



**COSMIC Measurement Manual
for ISO 19761**

**Part 1:
Principles, Definitions & Rules**

Version 5.0

August 2021

Foreword

Much of modern society depends on software and the importance of regulated software measurement has never been greater. Software functional sizing is a reliable and consistent means of estimating, planning and benchmarking software work.

The COSMIC measurement method is the modern evolved standardized way of measuring functional size of software. It evolved in the late 1990's, as a rework of the earlier function point sizing methods and it was endorsed as an ISO engineering standard in 2003.

COSMIC sizing is applicable to sizing all types of software. The methodology is equally suited to modern agile and scaled agile approaches to software development as traditional approaches.

The Measurement Manual

The COSMIC Measurement Manual describes the core measurement methodology. This version 5.0 shortens version 4.0.2 while not modifying its substance in terms of measurement definitions, rules and guidance. This manual consists of three parts:

Part 1: Principles, definitions & rules* (17 pages)

Part 2: Guidelines* (18 pages)

Part 3: Examples of COSMIC concepts and measurements, consisting of:

Part 3a Standard Measurement Strategy Examples (13 pages)

Part 3b Real-time Examples (32 pages)

Part 3c MIS Examples. (58 pages)

* Parts 1 and 2 describe the entire material necessary for certification.

Further explanations, guidelines, translations, practical examples and other publications are available from www.cosmic-sizing.org.

Editors:

Alain Abran, Ecole de technologie supérieure – University of Quebec (Canada),

Peter Fagg, Pentad (UK),

Arlan Lestherhuis (The Netherlands).

Other members of COSMIC Measurement Practices Committee:

Jean-Marc Desharnais, Ecole de technologie supérieure – University of Quebec (Canada),

Dylan Ren, Measures Technology LLC (China),

Bruce Reynolds, Tecolote Research (USA),

Hassan Soubra, German University in Cairo (Egypt),

Sylvie Trudel, Université du Québec à Montréal - UQAM (Canada),

Frank Vogelesang, Metri (The Netherlands).

August 2021 minor editing:

In the Foreword the names of the Parts have been updated. No other changes.

Copyright 2021. All Rights Reserved. The Common Software Measurement International Consortium (COSMIC). Permission to copy all or part of this material is granted provided that the copies are not made or distributed for commercial advantage and that the title of the publication, its version number, and its date are cited and notice is given that copying is by permission of the Common Software Measurement International Consortium (COSMIC). To copy otherwise requires specific permission.

Table of Contents

1. INTRODUCTION.....	4
1.1 Purpose.....	4
1.2 Overview.....	4
1.3 The models and principles of the COSMIC method.....	4
1.4 Definitions.....	5
2. COSMIC MEASUREMENT PROCESS.....	9
3. MEASUREMENT STRATEGY PHASE.....	9
3.1 Deriving the measurement strategy from the software context model.....	9
3.2 Determination of the purpose and scope of the FSM.....	10
3.3 Identification of the FUR.....	10
3.4 Identification of the layers.....	10
3.4.1 <i>The scope of the FSM and layers</i>	10
3.4.2 <i>Characteristics of layers</i>	11
3.5 Identification of the functional users.....	11
3.6 Identification of software boundaries.....	11
4. MAPPING PHASE.....	12
4.1 General – Mapping the FUR to the Generic Software Model.....	12
4.2 Identification of functional processes.....	12
4.3 Identification of objects of interest and data groups.....	13
4.4 Identification of data movements.....	13
4.5 Classification of data movements.....	14
5. MEASUREMENT PHASE.....	15
5.1 The measurement phase process.....	15
5.2 Calculation of the functional size.....	16
6. MEASUREMENT REPORTING.....	17
REFERENCES.....	17

1. INTRODUCTION.

1.1 Purpose.

The purpose of this document is to state the Principles, Definitions and Rules of the COSMIC Functional Size Measurement method (the 'COSMIC Method'), as well as the COSMIC measurement process.

This Part 1 of the COSMIC Measurement Manual document contains only reference material, i.e. **what** to do, as described in ISO 19761 [3]. For guidance developed by the COSMIC Group as to **how** to apply COSMIC to different situations refer to Part 2, and for **examples**, refer to Part 3. The COSMIC Group has also published additional documents to illustrate its use in various contexts contexts (Agile, Business Applications, Real-time, etc.) and technologies (SOA, Mobile, etc.)

