The COSMIC Functional Size Measurement Method

Version 4.0.1

Guideline on how to convert ‘First Generation’ Function Point sizes to COSMIC sizes

Version 1.0

November 2016

Acknowledgements

Guideline authors and reviewers 2016 (alphabetical order)		
Alain Abran École de Technologie Supérieure, Université du Québec Canada	Diana Baklizky ti MÉTRICAS Brazil	Jean-Marc Desharnais École de Technologie Supérieure – Université du Québec Canada
Cigdem Gencel* Free University of Bozen-Bolzano, Italy	Luigi Lavazza Università dell'Insubria Italy	Arlan Lesterhuis* The Netherlands
Luca Santillo Agile Metrics Italy	Federica Sarro University College London United Kingdom	Charles Symons* COSMIC United Kingdom
Frank Vogelezang Ordina The Netherlands	Chris Woodward QuantiMetrics United Kingdom	

* Authors of this Guideline version1.0

Version Control

The following table summarizes the changes to this ’Guideline’.

DATE	REVIEWER(S)	Modifications / Additions
2007	COSMIC Measurement Practices Committee	The first version of the contents of this Guideline appeared in Chapter 3 of the ‘Advanced & Related Topics’ document of the COSMIC method version 3.0, December 2007. This Guideline replaces that Chapter for v4.0.1 of the COSMIC method.
November 2016	COSMIC Measurement Practices Committee	First public version 1.0 of this document

Foreword

‘First generation’ Functional Size Measurement (FSM) methods such as the IFPUG, MkII and Nesma methods and the ‘second generation’ COSMIC method all express their sizes in some variety of ‘Function Points’. However, the measurements are not directly comparable as each method has its own measurement rules and processes and hence different scales of measurement.

An organization wishing to move from using a first generation FSM method to using the COSMIC method will have two common questions:

- Is there any formula to convert from one size scale to the COSMIC scale?
- How accurate are the converted sizes likely to be (to help decide e.g. how to use the converted data to re-calibrate an estimating method)?

This Guideline aims to help answer this question, by describing:

- the nature of the relationship between the size scales of first generation and COSMIC methods;
- processes to establish a statistical relationship between first generation and COSMIC method sizes from a sample of software items that can be used for converting the first generation sizes of software items;
- the results of studies to derive statistical conversion formulae from several sets of measurements of software sizes by first generation and by COSMIC methods;
- other non-statistical methods of converting sizes measured by a first generation FSM method to COSMIC sizes.

Scope of applicability of this Guideline

The methods described in this Guideline are applicable for converting first generation FSM method sizes to COSMIC sizes for ‘whole business applications’ and for converting sizes of whole chunks of new functionality delivered by projects to enhance those applications. (In this Guideline these two types are referred to as ‘software items’.) The methods are *not* applicable for converting:

- sizes of components of distributed business applications;
- sizes of any software items from domains for which first generation methods were not designed, e.g. real-time, infrastructure or algorithmic-intensive software.

The methods *may not* be applicable for converting IFPUG or Nesma sizes to COSMIC sizes delivered by projects to enhance applications that require a mixture of added, changed and deleted functionality (section 1.2.4 provides the reasons).

‘Total’ versus ‘component’ size conversion methods

Unfortunately exact conversion of sizes measured with a first generation FSM method to COSMIC sizes is not possible due to a number of theoretical reasons (see Chapter 1 for details). Most of this Guideline is therefore devoted to describing the commonest methods for an organization to convert sizes of whole software applications measured by a first generation method to COSMIC sizes by establishing a statistical relationship between their total sizes. We call this ‘total size conversion’.

More accurate converted sizes should be possible than can be achieved by a ‘total size conversion’ method by using a ’component size conversion’ method. Such a method works by converting at the level of components of the total sizes, rather than by converting at the level of the total sizes. However, this method requires some detailed raw data for the first generation measurements, which not all organizations will have recorded. So far, a component size conversion method has been tested (very successfully) only for MkII to COSMIC size conversion.

Structure of this Guideline

This Guideline is organized as follows:

- Chapter 1 discusses the similarities and differences between these FSM methods and discusses the implications for any conversion exercise that an organization may undertake.
- Chapter 2 describes the total size conversion method.
- Chapter 3 describes a ‘direct’ process for manual conversion of IFPUG to COSMIC sizes, and a more recent ‘component size conversion’ method and how it has been successfully applied to conversion of MkII FP sizes to COSMIC sizes. This Chapter also outlines how a component size

conversion method might be adapted to IFPUG/Nesma to COSMIC conversion, noting that the method has not yet been tested in practice.

- Appendix A presents a summary, and Appendix B the details, of the results of studies in the literature to apply total size conversion processes to IFPUG/Nesma-to-COSMIC, and MkII-to-COSMIC size conversion.
- Appendix C describes some recent research that suggests a way of converting an effort estimation formula based on first-generation sizes, to a formula based on COSMIC sizes without first needing to establish the correlation between the two sets of sizes.
- Appendix D has a summary of abbreviations used in this Guideline and a glossary of terms specific to this Guideline.

The Measurement Manual [1] contains the main Glossary of terms of the COSMIC method. For terms of the IFPUG, MkII and Nesma methods, see their respective manuals.

This document is developed from and replaces Chapter 3 of the COSMIC 'Advanced and Related Topics' document of December 2007 [2].

How to use this Guideline

We assume the reader works for an organization that has many software size measurements made by a first generation FSM method, is experienced in that method, and wishes to convert these measurements to equivalent COSMIC sizes. There are a few vital points to consider before starting a conversion exercise.

- Do ensure you have a good understanding of how the first generation FSM method and the COSMIC method measure functional sizes, where they are similar, and what are the significant differences. Chapter 1 describes these essential features and differences. For a general introduction to the COSMIC method see [35].
- Do first decide on the objectives of the conversion exercise and, in particular, what is the desired accuracy for the converted COSMIC sizes. Your choice here will help decide which conversion approach to use. For more on this, read Section 2.1 'Conversion objectives' and the Conclusions sections of Chapters 2 and 3 before deciding which approach to use.
- Whichever conversion approach you adopt we strongly recommend an organization to apply one or more of the methods described in this Guideline to its OWN data. Do NOT simply use any of the conversion formulae developed by other researchers based on their data that are reproduced in this Guideline in Appendices A and B[1].
- Most conversion processes described in this Guideline rely on statistical analyses, e.g. of the correlation of sizes measured by both the first generation method and by the COSMIC method. However, do NOT rely only on statistical tests to derive a conversion formula. Do use your understanding of the basic differences between how the first generation and the COSMIC methods measure size, for example so that you can deal correctly with any outliers in the data.

We also assume the reader has a basic understanding of how to use spreadsheets to establish Ordinary Least Squares (OLS) relationships between the two sets of sizes and to judge the significance of the resulting R-squared (or 'coefficient of determination') value. (For more on this see section 2.2, Step 4). Other validity tests that require a deeper statistical knowledge are also mentioned in footnotes.

COSMIC Measurement Practices Committee

[1] The reason for this recommendation is that the nature of the functionality of a particular organization's software may or may not match that of the applications used for the published studies. This means that the accuracy of sizes converted by any of the formulae given in Appendices A and B will be unknown. For more on this recommendation, see for example section 3.3.1.

Table of Contents

1 CONVERTIBILITY OF FUNCTIONAL SIZES: THEORY AND PRACTICE 6
1.1 Similarities and differences between FSM methods 6
1.2 Implications of differences of FSM Methods on defining conversion formulas 10
1.2.1 Non-matching BFC types 10
1.2.2 Measurement scale types: size bounds 11
1.2.3 Scope of measurement: 'layer' and 'level of decomposition' 12
1.2.4 Project development type (new development / enhancement) 12
2 CONVERSION OF TOTAL FIRST GENERATION FSM METHOD SIZES TO COSMIC SIZES 13
2.1 Conversion objectives 13
2.2 The total size conversion method 13
2.3 Detection of outlier observations; an example analysis 18
2.4 Conclusions and recommendations for use of total size conversion methods 23
3 'DIRECT' AND 'COMPONENT' METHODS FOR CONVERSION 24
3.1 The limitations of statistical conversion methods 24
3.2 A process to convert IFPUG/Nesma to COSMIC sizes directly 24
3.2.1. Direct conversion of sizes of applications or of additions of whole functionality 25
3.2.2. Direct conversion of enhancements to add, modify and delete existing functionality 25
3.3 'Component' size conversion methods 25
3.3.1 Examining the homogeneity of the data to be converted 27
3.3.2 The component size conversion method applied to MkII to COSMIC size conversion 29
3.3.3. Summary description of the component size conversion method 31
3.4. Adapting the component size conversion method for IFPUG/Nesma to COSMIC conversion 32
3.5 Conclusions on direct and component size conversion methods 33

REFERENCES 34

APPENDIX A – REVIEW OF SIZE CONVERSION STUDIES 36
A.1 IFPUG/Nesma to COSMIC conversion formulas in the literature 36
A.1.1 Converting IFPUG/Nesma total (i.e. transaction PLUS data) sizes 36
A.1.2 Converting IFPUG/Nesma transaction sizes 41
A.2 MkII to COSMIC total size conversion formulas in the literature 42

APPENDIX B – DETAILED RESULTS OF STATISTICAL CONVERSION STUDIES 44

APPENDIX C – FPA TO COSMIC MIGRATION IN ORDER TO ESTIMATE WEB APPLICATION DEVELOPMENT EFFORT 50

APPENDIX D – ABBREVIATIONS AND GLOSSARY OF TERMS 51

APPENDIX E - COSMIC CHANGE REQUEST AND COMMENT PROCEDURE 53

1

CONVERTIBILITY OF FUNCTIONAL SIZES: THEORY AND PRACTICE

Before attempting any size conversion exercise, it is essential to have a good understanding of the generic Functional Size Measurement ('FSM') process, the similarities and differences between different FSM methods and the limitations and conditions on using any conversion method. Otherwise, wrong assumptions may lead to significant errors when using a converted size for e.g. performance measurement or effort estimation.

1.1 Similarities and differences between FSM methods

FSM methods use different terminology for the same or very closely related concepts [21].

The Nesma method is a variant of the IFPUG method. For the purposes of this Guideline, it may be assumed to have the same measurement rules and to give similar sizes to those of the IFPUG method. From here onwards, differences of the Nesma and the IFPUG methods will be mentioned only when significant for conversion.

When we refer to the 'three methods', we therefore mean the IFPUG/Nesma, MkII and COSMIC methods and we always mean 'Unadjusted Function Points'. The IFPUG and MkII 'Value Adjustment Factors' are not recognized by the Nesma or COSMIC methods. All statistically based results reproduced here use 'unadjusted' function point sizes, abbreviated to 'FP' sizes.

Table 1 presents the common concepts used by the IFPUG/Nesma, MkII and COSMIC methods needed to describe the context of a software item to be measured and the type of measurement that is needed.

