INTERNATIONAL STANDARD

ISO/IEC 20968

First edition
2002-12-01

Software engineering — Mk II Function Point Analysis — Counting Practices Manual

Génie logiciel — Analyse des points fonctionnels Mk II — Manuel des pratiques de comptage

Reference number
ISO/IEC 20968:2002(E)

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

Published in Switzerland

Table of 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 20968 was prepared by the United Kingdom Software Metrics Association (UKSMA) 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.

Software engineering — Mk II Function Point Analysis — Counting Practices Manual

1 Introduction

1.1 Definition and Purpose of MkII Function Point Analysis

For the purposes of this document, the abbreviation 'Mk II FPA' is used for 'Mark II Function Point Analysis'.

Mk II FPA is a method for the quantitative analysis and measurement of information processing applications. It quantifies the information processing requirements specified by the user to provide a figure that expresses a size of the resulting software product. This size is suitable for the purposes of performance measurement and estimating in relation to the activity associated with the software product

In the context of Mk II FPA, 'information processing requirements' means the set of functions required by the commissioning user of the application software product (excluding any technical and quality requirements). 'The activity' could be the development, enhancement or maintenance of the software product needed to meet the requirements.

The MkII FPA method is intended to comply with ISO/IEC 14143-1: 1998, the International Standard for Functional Size Measurement (see Bibliography).

1.2 Purpose of the Counting Practices Manual ('CPM')

The Mk II Method of Function Point Analysis was defined by Charles Symons in "Software Sizing and Estimating: Mk II FPA" published in 1991. After development within KPMG in 1985/86, with the protected status of a proprietary method, the method is now in the public domain. The Metrics Practices Committee (MPC) of the UK Software Metrics Association is now the design authority for the method and is responsible for its continuing development.

The purpose of this Manual is to explain and promulgate the method, and to set out the rules for applying Mk II Function Point Analysis ('FPA'). Chapters 1 to 5 inclusive of this manual provide the authoritative standard of the Mk II FPA Method.

It is not the purpose of this manual to replace Charles Symons' or the other books in the Bibliography. The manual is not intended to provide a teaching introduction to MkII FPA, nor does it discuss the broader subject of software measurement.

This manual replaces all previous versions of the Counting Practices Manual.

The definition covers:

- The software domains for which MkII FPA may be applicable
- Application software requirement components recognised by Mk II FPA
- A process for applying the MkII FPA rules and documenting the result
- Interpretation of the rules for a variety of software technologies (e.g., GUI, client/server, objects, etc.)
- Basic Formulae used in Mk II FPA
- Terminology used in Mk II FPA.

An important aspect of this new version of the standard, in order to comply with ISO/IEC 14143-1:1998, is that the Technical Complexity Adjustment is no longer considered to contribute to the "Functional Size". Hence measurements previously expressed in MkII 'Unadjusted Function Points' should now be regarded as the MkII measures of the Functional Size, without further qualification. For the time being, the Technical Complexity Adjustment remains part of the method. If it has been applied then the result should be qualified as the "Adjusted Size".

1.3 Who should read this document ?

- Users of Mk II FPA
- Suppliers of tools, training or other services involving the method.
- Anyone interested in learning about the details of Mk II FPA

1.4 Albrecht/IFPUG Function Point Analysis

Allan Albrecht developed the original Function Point Analysis method. The design authority for the direct descendent of his approach is now the International Function Point Users Group ('IFPUG'). The briefest reference is made here to the relationship between the Mk II and the IFPUG FPA methods.

The two methods measure subtly but significantly different sizes of a software product (and therefore of the work-output of the processes of developing, maintaining and enhancing a software product).

In terms of the sizes produced, the major differences are that Mk II FPA, with its finer granularity, is a continuous measure whereas IFPUG limits component size once a threshold is reached and that the MkII method aims to better reflect the internal processing complexity of business ‘data-rich’ systems. As the concepts on which the size measure is based are logical transactions and entities, in which software requirements and functional specifications are typically expressed, a MkII Functional Size measure should be truly independent of the technology or methods used to develop or implement the software.