1.2 Overview.

The COSMIC method involves applying a set of models, principles, rules and processes to measure the Functional User Requirements (or 'FUR') of a given piece of software.

The result is a numerical 'value of a quantity' (as defined by ISO) representing the functional size of the piece of software according to the COSMIC method. This numerical value is on a ratio scale: therefore valid mathematical operations can be performed using the values.

The COSMIC method adopts the ISO definition of Functional User Requirements (FUR).

1.3 The models and principles of the COSMIC method.

The COSMIC method is based on software engineering principles categorized in two models:

The COSMIC Software Context Model: it contains the principles that pertain to identifying the nature and structure of the Software to be measured as required by the COSMIC method, leading to the identification of its FUR.

The Generic Software Model: it contains the principles to be applied to the FUR in order to extract and measure the elements that contribute to the functional size using the COSMIC method.

PRINCIPLES - The COSMIC Software Context Model.

1. A software application is typically structured into layers.
2. A layer may contain one or more separate pieces of software.
3. A piece of software is described by its Functional User Requirements (FUR).
4. The FUR are expressed at a level of granularity that exposes its functional processes.
5. A piece of software delivers functionality to its functional users as identified in the FUR.
6. A piece of software to be measured is defined by the scope of the functional size measurement (FSM), which is confined wholly within a single layer.
7. The scope of the FSM defines the functional processes to be measured, and depends on the purpose of the measurement.

PRINCIPLES - The COSMIC Generic Software Model.

1. A piece of software interacts with its functional users across a boundary, and with persistent storage within this boundary.
2. A functional process consists of sub-processes called data movements.
3. There are four data movement sub-types: Entry, Exit, Write and Read. A data movement sub-type includes any associated data manipulation.
4. A data movement moves a single data group.
5. A data group consists of a unique set of data attributes that describe a single object of interest.
6. Each functional process is initiated by a triggering event, detected by a Functional User and which in turn initiates a data movement called the triggering Entry.
7. The functional size is based on the types of the elements used for measurement, not on the number of occurrences.
8. The size of a functional process is equal to the number of its data movements where one data movement has a size of 1 COSMIC Function Point.
9. The size of a piece of software is the sum of the sizes of the functional processes within the scope of the FSM.

NOTE: The principles are written using the terminology of the COSMIC method.

1.4 Definitions.

In this sub-section:

- the definitions from ISO documents are reproduced 'as is' but without the ISO NOTES that have been transferred to the main body of the text;
- texts underlined refer to terms defined in this sub-section.

application.

software system for collecting, saving, processing, and presenting data by means of a computer. [Adapted from [4]]

Base Functional Component (BFC).

elementary unit of Functional User Requirements defined by and used by an FSM method for measurement purposes. [1]

boundary.

conceptual interface between the software being measured and its functional users.

component.

any part of a software system that is separate for reasons of the software architecture, and/or that was specified, designed or developed separately.

control command.

command that enables human functional users to control their use of the software but which does not involve any movement of data about an object of interest of the FUR of the software being measured.

COSMIC unit of measurement.

1 CFP (COSMIC Function Point), which is defined as the size of one data movement.

data attribute.

smallest parcel of information, within an identified data group, carrying a meaning from the perspective of the software's Functional User Requirements. [3]

data group.**data group type.**

distinct, non-empty, non-ordered and non redundant set of data attributes where each included data attribute describes a complementary aspect of the same one object of interest. [3]

data manipulation.

any processing of the data other than a movement of the data into or out of a functional process, or between a functional process and persistent storage. [3]

data movement.**data movement type.**

Base Functional Component which moves a single data group. [3]

E - Abbreviation for Entry type.

Entry.**Entry type.**

data movement that moves a data group from a functional user across the boundary into the functional process where it is required. [3]

error/confirmation message.

Exit issued by a functional process to a human user that either confirms only that entered data has been accepted, or only that there is an error in the entered data.