Table 1 - Concepts of the three FSM methods for defining the context of a software item to be measured

IFPUG/Nesma	MKII	COSMIC	Notes
Application	Application	Software item to be measured, e.g. an application	The concept of Application is the same in all methods.
Purpose of the count	Purpose of the count	Purpose of the measurement	Refers to main purpose of a measurement. This concept is used in a similar way by all methods.
Scope of the count	-	Scope of the measurement	Equivalent in both IFPUG/Nesma and COSMIC methods.
Application Boundary	Boundary	Boundary	Conceptually compatible in all methods.
User	User	Functional User	Conceptually compatible in all methods. The IFPUG/Nesma and MkII methods focus on human users and other interfacing software. COSMIC considers humans, other software, and hardware devices that interact with the software being measured as 'functional users'.
Files (Internal Logical and External Interface)	(Data) Entities	Data Groups stored in 'persistent storage'	Stored data is modelled: by IFPUG/Nesma as 'File-types' that are referenced by 'Elementary Processes'; by MkII as 'Entity-types' that are referenced by 'Logical Transactions'; by COSMIC as 'Data Groups' that are held in 'persistent storage' which is available to all 'functional processes'. These various concepts rarely correspond exactly to each other.

Figure 1 shows the relationships between the application (or software item) being measured and its context of the other concepts described in Table 1, for the three FSM methods.

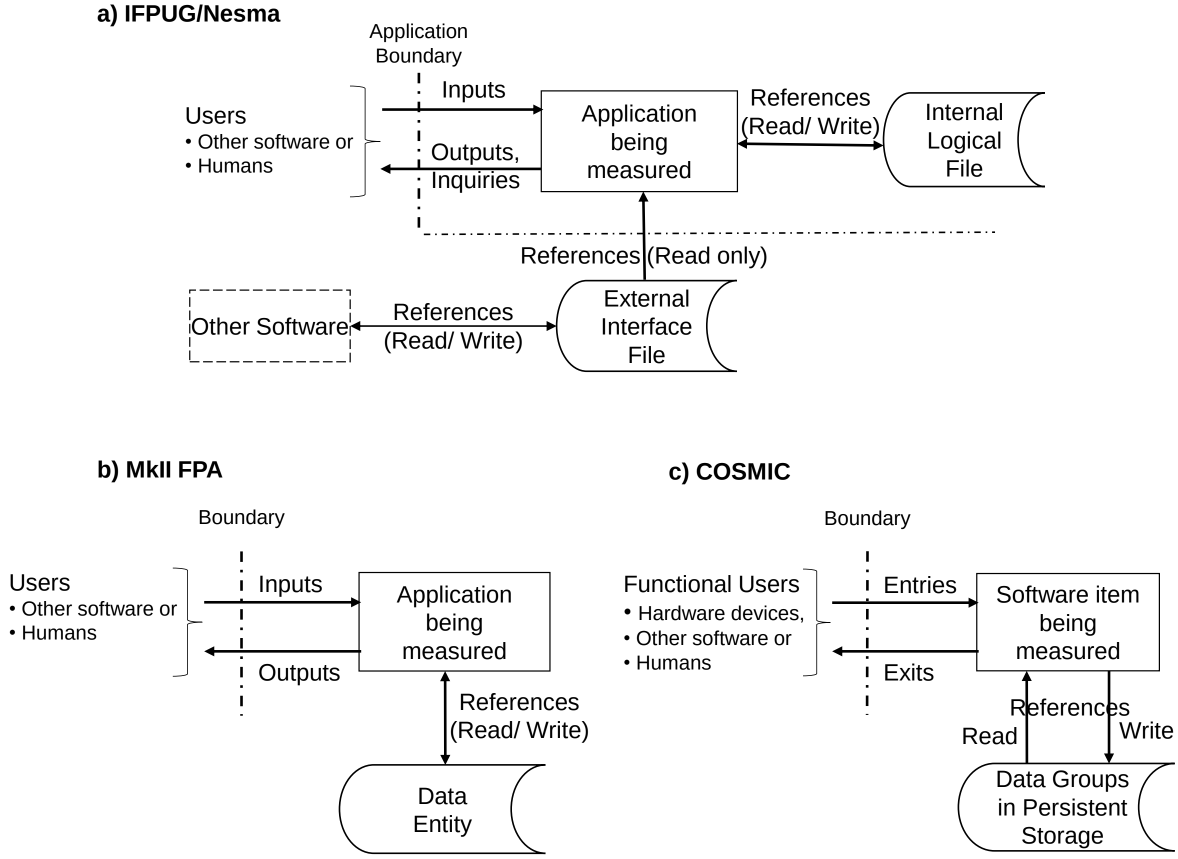


Figure 1 - Representations of the basic concepts used by the three methods to define the context of the software being measured

All FSM methods measure a 'functional size' of the Functional User Requirements (or 'FUR') of the application (or software item) that has been defined by the purpose and scope of the measurement.

The ISO/IEC 14143/1[2] [5] standard that established the principles of FSM, defines FUR as follows:

FUR: "a sub-set of the user requirements. The FUR represent the user practices and procedures that the software must perform to fulfil the users' needs".

The term 'FUR' is used by all FSM methods with the same meaning. All FSM methods require that the Measurer derive the FUR of the software to be measured from available artefacts. These are usually requirements specifications in various formats, designs, or the installed software.

After identifying the FUR, the Measurer maps the FUR to the 'Base Functional Component' types (BFC types), as defined by each FSM method in order to identify the 'things' that will be measured.

ISO/IEC 14143/1 defines BFC's and BFC types as follows:

- *BFC: "an elementary unit of FUR defined by and used by an FSM Method for measurement purposes". (Examples: "Add a new customer", "Report Customer Purchases")*
- *BFC type: "A defined category of BFCs". (Examples: "External Input", "Logical File", "Logical Transaction")*

[2] An International Standards Organization / International Electrotechnical Commission (ISO/IEC) Working Group established the ISO/IEC 14143 family of international standard and technical documents for functional size measurement (FSM) methods.

The BFC types are the main inputs to the measurement function of each FSM method (see Table 2). The main differences between the FSM methods are due to the different rules and principles for identifying and measuring the method's BFC types [14], [19]. It is these differences in BFC types and the rules for measuring them that result in different measured functional sizes.

Table 2 - The BFC Types defined by the three methods (See also Table 3).

IFPUG /Nesma	MKII	COSMIC	Notes
Elementary Processes (BFC types)	Logical Transactions (BFC types)	Functional Processes	These are the 'transaction types' as defined by each method. They roughly correspond to each other, though definitions and detailed measurement rules differ.
File Types (BFC types)	--	--	The IFPUG/Nesma methods define two File Types
--	--	Data Movements (BFC types)	These are the components of a functional process that move 'data groups' (comprising 'data attributes' each describing a single 'object of interest') between the software being measured and its functional users, and between the software and persistent storage.

Figure 2 shows how the BFC types relate to FUR, to each other, and to their components. This Figure uses a common representation of transaction types as each having an 'input', 'process' and 'output' component, as an aid to understanding the commonality of the three models. This is a convenient representation rather than an accurate description of the three models. (For more on this, see section 3.3, Assumption 3.)

The arrows loosely indicate 'consists of'.

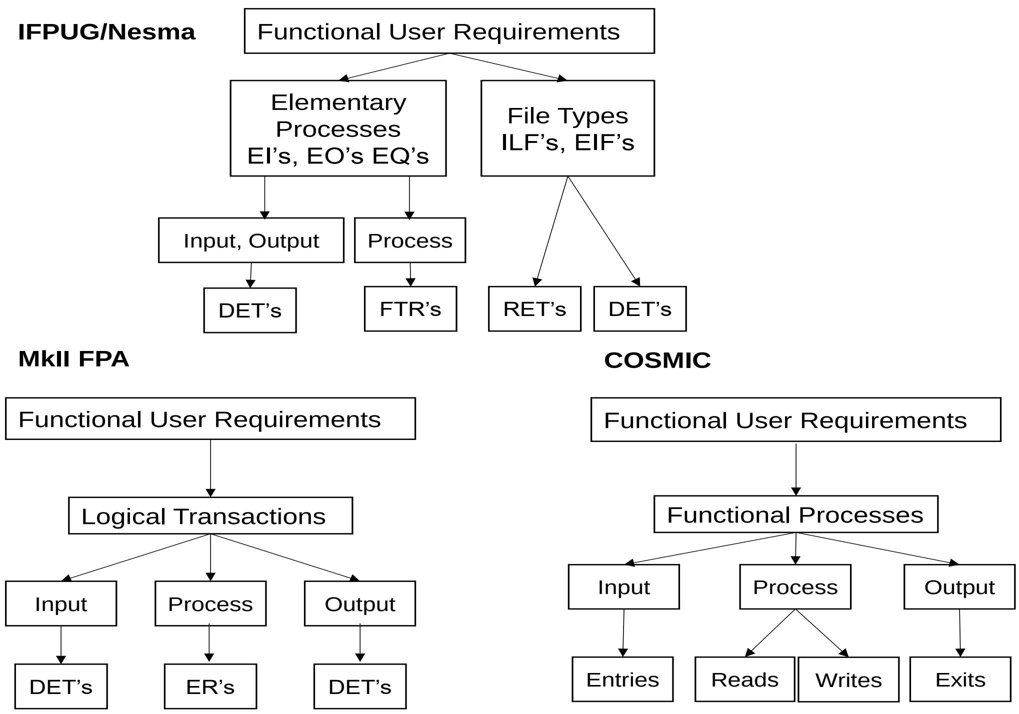


Figure 2 - Mapping of FUR to BFC types and their components by the three methods.

For the abbreviations used in **Figure 2** and in Table 3, see Appendix D.

After identifying the BFC types, the Measurer counts them and 'applies the measurement function', i.e. multiplies by their relative contribution to size (which in the case of the IFPUG/Nesma and MkII methods depends on counts of their components), and adds up the sizes. Table 3 shows a general comparison of BFC types, their components (IFPUG/Nesma), and measurement functions (= contribution to size) of the three FSM methods.

Table 3 - Comparison of the IFPUG/Nesma, MkII and COSMIC FSM methods [18]

FSM Method	BFC Types		Descriptions of BFC types (IFPUG/Nesma and COSMIC), or of their components (MkII)	Measured BFC components, or BFC types (COSMIC)	Functional Complexity	Contribution to Size
IFPUG/ Nesma	Elementary Processes	External Input	Input/Output Message crossing the boundary; Input Message to persistent storage	Input/Output DET's & FTR's	Low	3
					Average	4
					High	6
		External Output	Input/Output Message crossing the boundary; Output Message from persistent storage with derived data	Input/Output DET's & FTR's	Low	4
					Average	5
					High	7
		External Query	Input/Output Message crossing the boundary; Output Message from persistent storage with no derived data	Input/Output DET's & FTR's	Low	3
					Average	4
					High	6
	File Types	Internal Logical File	Files maintained by the application	DET's & RET's	Low	7
					Average	10
					High	15
		External Interface File	Files maintained by another application and referenced by the application being measured	DET's & RET's	Low	5
					Average	7
					High	10
MkII	Logical Transaction		Input Message crossing the boundary	Input DET's	-	0.58
			Output Message crossing the boundary	Output DET's	-	0.26
			Processing Part: within the boundary	ER's	-	1.66
COSMIC	Data Movement Type within a 'Functional Process'	Entry	Input of a data group from a functional user across the boundary to the software,	Entries	-	1
		Exit	Output of a data group from the software across the boundary to a functional user	Exits	-	1
		Read	Movement of a data group from persistent storage within the boundary, into the software	Reads	-	1
		Write	Movement of a data group from the software to persistent storage, within the boundary	Writes	-	1

1.2 Implications of differences of FSM Methods on defining conversion formulas

This section summarizes the implications of the differences in the measurement rules on the convertibility of sizes measured by first generation methods to COSMIC sizes.