The weightings introduced by Charles Symons were designed to deliver a size scale of similar magnitude for the MkII method as for the IFPUG method. On average therefore, the methods give roughly the same software sizes up to around 400 function points (though there can be quite a scatter about the average for individual items of software). For larger sizes, Mk II FPA tends to produce increasingly higher sizes than the Albrecht/IFPUG method.

For some purposes, e.g. portfolio management, the methods may be regarded as equivalent. However, for the commonest purposes of performance measurement and estimating it is preferable to use one scale or the other consistently, only converting between them if needed, using a formula which shows the average relationship.

1.5 Applicability of Mk II FPA

MkII FPA is a method that assists in measuring process efficiency and managing costs for application software development, enhancement or maintenance activities. It measures a software product size independent of technical characteristics of the software, in terms relevant to users. It can be:

- applied early in the software development process
- applied uniformly throughout the software's lifetime
- interpreted in business terms, and
- understood by users of the software.

MkII Function Points can be used to measure the functional size of any software application that can be described in terms of logical transactions, each comprising an input, process and output component. The sizing rules were designed to apply to application software from the domain of business information systems, where the processing component of each transaction tends to be dominated by considerations of the storage or retrieval of data. The method may be applicable to software from other domains, but the user should note that the sizing rules do not take into account contributions to size such as from complex algorithms as typically found in scientific and engineering software, nor do the rules specifically take into account real-time requirements. To apply MkII FPA to these other domains may be possible or may require extensions to or new interpretations of the rules given in this manual.

MkII FPA can be used for sizing:

- a requirements specification or functional specification of a new application or of a change to an existing application
- the requirements met by an existing, operational application, whether it be a bespoke application or an implementation of a packaged business software solution, and whether a batch or on-line implementation.

Either directly, or coupled with effort, defect counts and other measures, MkII FPA can be used for a variety of purposes, including to:

- measure project or organisational performance (productivity, delivery rate and quality).
- compare internal and external IT performance
- compare application quality and reliability
- compare normalised development, maintenance and support costs of applications on different platforms
- estimate the resourcing requirements, duration and cost of projects
- contribute to the cost and risk elements of the business case for a new project
- assist in identifying all requirements before an application has been developed
- control "creeping elegance" or scope change during projects
- assign work to team members
- determine the size of the application asset base
- produce useful, high-level, functional documentation of old 'legacy' systems that lack up-to-date functional documentation
- determine the replacement value of applications.

MK II FPA is independent of the project management method to be used (e.g. waterfall, spiral, incremental) and of the development method employed (e.g. object-oriented, SSADM, Information Engineering, etc.). It is a measure of the logical, business requirements, independent of how they are implemented.

1.6 Manual Structure

This manual is split into 10 Chapters. After this Introduction, the Chapters address the following:

Chapter 2 describes the Rules to which all Mk II Function Point counts must conform.

Chapter 3 lists the main steps to be carried out in performing a count.

Chapter 4 provides general guidelines and illustrations of how to perform the tasks involved in a count for many situations faced in practice.

Chapter 5 provides measurement guidelines for some sizing specific types of software and for some specific types of sizing requirements.

Chapter 6 describes the process to adjust the Functional Size derived using the MkII method to account for non-functional requirements.

Chapter 7 gives a definition of effort for the calculation of productivity.

Chapter 8 describes how to measure productivity.

Chapter 9 gives a brief introduction to the use of MkII FPA for estimating.

Chapter 10 provides a glossary of terms and definitions.

Appendix 1 contains a detailed definition of the Technical Complexity Adjustment.

Appendix 2 provides some forms which may prove helpful in performing a count.

Appendix 3 contains a Bibliography of publications referred to in this document, and other useful references.