Exit.**Exit type.**

data movement that moves a data group from a functional process across the boundary to the functional user that requires it. [3]

event.

something that happens.

functional process.**functional process type.**

elementary component of a set of Functional User Requirements, comprising a unique, cohesive and independently executable set of data movements. [3]

functional process level of granularity.

level of granularity of the description of a piece of software at which

- the functional user (-types) are individual humans or engineered devices or pieces of software (and not any groups of these) AND
- single event (-types) occur that the piece of software must respond to (and not any level of granularity at which groups of events are defined).

functional size.

size of the software derived by quantifying the Functional User Requirements. [1]

Functional Size Measurement

FSM.

process of measuring Functional Size. [1]

Functional Size Measurement method.

FSM method

specific implementation of FSM defined by a set of rules, which conforms to the mandatory features of ISO/IEC 14143-1:2007. [1]

functional user.

user that is a sender and/or an intended recipient of data in the Functional User Requirements of a piece of software. [3]

Functional User Requirements.

FUR.

sub-set of the user requirements describing what the software shall do, in terms of tasks and services. [1]

input.

data moved by all the Entries of a given functional process.

layer.

partition resulting from the functional division of a software system. [3]

level of decomposition.

any level resulting from dividing a piece of software into components (named 'Level 1', for example), then from dividing components into sub-components ('Level 2'), then from dividing sub-components into sub-sub components (Level 3'), etc.

level of granularity.

any level of expansion of the description of any part of a single piece of software (e.g. a statement of its requirements, or a description of the structure of the piece of software) such that at each increased level of expansion, the description of the functionality of the piece of software is at an increased and uniform level of detail.

measurement method

logical sequence of operations, describe generically, used in the performance of measurements. [5]

measurement process.

process of establishing, planning, performing and evaluating software measurement within an overall project or organizational measurement structure. [3]

measurement (strategy) patterns.

standard template that may be applied when measuring a piece of software from a given software functional domain, that defines the types of functional user that may interact with the software, the level of decomposition of the software and the types of data movements that the software may handle.

model.

description or analogy used to help visualize a concept that cannot be directly observed.

OOI - Abbreviation for object of interest

object of interest.

object of interest type.

any 'thing' that is identified from the point of view of the Functional User Requirements about which the software is required to process and/or store data. [3]

output.

data moved by all the Exits of a given functional process.

peer software.

piece of software that reside in the same layer as, and exchanges data with, another piece of software. [3]

persistent storage.

storage which enables a functional process to store data beyond the life of the functional process and/or which enables a functional process to retrieve data stored by another functional process, or stored by an earlier occurrence of the same functional process, or stored by some other process. [3]

purpose of measurement.

statement that defines why a measurement is being made, and what the result will be used for.

R – Abbreviation for Read type.

Read.**Read type.**

data movement that moves a data group from persistent storage within reach of the functional process which requires it. [3]

scope.**scope of the FSM.**

set of Functional User Requirements to be included in a specific functional size measurement instance. [3]

software

set of computer instructions, data, procedures and maybe documentation operating as a whole, to fulfill a specific set of purposes, all of which can be described from a functional perspective through a finite set of Functional User Requirements, technical and quality requirements.

software system.

system that consists only of software.

sub-process type.

part of a functional process that either moves data (into the software from a functional user or out of the software to a functional user, or to or from persistent storage) or that manipulates data.

system.

combination of hardware, software and manual procedures organized to achieve stated purposes. [adapted from [2]]

triggering Entry.**triggering Entry type.**

the Entry data movement of a functional process that moves a data group generated by a functional user that the functional process needs to start processing.

triggering Event.**triggering event type.**

event that causes a functional user of the piece of software to initiate ('trigger') one or more functional processes. [3]

unit of measurement.

particular quantity, defined and adopted by convention, with which other quantities of the same kind are compared in order to express their magnitudes relative to that quantity. [5]

user

any person or thing that communicates or interacts with the software at any time. [3]

value (of a quantity)

magnitude of a particular quantity, generally expressed as a unit of measurement multiplied by a number.

W - Abbreviation for Write type.

Write.

Write type.

data movement that moves a data group from a functional process to persistent storage. [3]

X – Abbreviation for Exit type.

See Exit.

2. COSMIC MEASUREMENT PROCESS.

The COSMIC measurement process consists of three phases – see Figure 2.1:

- the Measurement Strategy phase, in which the purpose and scope of the FSM are defined. The Software Context Model is then applied so that the software to be measured and the required measurement are unambiguously defined – see Section 3.
- the Mapping Phase in which the Generic Software Model is applied to the FUR of the software to be measured to produce the COSMIC model of the software that can be measured – see Section 4.
- the Measurement Phase, in which actual sizes are assigned - see Section 5.