1.2.1 Non-matching BFC types

As shown in section 1.1 the BFC types of first generation FSM methods do not map well to the BFC types of the COSMIC method. In particular the contribution to size of IFPUG/Nesma 'files' versus MkII and COSMIC references to persistent data is quite different – see Table 4.

Table 4 – Contribution of stored data to functional size in the considered methods

Method	Direct contribution to size	Indirect contribution to size
IFPUG/ Nesma	Yes (ILF's and EIF's contribute to size, dependent on the count of their RET's)	Yes (FTR's contribute to the size of elementary processes)
Mk II	No	Yes (Entity References contribute to the size of logical transactions)
COSMIC	No	Yes (Read and Write data movements contribute to the size of functional processes)

There are other significant differences in the processing of stored data. A single IFPUG/Nesma file type may have several record types (RET's). The count of these RET's determines the file type's size. However, the size of an elementary process is affected only by the count of File Types Referenced (FTR's) during its processing, ignoring how many RET's the file may have.

Further, it is most likely that an IFPUG/Nesma RET (not a file type) corresponds to a MkII 'entity type' or to a COSMIC 'object of interest type'. This matters when sizing 'transaction types'.

- A MkII logical transaction will measure a size contribution for each entity type referenced (ER's) during its processing (equivalent to counting RET's processed).
- A COSMIC functional process will usually measure one Read or one Write of a data group describing each object of interest needed for its processing; this is also usually equivalent to counting the RET's referenced in a functional process. But a RET and a data group type are not always exactly equivalent and there are cases where a COSMIC functional process may be required to have more than one reference to a data group describing the same object of interest. For example, an update functional process may count both a Read and then a Write of data describing the same object of interest, e.g. when processing is in batch mode. See also the 'Data Movement Uniqueness' rules in section 3.5.7 of the Measurement Manual [1] for other exceptions.

A MkII or COSMIC 'transaction type' may therefore gain a much larger contribution to its size as a result of counting their equivalent of references to RET's than an IFPUG/Nesma elementary process gains from its count of FTR's. This difference is offset to a degree by the fact that the MkII and COSMIC methods do not count a direct contribution of file types to functional size as shown in Table 4.

The consequences of these differences compared to an average IFPUG/COSMIC size ratio are:

- the IFPUG/Nesma size of an application that refers to many 'File types', especially 'complex' files, but has few references-per-file from its elementary processes will probably be larger relative to the corresponding COSMIC size.
- on the other hand, the COSMIC size of an application that has many data retrievals and/or stores per transaction will probably be larger relative to the corresponding IFPUG/Nesma size. Examples of the latter include software, which maintains life assurance policies, and billing systems with complex pricing rules, both of which manage highly inter-connected stored data structures.

These two effects are illustrated in Figure 3.

Similar relative size differences exist between MkII and IFPUG/Nesma sizes, for the same reasons.

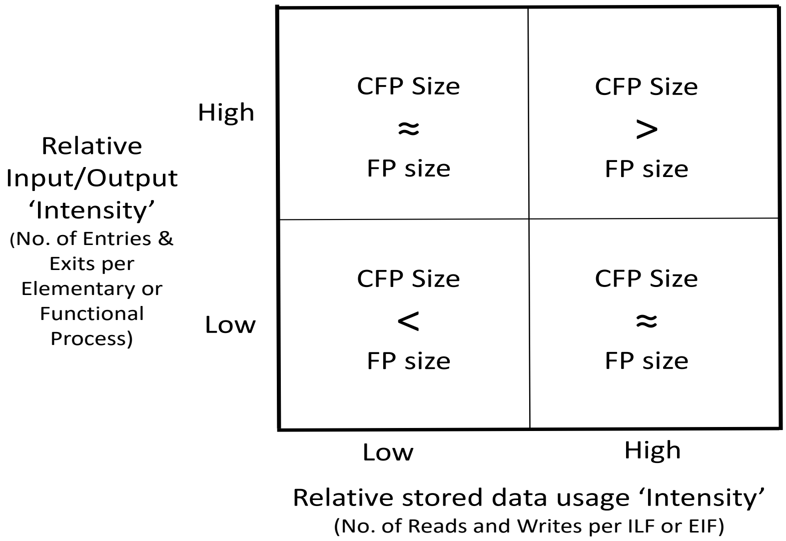


Figure 3 - The effects of relative data usage intensity and relative input/output data intensity on COSMIC versus IFPUG/Nesma sizes [5]

Some recent approaches to convert IFPUG/Nesma sizes to COSMIC sizes tried to take these fundamental differences into account and provided statistical conversion formulae directly from the size of IFPUG/Nesma 'Transactions' (i.e. by excluding the 'Data' size) (see section 2.2, Step 4d)). Even though this approach overcomes this problem to some extent, there are several other differences in the detailed measurement rules that limit convertibility.

For example, whilst the definitions of an IFPUG/Nesma 'elementary process', MkII 'logical transaction' and a COSMIC 'functional process' appear to have the same intent, the detailed counting rules of the FSM methods differ in what they consider to be separate processes for various types of functionality. A principal example is that functions to maintain 'code tables'[3] are not considered as elementary processes, i.e. they are ignored altogether by the IFPUG method, whereas the COSMIC method may consider such functions as functional processes for functional users such as system administrators who maintain the code tables. The MkII method has a similar but not identical convention to the COSMIC method. The Nesma method is similar to, but not identical to the IFPUG method.

Furthermore, the contributions to size of the input and output parts of IFPUG/Nesma elementary processes (and of MkII 'logical transactions') take into account the number of data attributes (= DET's) in the input and output. The COSMIC method considers only the numbers of 'objects of interest' about which data is moved in the input and output parts of a functional process (and not the number of data attributes moved) to derive the count of Entries and Exits. Also, as described above, it is more likely that the IFPUG/Nesma concept of a 'record type' maps more directly to a COSMIC 'object of interest' than does a logical file as the latter is typically at a higher level of abstraction than an object of interest.

Non-matching of BFC types is the first main reason why it is not possible to make 'exact' automatic conversion by a mathematical formula.

1.2.2 Measurement scale types: size bounds

The second major source of difference between the FSM methods is the way they define lower and upper limits, or 'bounds' of size that may be assigned to their BFC types. All FSM methods have a lower limit to the size of a transaction. The minimum size of a transaction is 3 FP, 2.5 MkII FP and 2 CFP when measured by the IFPUG/Nesma, MkII and COSMIC methods, respectively. As for the upper size, the MkII and COSMIC methods do not define any limit whereas IFPUG/Nesma has an upper limit to the sizes of all its BFC types.

Hence an application whose COSMIC functional processes have large numbers of data movements (say, well above the maximum size of an IFPUG/Nesma elementary process, namely 7 FP) will

[3] 'Code tables' mostly have two attributes, a code and a description that identify some 'thing', e.g. 'customer type', 'gender' (e.g. of an employee), 'currency', etc.

probably have a size measured in CFP that is much larger than the size measured in IFPUG/Nesma FP. Evidence for this includes the following:

- An analysis of COSMIC-measured banking applications found sizes of functional processes ranging from 2 up to 70 CFP, with an average size of about 8 CFP [private communication].
- Another study reported functional process sizes in the 3 – 30 CFP range [23].
- An analysis of business applications in a large software company [24] showed ratios of CFP/FP sizes for their Use Cases ranging from 0.8 to 7.3, with an average of about 2.0.

Similarly, an application that has a relatively high number of input and/or output data attributes per functional process will probably have a size measured by the MkII method that is larger relative to the sizes measured by the IFPUG/Nesma and COSMIC methods.

Therefore, when deriving a conversion formula, it is suggested to derive a different formula for each subset of data considering different types of applications. For examples of how this may be done, see section 2.2 and section 3.3.1, which describes a method of checking the homogeneity of a dataset by 'functional profiling'.

Information that is not captured in the IFPUG/Nesma size measure of a transaction (e.g. functionality exceeding 7 FP for an EO) can never be 'converted' to an equivalent COSMIC size. This point is becoming more important as transaction sizes increase in line with increasing business complexity.

1.2.3 Scope of measurement: 'layer' and 'level of decomposition'

First generation FSM methods were designed to measure the functionality of 'whole' business applications as seen by human users and/or other interfacing applications. However, the scope of a COSMIC measurement may include software in other 'infrastructure' layers (e.g. a layer for access control or for message handling) as well as components of applications. Size measurements of such infrastructure or component software obtained by using first generation FSM methods may not be comparable at all with the equivalent COSMIC measurements.

EXAMPLE. Suppose we have a software development project requiring a new business application and new functionality in multiple layers of the infrastructure software. The IFPUG/Nesma or MkII FP methods may give a size for the business application component that can be converted sufficiently accurately to a COSMIC size using a total size conversion formula. However, as none of the first generation methods were designed to measure software from other layers, it is unlikely that these methods could be used to measure any sizes of the new software in the infrastructure layers that would sensibly correlate with sizes measured by the COSMIC method.

1.2.4 Project development type (new development / enhancement)

The IFPUG rules for measuring the size of required enhancements to existing software (comprising additions of new functionality and changes to and deletions of existing functionality) differ fundamentally from the corresponding MkII and COSMIC rules.

- The COSMIC method (and the MkII method) measures the size of the required changes to the BFC's of the software.
- The IFPUG method measures the total size of the BFC's that must be changed, regardless of how much they must be changed.

As a consequence, if the required enhancements consist only of additions to existing functionality, then enhancement sizes measured by the IFPUG and COSMIC methods may correlate as well (or not) as sizes of new developments. However, sizes measured by the IFPUG and COSMIC methods are less likely to correlate well for enhancements that consist of additions and changes to, and deletions from, existing functionality.

The standard Nesma method measures changes to software exactly as per the IFPUG method. Nesma also has an experimental variant that applies an 'impact factor' to account for the proportion of a changed BFC that must be changed, resulting in a size in 'EFP'. Sizes measured by this variant may be more closely related to COSMIC measurements of the size of changes. Since there are no data about this relation, and because the method is experimental, we will not consider this variant in this Guideline.

It may be that first-generation method sizes versus COSMIC sizes of general enhancements (additions, changes and deletions) correlate sufficiently well that a statistical conversion formula may give sufficiently accurate converted sizes for practical purposes. However, there is currently a general lack of data on the convertibility of sizes of enhancements measured between all FSM methods.

2

CONVERSION OF TOTAL FIRST GENERATION FSM METHOD SIZES TO COSMIC SIZES

2.1 Conversion objectives

Before starting a conversion exercise, the objectives must be clearly defined as these will affect the required accuracy of the converted sizes and hence the choice of the conversion method. Here are some example objectives; they particularly apply to IFPUG/Nesma-to-COSMIC size conversion:

EXAMPLE 1: To determine a first generation-to-COSMIC total size conversion formula for a large population of software items so that the formula may be used to re-calibrate an existing project effort estimation method or tool. The accuracy of the individual converted COSMIC sizes may not be critical for this objective. The method described in section 2.2 should produce a conversion formula that is sufficiently accurate in relation to the objective. The effort for a new project measured using COSMIC should be estimated with the re-calibrated estimation method and tested, where practicable, against the effort obtained by the existing estimation method. (Note: see Appendix C for some research on a new way of re-calibrating an existing effort estimation method that may avoid the need to develop a conversion formula.)

