function	17
5.5.6 Determine the functional complexity for each transactional function	17
5.5.7 Determine the functional size of each transactional function	18
5.6 Measure conversion functionality	18
5.7 Measure enhancement functionality	18
5.8 Calculate functional size	19
5.9 Document the function point count	20
5.10 Report the result of the function point count	21
Annex A (informative) Consolidated complexity and functional size tables	22
Bibliography	24

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.

NOTES

Australian/New Zealand Standard

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 1 to entry: 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.

[SOURCE: 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

EXAMPLE 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.

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

3.9

boundary

conceptual interface between the software under study and its users

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

Note 1 to entry: 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 1 to entry: 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 1 to entry: The modification repairs the software product to satisfy requirements.

[SOURCE: 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 1 to entry: 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 1 to entry: The functional size of a development project can include the size of conversion functionality.