Rules for how measurements should be recorded are given in Section 6.

The relationship of the three phases of the COSMIC method is shown in Figure 2.1.

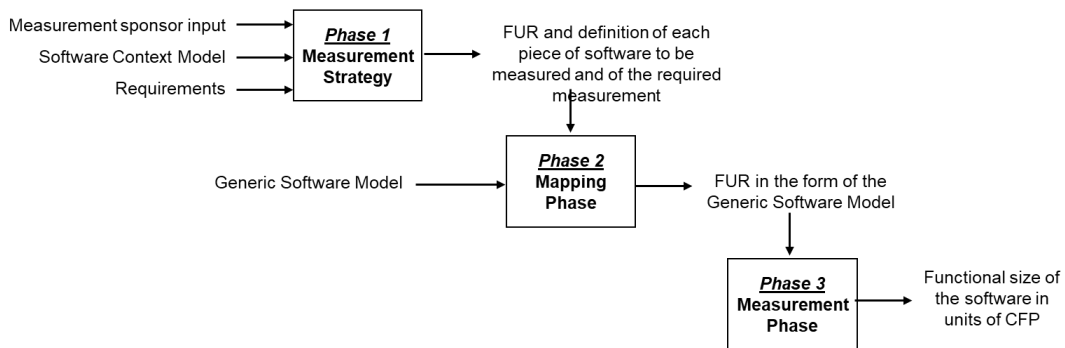


Figure 2.1 – The COSMIC method measurement process.

3. MEASUREMENT STRATEGY PHASE.

3.1 Deriving the measurement strategy from the software context model.

This section describes the key parameters that must be considered in the Measurement Strategy phase before actually starting to measure. The sub-sections give the rules to help the Measurer through the process of determining a measurement strategy, as shown in Figure 3.1.

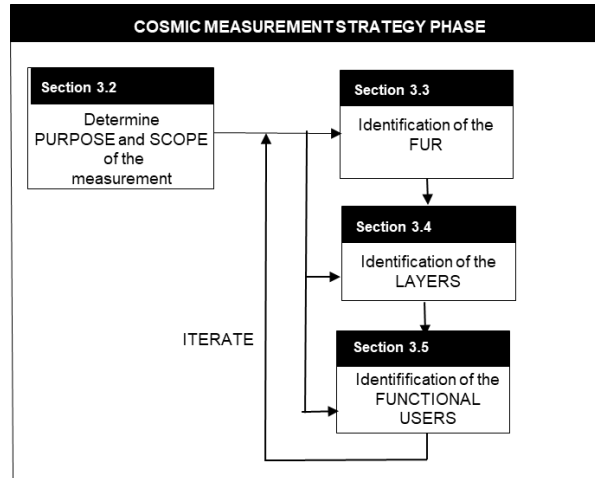


Figure 3.1 - The process of determining a Measurement Strategy.

RULE 1: Measurement activities.

The determination of the COSMIC functional size **shall** involve all of the activities and rules described in Sections 3.2 to 3.6.

3.2 Determination of the purpose and scope of the FSM.

RULE 2: Purpose and scope.

The purpose and the scope of the FSM **shall** be determined before commencing a particular measurement exercise.

NOTE: Once the purpose of the FSM has been determined, the process of determining the scope(s) of the FSM, the functional users, the layers and the boundaries may require some iterations.

3.3 Identification of the FUR.

RULE 3: Identification of the FUR.

The FUR identified to be within the scope of the FSM **shall** be used as the exclusive source from which the functional size of the software is to be measured.

NOTE: The term 'FUR' means those functional user requirements that are completely defined so that a COSMIC functional size measurement is possible.

3.4 Identification of the layers.

3.4.1 The scope of the FSM and layers

Software may have components of its functionality that exist in different layers of the software's operating environment.

RULE 4: If required for the purpose of the measurement exercise, each such layer **shall** be identified.

RULE 5: A single piece of software to be measured **shall not** have its scope defined to extend over more than one layer.

NOTE 1: FUR may state explicitly, may imply, or the measurer may infer, that the FUR apply to software in different layers or to different peer items whose size must be measured separately. Alternatively, the measurement analyst may be faced with sizing existing software which appears to be in different layers or to consist of separate peer items. In both cases, guidance is needed to help decide if the FUR of the software comprise one or more layers or peer items.