EXAMPLE 2: To determine the COSMIC sizes of a portfolio of software items; the individual sizes will be used to calibrate a system for paying a supplier for the maintenance and support of the individual software items. The method described in section 2.2 below may produce sizes that are sufficiently accurate on average for this purpose, but the accuracy of converted individual sizes must be checked carefully. Note especially Steps 1 to Step 4a) on refining the converted sizes and Steps 4b) to 4e) on identifying and removing 'outliers' from the sample used to establish the conversion formula. For small software items where conversion accuracy is often poor, it may be more economic to measure the COSMIC sizes of these items directly rather than to use a conversion formula.

EXAMPLE 3: To determine COSMIC sizes of a population of software items accurately, i.e. to within a few percent. The results of studies on the use of total size conversion methods given in Chapter 2 suggest that these methods cannot reliably produce the required accuracy of COSMIC sizes. The alternative is to use a 'direct' size conversion method as described in section 3.2. Effectively this means measuring the COSMIC size of each software item, but assisted by knowledge of components of the first generation size measurement and of the FUR of each item. Alternatively a 'component' size conversion method as described in section 3.3 may be tried.

EXAMPLE 4: To determine the COSMIC sizes of enhancements (that require adds, changes and deletes) to existing business application software items, or of components of such items, or for software items for which first generation FSM methods were not designed to measure a functional size. It may be that the approaches described in Chapter 3 (and possibly Chapter 2) will produce converted sizes of these types of software with sufficient accuracy. Unfortunately, we have no evidence from studies on the convertibility of such sizes, in particular for the sizes of enhancements. And from our knowledge of the very different rules of the IFPUG and COSMIC methods for sizing enhancements, good size correlations are not expected. Therefore it may be necessary to use the same approaches as advised for Example 3 for converting the sizes of these various types of software.

2.2 The total size conversion method

If an organization has a historical database of size measurements by a first generation FSM method of whole business applications and of whole additions of functionality (together referred to as 'software items') and wishes to adopt the COSMIC method, then the process described below and summarized in Figure 4 may be used to establish a local formula to convert the existing measurements. The process is based on our analysis of the studies in the literature (see Appendices A and B for details).

Provided the process is followed carefully, paying special attention to the quality controls of Steps 1 – 4, the process should yield sufficiently accurate results for the converted sizes to meet the Example 1 and 2 objectives described in section 2.1.

Details of the process follow after Figure 4.

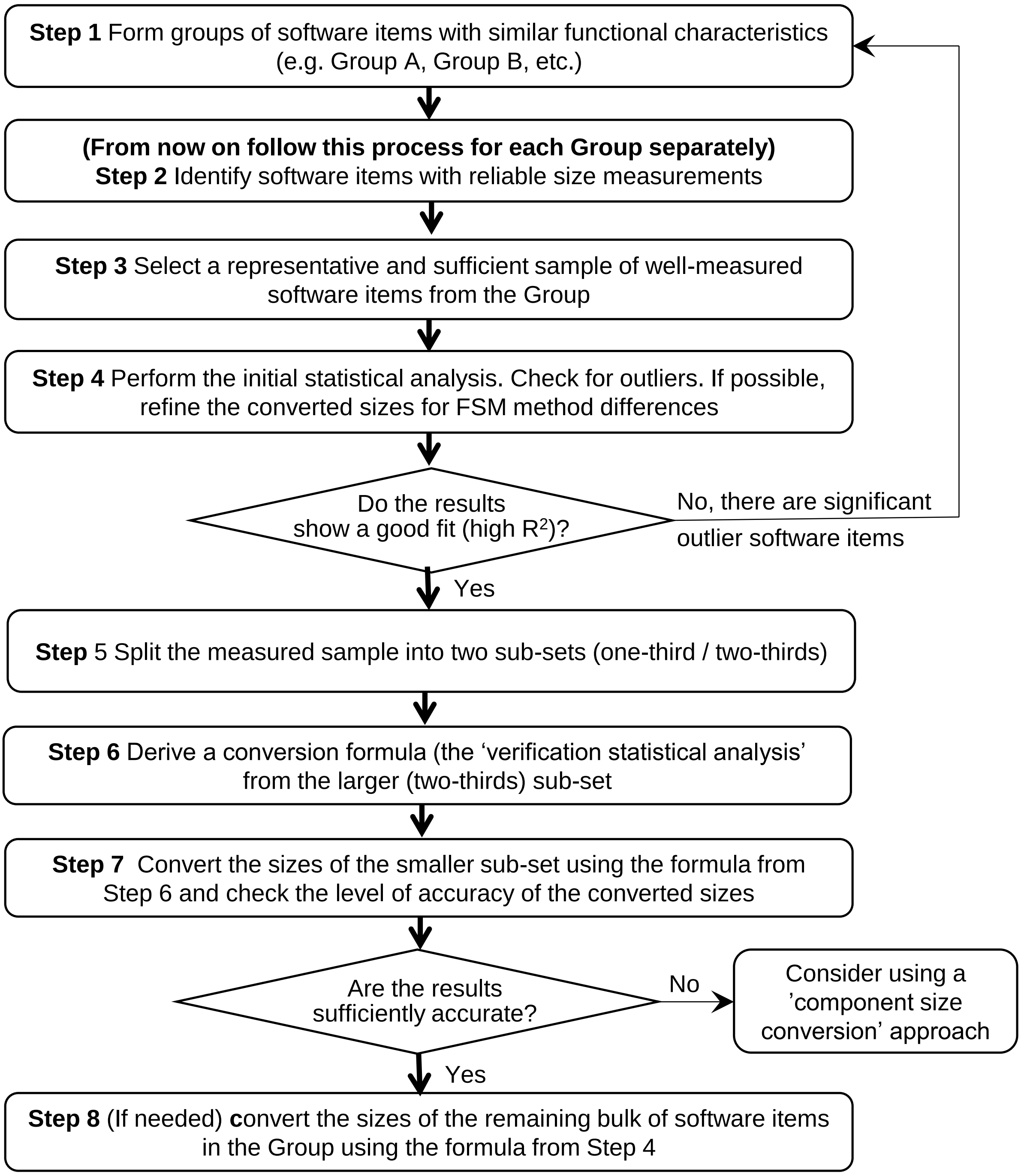


Figure 4 – Process steps of the total size conversion method

Step 1: Form groups of software items with similar characteristics (e.g. Group A, Group B, etc.)

From the whole population of software items whose existing first generation functional sizes must be converted to COSMIC sizes, sort them into separate, functionally homogeneous Groups. As evidenced by Figure 3, local characteristics of the software can influence the nature of the relationship between first generation and COSMIC sizes.

For example, consider forming separate groups for the software items from different functional domains, such as different divisions of a company (e.g. engineering versus sales) or operational versus management information systems (e.g. transaction-processing software versus data warehouse software).

The ISO 14143-5:2004 Technical Report [6] describes two methods of distinguishing functional domains for the purposes of functional size measurement. Another test for functional homogeneity of software items as measured by the IFPUG/Nesma method is described in section 3.3.1.

Decisions on grouping software items may need to be made iteratively. Step 4 of this process may reveal that the first generation and COSMIC sizes correlate poorly. This may be due to the software items not being a homogeneous group or that there are outliers that must be removed. The causes can only be understood by examining the FUR of the software being converted for the characteristics that cause the poor correlation. Examining the factors described in Step 4 may reveal some of these causes, enabling sizes to be corrected to improve the correlation. The groups may even need to be re-formed.

Another reason for forming groups from the whole population of software items whose sizes must be converted will be found in Appendix A.1 section A.1.1, which discusses results of conversion studies. These suggest, for example, that grouping IFPUG/Nesma-measured software items into size bands (e.g. <400 FP; and >400) before fitting a straight line to pairs of sizes in each band separately may give better correlations with COSMIC sizes than fitting one straight line to the whole dataset.

We advise this because from our knowledge of how the IFPUG/Nesma and COSMIC methods measure functional size, if we had a very large set of measurements (say a few hundred), we would expect the best-fit relationship between CFP and FP sizes to be a shallow curve, with CFP sizes increasing faster than FP sizes. However, a real-world conversion study typically involves measuring at most 20 –30 software items, with a tendency to have more measurements of smaller software items than larger. For such a dataset it is normally risky to attempt to allow for the continuous curve by fitting a second-order curve of the form $y = ax^2 + bx + c$, or to fit a straight line to a log-log transformed dataset, because the few large data points will dominate the Ordinary Least Squares (OLS) fitted regression curve. Consequently, it may be safer to group the data into size bands and to derive OLS-fitted straight-line conversion formulae separately for each band.

Step 2: Within each group, identify software items with reliable size data

From now on, the conversion method Steps will be described as they apply for just one group formed in Step 1. The same process should be applied for each group separately, except that some re-grouping may be necessary as a result of Step 4.

Within the Group, identify the software items whose existing size measurements are reliable. For example, sometimes sizes are ´estimated´ rather than measured due to not having sufficient information for accurate sizing early in the life cycle, time constraints etc. As estimated sizes always involve some amount of error, these sizes should not be included in the sample to be chosen in the next Step 3 for re-measurement using the COSMIC method. (The sizes of all members of each group, whether estimated or measured accurately may, of course, be converted when the conversion formula is eventually decided in Step 8 of the process.)

Step 3: Measure the COSMIC size of a sample of software items from a group

Select a sample of software items whose sizes will be re-measured using the COSMIC method.

The sample software items should be representative of the group both in terms of functionality and range of sizes of the group. Subject to these constraints, the selection should be random.

The number of sample items to be re-measured should be at least 15 and ideally more, e.g. 20 – 30. (Obviously, the number to be re-measured involves an economic trade-off decision between re-measurement effort and desired conversion accuracy.

Measure the COSMIC sizes of the software items in the sample, preferably using experienced, COSMIC-certified Measurers.

Step 4: Perform the ‘initial’ statistical analysis

a) Plot the results of the first generation and COSMIC size measurements on a scatter diagram. First fit a straight line of the form $y = ax + b$, where y is the COSMIC size and x is the 1G size, by the

OLS method. Then, determine the R-squared (or 'coefficient of determination') value. R-squared is basically a measure showing what percentage of the total variation in y is described by the variation in x. Therefore, a high R-squared for a fitted straight-line indicates that it can describe well the relationship between two variables. Results of actual OLS fits to IFPUG/Nesma and COSMIC measurements in Appendix B have R-squared values mostly greater than 0.8, which implies a reasonably good correlation. This straight line will be the 'conversion formula' that will be used to convert the first generation sizes of the main population of measurements.

As well as the R-squared value, also make sure that the 'p-value' of the dataset is not greater than 0.05 (the value generally used for this type of analysis). If the p-value is greater than 0.05, the formula for the fitted straight line is not reliable, and you should try with a different (larger) dataset. Fitting the straight line (the 'Regression' tool in Excel), as well as the R-squared and p-value can be obtained easily using standard spreadsheet functions. An example of a regression line is given in Figure 5 below.

(Some studies have suggested that a better conversion formula can be obtained by plotting the logarithm of the IFPUG/Nesma size versus the logarithm of the COSMIC size, a so-called 'log-log' fit, and then fitting a straight line. It may be the case that the R-squared value obtained this way is higher than from a linear-linear fit, but beware; it does not follow that converted COSMIC sizes will be more accurate. Be very careful about using a conversion formula obtained from a 'log-log fit'– see Step 7 below.)

b) Study very carefully the distributions of the data on each axis as well as the scatter of observations (pairs of data points) and the goodness of fit (of a straight line or otherwise) to the two sets of measurements, to see whether there are some potential outlier observations. Outliers are data points that are noticeably different in some way from the trend line that best fits the bulk of the data points. If they are not excluded, the trend line may not be the best representation of the relationship between the bulk of the data pairs (see section 2.3 for how to detect outlier data pairs).