1.7 Metrics Practices Committee

The Metrics Practices Committee was established by the Committee of UKSMA to exercise delegated authority on a day-to-day basis over the Mk II Function Point method. The Committee acts on behalf of UKSMA and the user community with the following objectives:

- to act as the Design Authority and owner of the MkII method
- to maintain control over the rules, interpretation and documentation of the MkII method
- to ensure that the rules are interpreted in a consistent and valid way for both MkII and IFPUG methods
- to supply advice on the interpretation of both the MkII and IFPUG methods
- to foster consistency in the application of the methods, and thus comparability of results
- to promote the use of the Functional Size as a key component of software metrics
- to provide a reference point for interpretation
- to keep the user community informed of any developments via the UKSMA Committee and general meetings.

1.8 Procedure for raising a Query or Issue with the MPC

Should the reader wish to comment or need advice on any aspect of this manual, please contact the UKSMA Administrator. Suggestions for improvement to this document, and UKSMA services are warmly welcomed!

(Blank page)

2

The Mk II Function Point Analysis Rules

Below are listed the rules to which all Mk II Function Point Counts must conform. General guidelines to assist in conformance to these rules are given in Chapter 4 of the manual.

Rule 1 Boundary

1.1 Mk II FPA is used to measure the size of the functionality required by the users of an application, within a boundary defined for the purpose of the FP count.

1.2 The application or part of the application enclosed by the boundary must be a coherent body of functionality, comprising one or more complete Logical Transaction Types. (In the following, 'Type' is dropped for ease of reading.)

Rule 2 Functional Size and Logical Transactions

2.1 The Functional Size of an application is the sum of the sizes of each of the Logical Transactions whose input and output components cross the enclosing boundary.

2.2 A Logical Transaction is the lowest level of self-consistent process. It consists of three components; input across an application boundary, processing involving stored data within the boundary, and output back across the boundary.

2.3 It is triggered by a unique event that is of interest to the user, which, when completed, leaves the application in a self-consistent state in relation to the unique event. It may also be triggered by a user requirement to extract information from the application, leaving it unchanged.

2.4 A Logical Transaction is counted once when sizing an application, even though it may be executed from more than one point in the application.

2.5 Changes to are counted by summing the size of the added, changed, and deleted Logical Transactions.

2.6 In the case of changed Logical Transactions, the size counted is the size of the changes made to the Logical Transactions, not the overall size of the Logical Transactions. The size of the changes made is found by summing the counts of the added, changed and deleted input, process and output components of the Logical Transactions.

Rule 3 Processing Component of Logical Transactions

3.1 The processing component of a Logical Transaction is analysed by reference to its manipulation (i.e. create, update, delete, or read) of stored data.

3.2 The processing component is sized by counting the number of Primary Entity Types that are referenced by the Logical Transaction, plus the System Entity if referenced. See Section 4.4 for definitions of 'Entity-Type' and 'Primary'.

3.3 All accesses to Non-Primary Entity Types within the entity model are considered to be references to a single entity called the System Entity.

3.4 Within an application boundary there can be only one System Entity, to which a maximum of one reference may be included in the processing component of a Logical Transaction.

Rule 4 Input and Output Components of Logical Transactions

4.1 The input and output components of a Logical Transaction are sized by counting the number of Data Element Types crossing the application boundary, via each component respectively.

4.2 A Data Element Type is a uniquely processed item of information that is indivisible for the purposes of the Logical Transaction being sized. It is part of an input or output data flow across an application boundary.

Rule 5 Logical Transaction Size

5.1 The Functional Size of a Logical Transaction is the weighted sum of the input, processing, and output components of the Logical Transaction.

5.2 The industry standard weights are as follows: Input Weight is 0.58 (per Input Data Element Type), Processing Weight is 1.66 (per Entity Type Reference), and the Output Weight is 0.26 (per Output Data Element Type).

Rule 6 Reporting a MkII Function Point Count

6.1 A MkII FP count obtained according to the rules of this Counting Practices Manual should include reference to the CPM Version Number when it is reported, for example:

'557 MkII Function Points (ISO/IEC 20968:2002)'