NOTE 2: Layer identification is an iterative activity. The exact identification of layers can be refined as the measurement activity progresses.

3.4.2 Characteristics of layers

RULE 6: Characteristics of layers.

Layers identified within the scope of the FSM **shall** have the following characteristics:

- a) Software in each layer shall deliver functionality to its functional users.
- b) Software in a subordinate layer shall provide functional services to software in a layer using its services.
- c) Software that shares data with other software shall not be considered to be in different layers if they identically interpret the data attributes that they share.

3.5 Identification of the functional users.

RULE 7: Functional Users.

All functional users that trigger, provide information to, or receive information from functional processes in the FUR of the software within the scope of the FSM **shall** be identified.

NOTE 1: This rule above corrects an omission in ISO 19761. An amendment to the standard is in preparation.

NOTE 2: As persistent storage is on the software side of the boundary, it is not considered to be a functional user of the software being measured.

3.6 Identification of software boundaries

RULE – Identification of boundaries.

RULE 8: The boundary of each piece of software within each layer and in the scope of the FSM **shall** be identified.

RULE 9: Once the boundaries have been identified, each FUR within the scope of the FSM shall be allocated to a piece of software.

4. MAPPING PHASE.

4.1 General – Mapping the FUR to the Generic Software Model.

Figure 4.1 shows the steps of the process for mapping the FUR as in the available software artefacts to the form required by the COSMIC Generic Software Model.

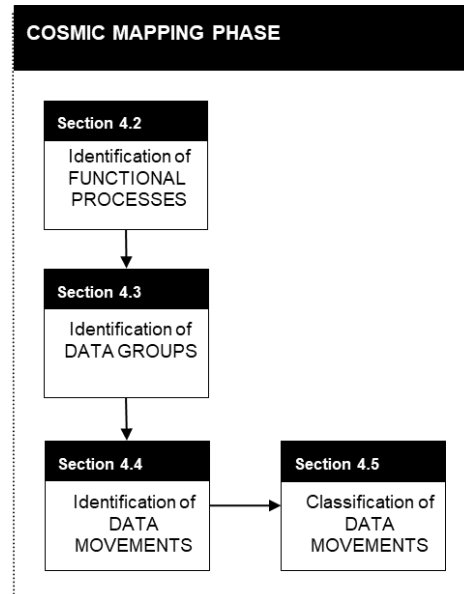


Figure 4.1 –The process of the COSMIC Mapping Phase.

4.2 Identification of functional processes.

RULE 10: Identification of functional processes.

Each functional process identified in the scope of the FSM **shall**:

- be derived from at least one identifiable FUR,
- be initiated by an Entry data movement from a functional user informing the functional process that it has detected a triggering event,
- comprise at least two data movements, namely always one Entry plus either an Exit or a Write.
- belongs to one, and only one layer,
- be complete when a point of asynchronous timing is required to be reached according to its FUR.

NOTE 1: the COSMIC Group has subsequently clarified sub-clause e) above as the equivalent of the following statement: 'the set of all data movements that is needed to meet its FUR for all the possible responses to its triggering Entry'.

NOTE 2: The Generic Software Model is a logical model. A functional process occurrence may start processing before data has been entered e.g. when a human user clicks on a menu to display a blank screen for data entry.

NOTE 3: In a set of FUR, each event which causes a functional user to trigger a functional process

- cannot be sub-divided for that set of FUR,
- has either happened or it has not happened.

4.3 Identification of objects of interest and data groups.

RULE 11: Identification of objects of interest and data groups.

Each data group identified in the scope of the FSM **shall**:

- a) be unique and distinguishable through its unique collection of data attributes,
- b) be directly related to one object of interest described in the software's FUR.

NOTE 1: An object of interest can be any physical thing, as well as any conceptual thing or part of a conceptual thing in the world of a functional user.

NOTE 2: Examples of 'thing' include, but are not limited to, software applications, humans, sensors, or other hardware.

NOTE 3: The term object of interest is used in order to avoid terms related to specific software engineering methods. The term does not imply objects in the sense used in Object Oriented methods. Similarly, the word Entity is avoided because of its use in Data Modelling.