   Pay careful attention to understanding why some observations appear as 'outliers' and consider whether there is a concrete reason (e.g. the software belongs to another application type). It is vital to understand if and why outliers should be discarded from a dataset. Do not eliminate outliers only on the basis of statistical tests; you may be throwing away valuable information. There may be data points that look perfectly fine on any single axis but are multivariate outliers. For example, for the general population there is nothing unusual about a 6-foot (1.83m) man or a 120-pound (54.5 Kg) man, but a 6-foot man that weighs 120 pounds is unusual and might be regarded as an outlier. On the other hand, if some data points on one axis are located far from the others (e.g. a 7-foot man), these data points may belong to another group. Outliers may be due to a systematic difference in some way from the bulk of the data points or may simply occur due to a mistake in measurement. If there appear to be systematic differences, then better fits may be obtained by dividing the group into sub-sets of different size ranges and observing goodness of fit in different subsets, as suggested in Step 1.

c) Another useful test of whether a fitted straight line will be suitable for use as a conversion formula is to examine the value of the constant 'b' obtained from the straight line fit. The results in Appendix B mostly show a negative value of 'b', which is to be expected. When the value of the COSMIC size 'y' is zero, the corresponding IFPUG/Nesma size 'x' according to the fitted straight line is '-b/a' which should be a small positive number. This result is to be expected because the IFPUG/Nesma methods measure a contribution to total size of the file-types, separately from the 'transaction functions'. If a straight-line fit for projects of size less than, say, 400 FP yields a value of '-b/a' that is negative or large (greater than, say, 30 - 50 FP), proceed to Step 8a). (Such values are theoretically impossible if negative, and unlikely if very large.)

   Note: this test may also be applied to a set of software items whose sizes exceed ≈400 FP. However, results discussed in Appendix A, section A.1.1 suggest that above ≈400 FP, COSMIC sizes increase faster than FP sizes. If this observation is generally valid, a straight line fitted to these larger sizes will tend to have an increasing value of the intercept '-b/a' for zero COSMIC size. The outcome of this test will therefore be more unpredictable and less useful as a test of validity of the conversion formula.

d) Consider whether, for each group and its sample of software items, you can exploit knowledge of their FUR to identify, or better to correct for, some of the differences between the first generation and COSMIC sizes arising from the factors described in section 1.2. For example:

i. As noted in Step 4 c), IFPUG/Nesma FP sizes have contributions from ‘transactions’ and from ‘file types’; COSMIC CFP sizes are based only on transactions. Do CFP/FP sizes correlate better if only the FP transaction sizes are used? (**Theory and the results presented in Appendix A, section A.1.2, suggest this is well worth exploring.)**
ii. FP sizes do not measure functionality such as the transactions needed to maintain code tables; CFP sizes do measure this functionality (see section 2.1). Do CFP/FP sizes correlate better if the contribution to the CFP sizes of code table maintenance is dealt with separately, e.g. by a ‘direct’ conversion method (see Chapter 3)?
iii. FP sizes are limited by the upper bounds of transaction ‘complexity’; CFP sizes have no such limit (see section 2.2). Is there a way whereby you can recognise which FP sizes will be particularly inaccurately converted due to this difference, and can you either group these software items separately or apply a correction to their converted sizes?

For each such factor that is found to contribute to a poor correlation you will need to decide how to proceed not just to correct the sample for each group but for the converted sizes of the main populations of size measurements.

e) If the decision is made in step 4-d-i above to investigate the relationship between IFPUG/Nesma transaction sizes and COSMIC sizes, explore fitting a straight line that is constrained to pass through the origin at (0,0). This might produce a better-fitted conversion line because the value of ‘b’ must effectively be zero (given that the smallest size of a transaction is 3 FP or 2 CFP according to the IFPUG/Nesma and COSMIC methods respectively.)

f) At the end of this step, we assume you have a straight line fitted to the first generation and COSMIC sizes that satisfies the statistical tests, but that there may be some data pairs rejected as outliers. These outlier data points should be ‘recycled’ to Step 1 for possible inclusion in another group. Alternatively, the re-measured CFP size can simply be accepted.

Step 5: Split the measured sample into two subsets (One-third / Two-thirds)[4]

This step and the following two steps are commonly used in statistical analysis when deriving a regression formula. However, this step does require sufficient data points in the sample (e.g. > 15 software items). Select randomly two-thirds of the software items from the whole sample; these will form the large subset to be used to derive the conversion formula. The remaining one-third of the software items will form the smaller subset, which will be used to test the accuracy of the derived conversion formula.

Step 6: Derive a conversion formula from the large subset (the ‘verification ‘statistical analysis)

As in Step 4 a), plot the results of the first generation and the COSMIC measurements for the large subset on a scatter diagram. Fit a straight line of the form $y = ax + b$, where y is the COSMIC size and x is the first generation size, by the OLS method and determine the R-squared value and p-value for reasonableness tests.

Step 7: Convert the FP sizes of the smaller subset to COSMIC sizes and analyse the accuracy of the converted sizes against the conversion objectives

Use the conversion formula derived in the previous step to predict the COSMIC sizes for the software items in the smaller subset and compare these converted sizes against the measured sizes. This comparison will give an indication of whether the converted sizes of individual software items will be accurate enough in relation to the conversion exercise objectives. Conversion accuracy is likely to be low, in percentage terms, for small software items.

If the outcome of the analysis of the process shows that using a fitted conversion formula is unlikely to produce sufficiently accurate COSMIC sizes in relation to the conversion objectives, consider using a ‘direct’ or a ‘component size conversion’ method as described in Chapter 3. Otherwise, continue with the next step, if needed to meet the conversion objectives.

Step 8: Convert the sizes of the bulk of software items in the group (if needed)

Apply the fitted conversion formula derived from the whole sample, i.e. as at the end of Step 4, to convert the main population of the first generation measurements in the group to COSMIC sizes. Make sure that the software sizes to be converted fall in the same size range of the sample from which the

[4] Readers who have a strong statistical background are recommended to use a ‘leave-one-out’ analysis technique instead of the simpler process of Steps 5 - 7 described here.

conversion formula was derived. (Take for example the data plotted in Figure 5, which is based on whole applications that are mostly smaller than 400 FP: this fitted straight line should not be used to estimate the CFP of applications of size greater than 400 FP. There are not enough data to be sure that the fitted straight line is statistically reliable above 400 FP.)

2.3 Detection of outlier observations; an example analysis

In statistics, an outlier data point (or 'data pair' (x, y)) is an observation that appears to deviate markedly from other members of the set of observations in which it occurs [28].

Outliers may be a result of measurement errors or they may be members of a different population that does not really belong to the set being studied. Correct elimination of outliers is important as their inclusion may distort the line fitted to the majority of the measurements, thus producing inaccurate converted sizes for the majority of the measurements.

The term 'outlier' is commonly used in two senses that should be treated separately (please see Section 2.2 Step 4-b above). In this Guideline we refer to them as 'Type 1' and 'Type 2' outliers.

> Type 1: These are sometimes called ´leverage points´. They are observations that are distant from the bulk of other observations in the set. Example: a dataset of FP and CFP size measurements to be analysed may have a data point that is well outside the size range of the bulk of the measurements. Because of this distance outside the range of the bulk of the observations, a leverage point will have a very significant influence (or 'leverage') on the parameters of the line ('a' and 'b' in the case of a straight line) that is fitted to all the observations in the set. Leverage points must therefore be treated with caution. Specifically, their influence on the line fitted to the remaining (bulk) of the observations should be explored (see below for a worked example). Leverage points may not necessarily correspond to Type 2 outliers (see below). A leverage point is considered to be an outlier if it has an x value that is far from the bulk of x values and it is also relatively far from the fitted regression line.
>
> Type 2: These are the ´outliers´ that seem to be significantly dispersed from the line fitted to the great majority of the data points, even though they are within the size range of the majority.

Experienced data analysts will recognise Type 1 outliers and may well discard them automatically because a line fitted to all the observations cannot be trusted for safe predictions in the region of the outlier. 'Cook's distance test' [34] may be used to decide whether or not to treat a data point as a Type 1 outlier. In the example in this section, we use a simple process that exploits our knowledge of the nature of relationship between FP and CFP sizes (see Section 1) to decide whether to eliminate the Type 1 outliers.

There are established statistical tests for detecting Type 2 outliers in a given dataset. The most common methods are outlier-labelling methods. One such test is to compute the 'standardized (normalized) residuals', as follows[5].

a) Use the line fitted to the FP / CFP sizes as in section 2.2, Step 4a) or 6 to calculate the 'predicted' CFP sizes.
b) Calculate the ordinary or simple residuals ('e'), [i.e. 'e' = predicted CFP size – measured CFP size] for each data point.
c) Calculate the standardized residuals by dividing the ordinary residuals (e_i) by the estimate of their standard error (S) (i.e., the residual mean square error from the line fitted to the full dataset). The standard error (S) can be calculated using the formulae: $S = \sqrt{[\sum e_i^2/(n-2)]}$. Linear regression analysis also provides this value automatically). The ordinary residuals have a mean of 0 and a standard deviation 1.
d) Treat as Type 2 outliers the observations which fulfill two rules: i) they have e/S values exceeding +2 or -2 and ii) these e/S values are not consistent with their neighboring data points. A chart visualizing FP versus e_i/S can be very helpful in this exercise.

It is important to note that whichever method is used, the suspected outliers should not be directly eliminated without first trying to find the reason(s) why they are outliers. Perhaps the reason is simply a measurement error that may be corrected, or perhaps there is a feature of the functionality of the

[5] For a video showing step by step how to identify outliers in regression analysis using Excel, see (https://www.youtube.com/watch?v=IX-Na7fn6Zk)

software item that causes its measured size(s) (either first generation or COSMIC, or both) to differ markedly from the size(s) that would be expected from the OLS-fitted line (see Step 4 for other potential causes).

We now illustrate the process for dealing with outliers with some real data.

Figure 5 shows a real dataset of COSMIC and IFPUG size measurements of eleven banking applications [10]. It is an example of the output that might be obtained from Steps 4 a) described in section 2.2. However, the software items in the dataset were not selected specifically for a conversion study. At first sight, the dataset includes one very large Type 1 'leverage' data point (1424 FP, 1662 CFP), one smaller Type 1 'leverage' data point (766 FP, 810 CFP), and a Type 2 data point (260 FP, 81 CFP). These will need to be examined carefully to see if they really should be included in a sample set for a total size conversion study.