NOTE 4: Constants or variables which are internal to the functional process, or intermediate results in a calculation, or data stored by a functional process resulting only from the implementation, rather than the FUR, are not data groups.

4.4 Identification of data movements.

This step consists in identifying the data movements (Entry, Exit, Read and Write) of each functional process. Figure 4.2 illustrates the overall relationship between the four types of data movement, the functional process to which they belong and the boundary of the measured software.

RULE 12: Identification of data movements.

Each functional process identified in 4.2 **shall** be partitioned into its component data movements.

NOTE: The COSMIC method defines a data movement type as a BFC.

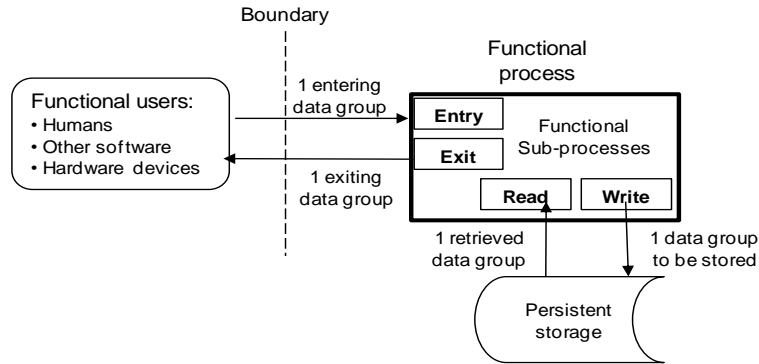


Figure 4.2 – The four types of data movement & their relationship with a functional process.

RULE 13: Functional Process – Single Entry.

For any one functional process, a single Entry data movement **shall** be identified and counted for the entry of all data describing a single object of interest that the FUR require to be entered, unless the FUR explicitly require data describing the same single object of interest to be entered more than once in the same functional process.

RULE 14: Functional Process – Single Exit, Read or Write.

Similarly, a single Exit, Read or Write data movement **shall** be identified and counted for the movement of all data describing a single object of interest that the FUR requires of that type (e.g. Exit, Read or Write, respectively), unless the FUR explicitly require data describing the same single object of interest to be moved more than once in the same functional process by a data movement of the same type (e.g. Exit, Read or Write, respectively).

RULE 15: Functional Process – Occurrences.

If a data movement of a particular type (Entry, Exit, Read or Write) occurs multiple times with different data values when a functional process is executed, only one data movement of that type **shall** be identified and counted in that functional process.

4.5 Classification of data movements.

RULE 16: Entry.

An Entry **shall**:

- a) receive a single data group which originates from the functional user side of the boundary,
- b) account for all required formatting and presentation manipulations along with all associated validations of the entered data attributes, to the extent that these data manipulations do not involve another type of data movement.

NOTE: An Entry accounts for all manipulations that might be required to validate some entered codes or to obtain some associated descriptions. However, if one or more Reads are required as part of the validation process, these are identified and counted as separate Read data movements.

- c) include any 'request to receive the Entry data' functionality, where it is unnecessary to specify what data should be entered.

RULE 17: Exit.

An Exit **shall**:

- a) send data attributes from a single data group to the functional user side of the boundary,
- b) account for all required data formatting and presentation manipulations, including processing required to send the data attributes to the functional user, to the extent that these manipulations do not involve another type of data movement.

RULE 18: Read.

A Read **shall**:

- a) retrieve a single data group from persistent storage.
- b) account for all logical processing and/or mathematical computation needed to read the data, to the extent that these manipulations do not involve another type of data movement,
- c) include any 'request to read' functionality.

RULE 19: Write.

A Write **shall**:

- a) move data attributes from a single data group to persistent storage.
- b) account for all logical processing and/or mathematical computation to create the data attributes to be written, to the extent that these manipulations do not involve another type of data movement.

RULE 20: Write – Delete.

A requirement to delete a data group from persistent storage shall be a single Write data movement.

5. MEASUREMENT PHASE.

5.1 The measurement phase process.

The general method for measuring a piece of software when its FUR have been expressed in terms of the COSMIC Generic Software Model is summarized in Figure 5.1.