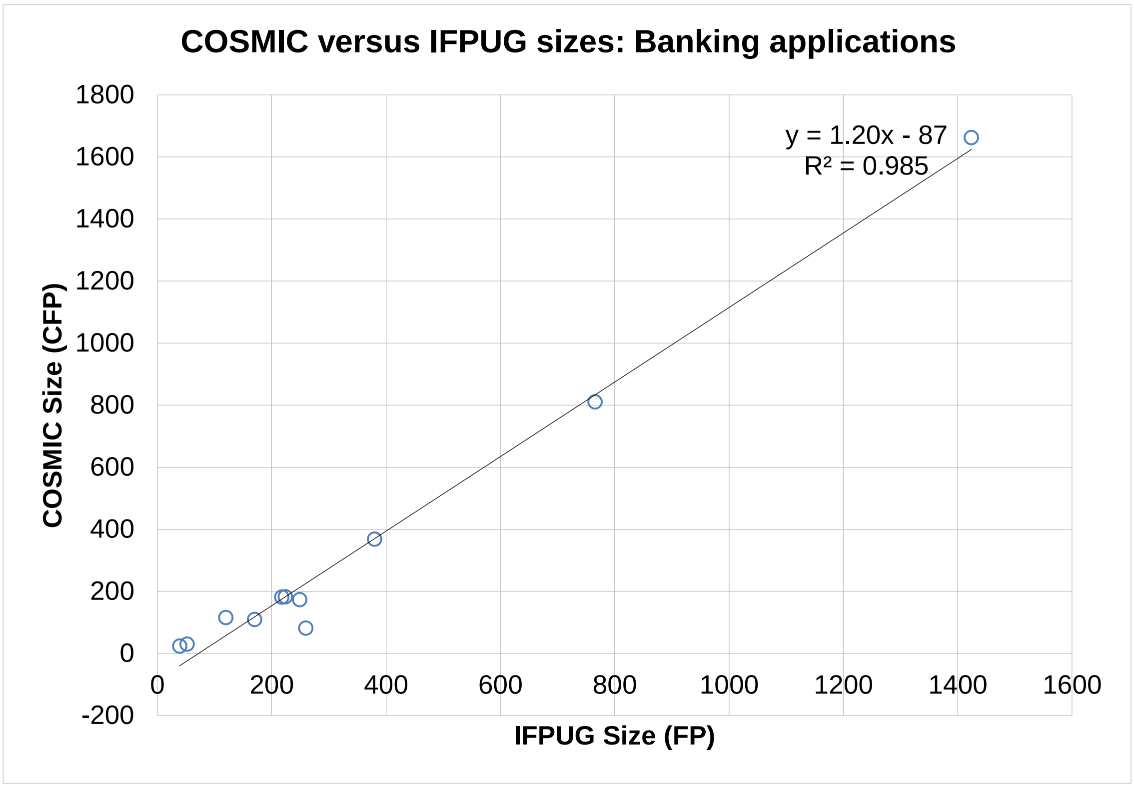


Figure 5 - Example COSMIC versus IFPUG measurements of a set of banking applications

Note the fitted straight line:

$$CFP = 1.20 \times FP - 87 \quad (1)$$

Table 5 shows the results of applying the standardized residuals process to detect Type 2 outliers described above, using formula (1).

Figure 6 shows the results from step d) of the process described in this section, plotting the standardized residuals versus FP size.

Table 5 – Results of applying the standardized residuals process to detect outliers (11 data points)

Measured		Predicted CFP	Error (residual) 'e' (Predicted – Measured CFP)	Standardized error (e/S)	% error (e/CFP)
IFPUG FP	CFP				
260	81	225	145	2.33	178,5%
218	181	175	-6	-0,10	-3,3%
224	182	182	0	0,00	0,2%
380	368	369	2	0,03	0,5%
39	23	-39	-63	-1,01	-273,7%
170	109	117	8	0,14	7,7%
120	115	57	-58	-0,93	-50,1%
249	173	212	40	0,63	22,7%
766	810	833	23	0,38	2,9%
53	29	-24	-53	-0,86	-1.8%
1424	1662	1624	-38	-0,61	-2.3%

* Standard error (S) is calculated as 62,07

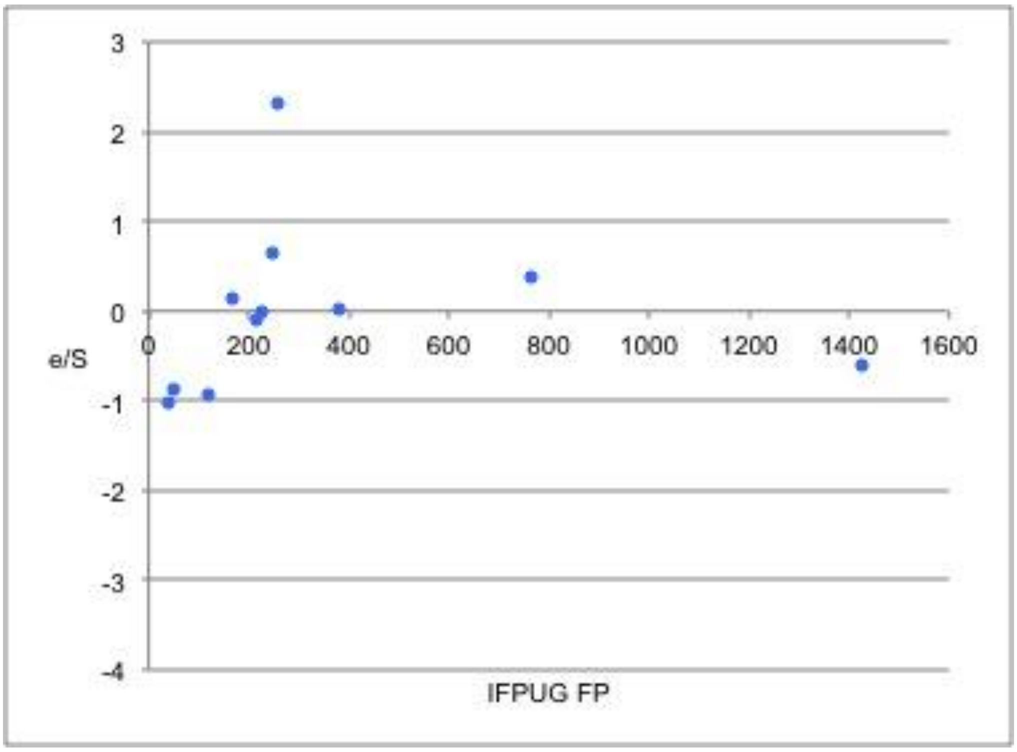


Figure 6 - Plot of standardized residuals vs IFPUG FP size

From Table 5 and Figure 6, it is clear that the data point at (260 FP, 81 CFP) is indeed a Type 2 outlier (e/S > 2 and this point is not consistent with the neighbouring points), whereas the Type 1 'leverage' data points are not outliers according to the statistical test for outliers. (Their x values are far from the bulk of x values, but these data points are relatively close to the fitted regression line.)

However, as noted before, one should not directly eliminate any outlier point before understanding the reason why the point is an outlier.

Regarding the Type 2 outlier, an investigation showed that almost 50% of the FP size of this software item was accounted for by ILF's and EIF's. Studies described in Appendix A, section A.1 have shown that the contribution of logical files to total FP sizes lies typically in the 30% - 40% range, so there is something untypical about this software item.

Further, this high contribution of files caused the complexity of the transactions to be classified as average (rather than low), or high (rather than average). This data point was therefore eliminated as an outlier. Both effects cause the FP size of this data point to be much higher than expected from the typical FP versus CFP relationship.

Eliminating the Type 2 outlier data point at (260 FP, 81 CFP) and re-fitting the straight line to the remaining 10 data points yields the following conversion formula:

CFP = 1.19 x FP – 69. (2)

The R-squared now improves to 0.995. Having dealt with the obvious Type 2 outlier, we can now repeat the process of calculating standardized residuals and then check if the converted sizes are sufficiently accurate for continuing with Step 5 to derive the final conversion formula. We therefore apply conversion formula (2) to the measure FP sizes and compare the predicted CFP sizes against the measured CFP sizes as in Table 6 below.

Table 6 - Comparison of actual and converted CFP (10 data points)

Measured		Predicted CFP	Error (residual) 'e' (Predicted – Measured CFP)	Standardized error (e/S)	% error (e/CFP)
IFPUG FP	CFP				
218	181	191	10	0,26	5,4%
224	182	198	16	0,42	8.8%
380	368	384	16	0,42	4.3%
39	23	-23	-46	-1,2	-197.9%
170	109	134	25	0,65	22.6%
120	115	74	-41	-1,08	-35.6%
249	173	228	55	1,44	31.7%
766	810	844	34	0,9	4.2%
52	29	-7	-36	-0,95	-124.2%
1424	1662	1628	-34	-0,88	-2.0%

* Standard error (S) is calculated as 38.

The average of the absolute values of the percentage errors for this dataset is 43.7%. However, there is one obviously inaccurately converted size at 39 FP for which the predicted CFP size is negative. The next-lowest FP size at 52 FP is also not well converted.

There are now two approaches we could take to deciding on the best formula to use for FP-to-CFP conversion purposes for the remaining 10 data points. First there is a pure statistical approach and second we can introduce our knowledge of the nature of FP and COSMIC sizes to improve on the pure statistical approach.

A pure statistical approach might start with the first observation that there are two data pairs in which the converted CFP sizes have the biggest percentage errors (-198% for the data point (39 FP, 23 CFP) and -124% for the data point (52 FP, 29 CFP)). Second, although the two large data points (1424 FP, 1662 CFP) and (766 FP, 810 CFP) are well outside the range of the other 8 data points, they fit very closely to the straight-line of formula (2), so cannot be automatically dismissed as Type 1 outliers according to the criteria given above.

At this point, if we introduce our knowledge of FP and COSMIC sizes, there are three elements to consider.

i. The 'b' parameter of the fitted straight line of formula (2) is a large negative value (-69). From Step 4 c) of the process described in section 2.2, we learned that -b/a (i.e. 69 / 1.19) = 58 is the intercept for the value of the FP size when the COSMIC size is zero. This IFPUG size of

58 FP, which reflects the size contribution of IFPUG 'file types' at zero contribution of the 'transaction functions', is implausibly large. Formula 2 is clearly a poor predictor of CFP sizes for low FP sizes.

ii. Without any statistical analysis at all, we can be sure from the basic structure of the two methods that a straight-line conversion formula will inevitably be a poor predictor of a CFP size from the measured IFPUG size for small software sizes, due to the stepped IFPUG measurement scale. For example, consider the smallest measured data point (39 FP, 23 CFP). Suppose for the sake of argument that this software item had to reference one more EIF than was actually measured.

The IFPUG size would increase by 5 – 10 FP depending on the EIF complexity, and the total size might further increase as a result of an Elementary Process that references the EIF having an increased complexity. So the measured size might increase from 39 FP, to somewhere in the range of 44 – 49 FP, or more, i.e. by 13 – 26%, or more. In contrast the CFP size, as a result of adding one Read would increase from 23 to 24 CFP, i.e. by 4%. Given this asymmetrical sensitivity, it makes no sense at all to apply a statistical test that assumes a normally distributed spread of data to eliminate a small data point as an outlier from this dataset. If this data point must be eliminated, there must be better reasons than the result of applying this statistical test.)

iii. Now consider the two large Type 1 data points. The largest is almost four times larger than, and the second-largest twice as large as, the maximum size (380 CFP, 368 CFP) of the cluster of the other 8 sizes. These two large data points will carry enormous weight in the OLS curve-fitting process.

The impact of these two large Type 1 data points on the curve-fitting process can be seen from Table 7, which shows the effect on the fitted straight line of including and then eliminating in turn the two Type 1 data points.

The results in Table 7 show very clearly that as the two large data points are removed in two successive steps:

- the slope of the fitted straight-line decreases significantly;
- the intercept (-b/a, i.e. the FP size when CFP = 0) decreases from an implausible 58 FP to a reasonable 26 FP;
- the average and maximum residuals decrease significantly.

Table 7 - The effect on the fitted straight-line of eliminating the large Type 1 data points

FP Size Range	No. of Data Points	Formula (CFP =	R^2	-b/a (FP)	Abs. Residuals %		
					Minimum	Average	Maximum
39 – 1424	10	1.19 x FP - 69	0.995	58	2%	43.7%	198%
39 – 766	9	1.09 x FP - 48	0.987	44	0%	32.6%	124%
39 – 380	8	0.95 x FP - 25	0.951	26	0%	23.8%	74%

The results in Table 7 are in line with the expectation outlined in Step 1 of the Conversion Method that 'the best-fit relationship between CFP and FP sizes (will) be a shallow curve, with CFP sizes increasing faster than FP sizes'. This expectation is supported by the analysis of the data in Appendix A.1, Figure 13 showing that the CFP/FP relationship becomes steeper somewhere around 400 FP – though the evidence is not very strong. Other analyses [30] have reached similar conclusions.

The results indicate that, notwithstanding the findings of the high R-squared from the simple OLS line-fitting, it would be unsafe to use formula (2) as a conversion formula for FP sizes in the range above ≈ 400 FP.

We may draw the following conclusions from this small example dataset and its analysis.

- Supposing our sample of 12 data points were taken from a large population of IFPUG size measurements that should be converted to COSMIC sizes, the best-fit straight line to be used is the formula in the bottom row of Table 7 for FP sizes in the range 39 – 380 FP.
- If there is a need to convert much larger sizes than ≈ 400 FP, there would be significant risk and unknown inaccuracy in using this formula [CFP = 0.96 x FP – 25], or any of the other formulas for this purpose. It is simply unsafe to reply on any conversion formula for this dataset, for sizes above ≈ 400 FP, where there are only two data points.
- Do not rely on statistical tests alone, e.g. R-squared or the use of standardized residuals, to eliminate outliers. Try to understand why a data point is a potential outlier before eliminating it, using the functional requirements and your knowledge of how the IFPUG and COSMIC methods measure software size.
- This small example has illustrated how one can find a test for functional homogeneity that can be applied not only to a sample of software sizes used to develop the conversion formula but also to the main population of 1G sizes to be converted. The test is to examine the ratio of the percentage contribution to total IFPUG size of the file types for each item in the main population. If this contribution exceeds, say 40%, then the chosen conversion formula is likely to be inaccurate. If there are many such software items, form their sizes into a separate group and develop a separate conversion formula for that group.

2.4 Conclusions and recommendations for use of total size conversion methods

The total size conversion method applied to your own first-generation size measurements of 'whole' software items should produce sufficiently accurate converted COSMIC sizes to satisfy objectives as per Examples 1 and 2 in section 2.1. This method also demands the least effort for the conversion exercise.

However, the total size conversion method may not produce sufficiently accurate converted COSMIC sizes of all individual software items in a portfolio to meet the objective of Example 3 in section 2.1. For an explanation of the limitations on the accuracy of the total size conversion method, see section 3.1. This method is also unlikely in our judgement to produce sufficiently accurate converted sizes to meet the objective of Example 4. For these two objectives, the methods of Chapter 3 should be considered.

If an organization chooses to use a total size conversion method to convert first generation sizes to COSMIC sizes, we recommend the following.

- Establish your own total size conversion formulas based on data from your own software.
- Follow the process described in section 2.2 Steps 1 and 2 for selecting the sample for a conversion exercise, in particular grouping the software items to be converted on the basis of their functional homogeneity.
- Consider dividing the software items into size bands (limits to be determined locally) to obtain better OLS-fitted straight lines than fitting a single line over the whole size range.
- Consider 'direct' conversion (see section 3.2) to COSMIC for software items whose size is less than, say, 100 FP, depending on the objective for the conversion accuracy.
- Deal carefully with outliers from an OLS-fitted straight line as in section 2.3. (Do not remove 'Type 2' outliers purely on the basis of statistical tests; try to understand why a data point is an outlier).
- Compare FP 'Transaction' sizes (i.e. exclude the size contribution of files types) against CFP sizes (rather than total FP against CFP sizes) as in section 2.2, Step 4-d-i, to see if a better OLS fit can be obtained. (For other variations on seeking correlations between IFPUG/Nesma BFC types (sizes or numbers) and CFP sizes, see Appendix A.1.1.)
- Be prepared to correct converted sizes for effects such as described in Steps 4-d-I and ii.

3

‘DIRECT’ AND ‘COMPONENT’ METHODS FOR CONVERSION

3.1 The limitations of statistical conversion methods

Methods to convert sizes by the total size (statistical) conversion method described in Chapter 2 and used for the results shown in Appendices A and B have two inherent weaknesses.

- When deriving a conversion formula by a statistical method applied to a representative sample of the measurements to be converted, we can use experience and judgement, and even statistical methods, to eliminate outliers. However, if we need to apply the derived conversion formula to convert the bulk of existing measurements, we will not know the accuracy of any of the converted sizes. In particular, we will have no means of predicting which, if any, of the individual data points would be very inaccurately converted, i.e. would be outliers.

  All statistical conversion methods rely on the assumption that the dataset to be converted is reasonably homogenous in some way. However, they do not usually provide any formal test of that assumption other than tests to remove outliers from the sample used to determine the conversion formula.
- The conversion formula obtained for the sample may have a high (good) coefficient of determination (R-squared) implying good converted sizes ‘on average’. However, *individual* converted sizes can still be seriously wrong. The data shown in Figure 10, Figure 11 and Figure 12 in Appendix A.1 illustrate how the measured CFP versus FP sizes are scattered around an OLS regression-fitted straight line that could be used to predict converted CFP sizes. The error on converted sizes implied by this scatter could be very significant in contractual situations, or misleading in performance studies, in deriving estimation formulae, etc., especially for small software sizes. Ideally, we would like to improve on the accuracy of converted sizes. Even using the very closely-fitted straight lines shown in
- Figure 16, 17 and 18 in Appendix A.2 may result in significant inaccuracy of individual CFP sizes converted from MkII FP sizes, as will be discussed in section 3.3.

Therefore, in this chapter we introduce ‘direct’ and ‘component’ processes for conversion, which should provide more accurate converted sizes, but at the expense of needing more detailed data and greater conversion effort.

3.2 A process to convert IFPUG/Nesma to COSMIC sizes directly

‘Direct’ conversion of a functional size measured by IFPUG/Nesma to a COSMIC size of a ‘whole’ business application (i.e. not a component of an application) or of an enhancement to an existing application, may be possible when:

- the basic raw data of the first generation size and the artefacts used as input to sizing are available
- the measurer(s) have expertise in both IFPUG/Nesma and COSMIC methods and
- some expertise is available on the application, which can help make good judgments or intelligent guesses on the equivalence between the BFC types on the two methods.

Direct processes require knowledge of the FUR of the software whose size must be converted as well as detailed knowledge of the two measurement methods. The process must therefore almost inevitably be carried out manually.

The ‘raw data’ needed for the IFPUG/Nesma sizes would ideally be:

- the list, including names of file types and their record types;

- the counts of ‘DET’s (‘data attributes’ in COSMIC terminology) and FTR’s for each of the types of elementary processes.

As the IFPUG/Nesma and COSMIC methods measure the size of required enhancements to existing whole applications in quite different ways, we must separate the description of the direct processes for converting sizes of:

- applications, and enhancements to existing applications that consist of the addition of whole chunks of functionality (section 3.2.1), from
- enhancements to applications that consist of any mix of additions, modifications and deletions to existing functionality (section 3.2.2).

3.2.1. Direct conversion of sizes of applications or of additions of whole functionality

The process consists of the following steps.

1. Use the list of files and record element types (RET’s) to draw up the corresponding list of objects of interest about which persistent data is stored, showing the mapping. Consider every RET to see if it corresponds to a COSMIC data group.
2. Examine each IFPUG/Nesma elementary process (EI, EO or EQ) to determine the equivalent COSMIC functional process(es). They will mostly be equivalent, but not always, e.g. consider the text in Section 1.2.1. It may be necessary to measure some functionality using the COSMIC method directly, e.g. the maintenance and use of code tables that the IFPUG/Nesma method does not measure.
3. Examine the DET’s of each elementary process and, using the list of the objects of interest and other data (e.g. transient data arising in enquiries), determine the number of Entries and Exits for each functional process. Every group of DET’s that crosses the boundaries of the application inwards (or outwards) is potentially a COSMIC Entry (or an Exit, respectively).
4. Examine the FTR’s of each elementary process and, using the list of the objects of interest, determine the number of Reads and Writes for each functional process. Every FTR indicates a potential Read or Write, or both, depending on what the process does with the referenced file type.
5. The total of Entries, Exits, Reads and Writes over all functional processes will then be the size in CFP.

As a check on the accuracy of the result, an analogous manual conversion process could be run in reverse, starting with the COSMIC size components and list of objects of interest, to generate the equivalent IFPUG/Nesma size.

3.2.2. Direct conversion of enhancements to add, modify and delete existing functionality

This process is the same as that described in section 3.2.1 except that after step 4 examine the FUR to determine, for each EI, EO and EQ, which of the corresponding Entries, Exits, Reads and Writes are impacted by the required addition, change or deletion of functionality.

The total of the Entries, Exits, Reads and Writes that have been added, modified or deleted will then be the size of the enhancement in CFP. (For the definition of ‘modified’, see the Measurement Manual [1], section 4.4.1.)

3.3 ‘Component’ size conversion methods

To improve on the accuracy achievable by statistical conversion methods, we need a method that:

- produces a conversion formula that can be used to *calculate each COSMIC* size *directly* from data gathered for *all* the individual size measurements on the first generation method. (This contrasts with developing one OLS regression-fitted straight line which is used to convert all first generation method sizes to COSMIC sizes.)
- helps in assessing the homogeneity of the software items whose sizes have been measured using the first generation and COSMIC methods. Ideally, we would like to test the homogeneity of both the *sample* of software items whose sizes will be measured on both methods to establish the component size conversion formula, and of *all the other* software items whose first generation method sizes must be converted to COSMIC sizes. This homogeneity assessment test can be

used to predict which, if any, of the individual first generation method sizes would be poorly converted by the formula derived from the sample.

The statistical conversion processes discussed in Chapter 2 mainly rely on establishing a relationship between the total sizes measured by the first generation method and by the COSMIC method (though the hybrid formulas discussed in section Appendix A.1, section A.1.2 use only the size of IFPUG/Nesma transaction functions instead of the total FP size.)

In contrast, the component size conversion method relies on seeking relationships between:

- the counts of the components of BFC types that must be identified by the first generation method, i.e. the DET's and FTR's for the IFPUG/Nesma methods, or DET's and ER's for the MkII method;
- and the counts of COSMIC method BFC types.

In other words, a component size conversion method relies on measurements at a lower level of decomposition of software for first generation sizes than measurements at their BFC type levels. This level is the lowest level of decomposition that is shown as measured in **Figure 2**, in common for all three methods.

Establishing such relationships depends on three assumptions.