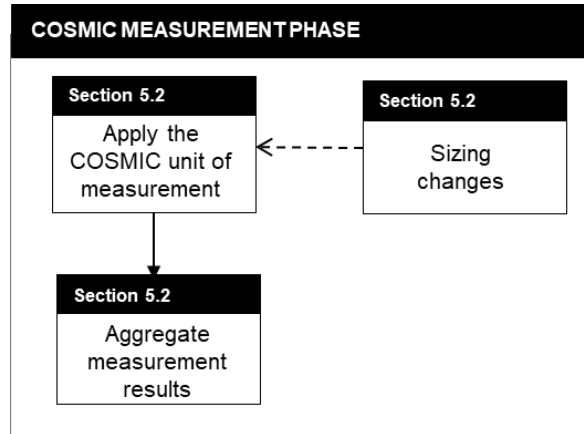


Figure 5.1 – The process of the COSMIC Measurement Phase.

5.2 Calculation of the functional size.

RULE 21: Size of a data movement.

A unit of measurement, 1 CFP¹, **shall** be assigned to each data movement (Entry, Exit, Read or Write) identified in each functional process.

RULE 22: Size of a functional process.

The results of 5.2, as applied to all identified data movements within the identified functional process, **shall** be aggregated into a single functional size value for that functional process by:

- multiplying the number of data movements of each type by its unit size,
- totaling the sizes from step a) for each of the data movement types in the functional process.

Size (functional process) = Σ size(Entries) + Σ size(Exits) + Σ size(Reads) + Σ size(Writes).

RULE 23: Functional size of the identified FUR of each piece of software to be measured.

The size of each piece of software to be measured within a layer **shall** be obtained by aggregating the size of the functional processes within the identified FUR for each piece of software.

NOTE: Within each identified layer, the aggregation function is fully scalable. Therefore a sub-total can be generated for individual functional processes, individual software pieces or for the whole layer, depending on the purpose and scope of the FSM.

RULE 24: Functional size of changes to the FUR.

Within each identified layer, the functional size of changes to the FUR within each piece of software within the scope of the FSM **shall** be calculated by aggregating the sizes of the corresponding impact data movement according to the following formula:

¹The unit of measurement was known as a 'Cfsu' (COSMIC functional size unit) prior to v3.0 of COSMIC method.

$$\begin{aligned} \text{Size (Changes to a piece of software)} = & \Sigma \text{ size (added data movements)} + \\ & \Sigma \text{ size (changed data movements)} + \\ & \Sigma \text{ size (deleted data movements)} \end{aligned}$$

summed over all functional processes for the piece of software.

NOTE: A data movement is considered to be changed if any of the attributes of the data group are changed, or if any changes are needed to the data manipulation associated with the data movement.

6. MEASUREMENT REPORTING.

The result must be reported and data about the measurement recorded so as to ensure that the result is always unambiguously interpretable.

RULE 25: Labelling.

A COSMIC measurement result on the FUR for a piece of software that conforms to the mandatory rules of ISO 19761 **shall** be labelled according to the following convention:

CFP (ISO/IEC 19761:2011).

RULE 26: Documentation of the measurement results.

The documentation of the COSMIC measurement results **shall** include the following information:

- a) Identification of each piece of software in the scope of the FSM (name, version identification or configuration identification).
- b) A description of the measurement purpose and scope.
- c) A description of the relationship of each piece of software within the scope of the FSM with its functional users, both peer-to-peer and between layers.
- d) The functional size of each piece of software within the scope of the FSM, calculated according to 5.2 and reported according to 6.

NOTE: Documentation is required for each piece of software measured within each layer.

REFERENCES.

[1] ISO/IEC 14143-1: 2007 Information technology – Software measurement – Functional size measurement – Part 1: Definition of concepts, International Organization for Standardization – ISO, Geneva, 2017.

[2] ISO/IEC 15288: 2008 Systems and Software Engineering – System Life Cycle Processes, International Organization for Standardization – ISO, Geneva, 2008.

[3] ISO/IEC 19761: 2011 Software engineering – COSMIC: a functional size measurement method, International Organization for Standardization – ISO, Geneva, 2011.

[4] ISO/IEC 24570:2005 Software engineering -- NESMA functional size measurement method version 2.1, International Organization for Standardization – ISO, Geneva, 2010.

[5] ISO Guide 99: 1993, International vocabulary of basic and general terms in metrology (VIM), International Organization for Standardization – ISO, Geneva, 2019.