Assumption 1. We assume that IFPUG/Nesma Elementary Processes, MkII Logical Transactions and COSMIC functional processes are equivalent (see Table 2 and **Figure 2**). As mentioned in section 1.2, this may not be exactly true in all circumstances. The intention when drafting the three definitions was almost certainly the same and experience suggests the definitions are usually interpreted in the same way. MkII logical Transactions and COSMIC Functional processes are usually identical. If the task is to convert from IFPUG/Nesma to COSMIC sizes, the Measurer should check the equivalence and adjust this component conversion method if needed.

For convenience from here on, we will use the COSMIC term 'functional process' as the name for this concept for all three FSM methods.

Assumption 2. We assume that IFPUG/Nesma method references to Logical Files (known as 'File Types Referenced' or FTR's), MkII method references to stored data (known as Entity References or ER's) and COSMIC method references to persistently stored data (Reads and Writes) are equivalent. This assumption is less likely to be true than the first assumption, because:

- the IFPUG/Nesma Logical Files (ILF's or EIF's) rarely correspond exactly to the COSMIC 'objects of interest' about which data are stored (see section 1.1);
- the 'entities' about which data are stored according to the MkII method do not correspond exactly to the 'objects of interest' about which data are stored according to the COSMIC method. However, the correspondence is quite close. For a discussion of the differences, see [26];

Although the correspondences may not be exact in theory, what matters for the component size conversion method is whether we can establish some reasonably constant relationships between the ways the three methods measure references to these stored data artefacts in functional processes.

Assumption 3. We assume that the BFC types of all FSM methods may be regarded as conforming to a pattern where all functional processes have an Input, a Process, and an Output component. A consequence of this assumption is that:

a) The Input component of a functional process is measured by the count of input Data Element Types (DET's) identified for the functional process on the IFPUG/Nesma and MkII methods, and by the count of Entries on the COSMIC method.
b) Similarly, the Output component of a functional process is measured by the count of output DET's identified for the functional process on the IFPUG/Nesma and MkII methods, and by the count of Exits on the COSMIC method.
c) The Process component of a functional process is measured by the count of FTR's identified for the functional process on the IFPUG/Nesma method, the count of ER's for the MkII method, and the count of Reads-plus-Writes for the COSMIC method.

There are two important points to note at this stage.

- The 'Input/Process/Output' ('IPO') formula of a functional process is useful for our size conversion purpose but it is a convenient label, not an accurate description. For example, the Input phase of a functional process may well involve validation of input data against stored data (e.g. to see if the entered data is already stored). So in reality, Input functionality is better represented by both the entered DET's, and by one or more file or entity references, or Reads. Similarly, the Output phase may require one or more Reads to produce the DET's to be output. But for convenient labelling, we allocate all FTR's, ER's and the sum of Reads + Writes to the Process phase in this 'IPO' formula.

- The steps of the component size conversion method depend on the availability of the DET and FTR/ER counts for the first generation method sizes. This is usually no problem for MkII FP measurements, but is more likely to be a problem when adapting the process for IFPUG/Nesma to COSMIC conversion (see section 3.2).

This particular point about the unavailability of detailed DET and FTR counts for IFPUG/Nesma measurements may be the biggest obstacle to using the component size conversion method for converting IFPUG/Nesma FP to COSMIC CFP. It arises due to the 'size bounds' problem discussed in section 1.2.2 which means it is unnecessary to count and record DET's and FTR's above the upper size bounds in order to measure a IFPUG/Nesma size

We now describe such a 'component size conversion method' for MkII to COSMIC conversion and how it was applied in [26] to two samples of software items measured on both methods:

- first, to check the homogeneity of the data to be converted (section 3.3.1)
- second, to convert individual MkII sizes to COSMIC sizes (section 3.3.2).

The two samples of MkII and COSMIC size measurements are those for which the results of total size conversion by traditional statistical analysis are presented and discussed in Appendix A.2, namely:

- four Command and Control software systems from a single organization (see Figure 17)
- 13 Information Systems from a single organization (SITA, see Figure 18)

The main conclusion in Appendix A.2 from statistical analysis of the total sizes of these two sample datasets is that MkII FP and COSMIC sizes correlate very well. The R-squared values of both sample datasets (from different organizations and software domains) exceeded 0.99. However, this high R-squared masks the fact that the COSMIC size of individual software items converted from its MkII size may be quite inaccurate, especially for small software items.

In contrast, the results from applying the component size conversion method produce much more accurate individual converted COSMIC sizes than using the total size conversion method. For full details of the method, including potential refinements that are not described here, see [26].

In section 3.3.2, we also describe how this same component size conversion method could in principle be adapted for IFPUG/Nesma to COSMIC size conversion. Note, however, at the time of writing this application of the method has not been tested, so there is currently no evidence that it would yield more accurate converted CFP sizes than any of the 'traditional' total size conversion processes. **However, we believe it would be well worth exploring to find out if IFPUG/Nesma sizes can be more accurately converted to COSMIC sizes by component size conversion.**

3.3.1 Examining the homogeneity of the data to be converted

With the above three assumptions, we can determine the counts of BFC types for the Input, Process and Output components of each functional process as measured on the MkII method and for the COSMIC method. We then sum these counts over all functional processes for each software item as shown in Table 8, where '#' = 'count, summed over all functional processes'.

Table 8 - The counts needed to measure the functional size of the Input, Process and Output components, according to the MkII and COSMIC methods

FSM Method	Input component	Process component	Output component
MkII FPA	# Input DET's	# ER's	# Output DET's
COSMIC FSM	# Entries	# (Reads + Writes)	# Exits

The test that was applied in [26] to check the functional homogeneity of the sample dataset whose MkII size measurements should be converted to COSMIC sizes was to examine the 'IPO Profile' of the measurements. An IPO profile shows the relative contributions to the total size of the Input, Process and Output components of each software item that must be converted.

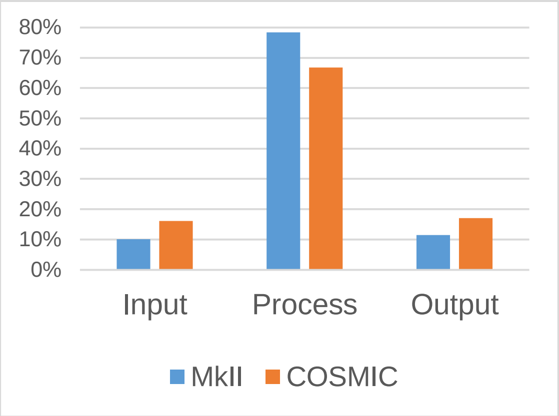
For MkII and COSMIC measurements, this test is straightforward. The relative Input, Process and Output contributions to total size are calculated as shown in Table 9. (The MkII relative size contributions use the weights of the Input, Process and Output components shown in Table 3.)

Table 9 - The contributions to total functional size of the Input, Process and Output components of a software item according to the MkII and COSMIC methods

FSM Method	Input component	Process component	Output component
MkII FPA	0.58 x (# Input DET's)	1.66 x (# ER's)	0.26 x (# Output DET's)
COSMIC FSM	# Entries	# (Reads + Writes)	# Exits

Figure 7 shows the average IPO profile, expressed as percentages of the contributions to total size, for the four Control Systems from a single organization, for which COSMIC versus MkII total sizes are shown in Appendix A.2, Figure 17.

Figure 8 shows the average IPO profile, expressed as percentages of the contributions to total size, for the 13 SITA Information Systems, for which COSMIC versus MkII total sizes are shown in Appendix A.2, Figure 18.



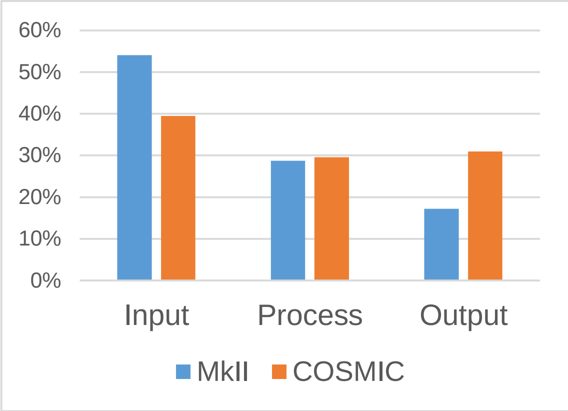


Figure 7 and Figure 8 - The 'IPO Profiles' according to the MkII and COSMIC methods for the Control Systems and SITA datasets respectively

From above figures, we note that:

- Considering each dataset separately, the MkII and COSMIC IPO profiles are very similar. The differences are due to the number of DET's on the input and output components which influence the MkII sizes but that do not influence the COSMIC sizes.
- Comparing the two datasets, the IPO profiles for the four Control Systems (in Figure 7) and for the 13 SITA Information Systems (in Figure 8) are quite different. (The former profile is unusual, with a high proportion of size due to the process component. This is because these are four air traffic control systems that need to read and write very large amounts of persistent data to keep track of aircraft in relation to airports, 'way-points' along their route, etc.)

Besides the 13 SITA systems and the four control systems, the authors of the MkII to COSMIC size conversion study [26] examined the MkII and COSMIC IPO profiles of seven other systems from three other organizations. Their IPO profiles were found to be different from those shown in Figures 7 and 8. The data suggests that the homogeneity of a set of software systems can be tested by this form of IPO profiling before starting to apply the component size conversion method. The test would aim to identify individual software items in a dataset that might not be accurately converted because they have an IPO profile different from the norm for the dataset.

3.3.2 The component size conversion method applied to MkII to COSMIC size conversion

From the discussions in section 1.2.1 on non-matching BFC types and in section 3.2.1 on IPO profiling, we know that the CFP size of an individual software item converted from a MkII FP size using the OLS fitted straight line process described in section 2.2, may vary significantly from the measured CFP size because the software item has:

- an exceptionally high or low count of DET's on its input and/or output compared with the average for all software, which results from the 'smoothing out' of the fitted straight line process;
- and/or an exceptionally high or low count of entity references compared with the average for all software implied by the fitted straight line (see Figure 2).

This suggests that a MkII to COSMIC size conversion method that takes into account the actual relative contributions of input, process and output functionality to the total size for each individual software item would lead to more accurate converted CFP sizes than using the OLS fitted straight line process. This is because statistical processes use one 'average' formula for the whole set based on total sizes, ignoring these variations in counts of DET's and entity references.

The 'component size conversion method' as it was applied in [26] for MkII to COSMIC size conversion (designated 'component size conversion method I' in Table 10) is as follows. (Entries, Exits, Reads and Writes are abbreviated as E, X, R and W respectively.)

a) Each set of software items with a 'common IPO profile' was considered separately and used the counts of their component DET's and ER's for the MkII sizes, and counts of E's, X's, R's and W's for the COSMIC sizes.

b) These counts were summed for each component over the whole set, designating the totals as:

   ∑ Input DET's, ∑ Output DET's and ∑ ER's, for the MkII FP counts

   ∑ E's, ∑ X's, ∑ R's and ∑ W's for the COSMIC counts

c) The following ratios were computed from these sums for the whole set:

   AIDE = Average Input DET's per Entry = (∑ Input DET's) / ∑ E's

   AODX = Average Output DET's per Exit = (∑ Output DET's) / ∑ X's

   AERP = Average Entity Refs per (R + W) data movement = ∑ ER's / (∑ R's + ∑ W's)

d) The CFP size of each individual software item was determined using the sums of the DET's and ER's for the components of its measured MkII FP size, and using formula (4) in order to obtain a 'component converted CFP' size.

$$\text{Component converted CFP} = (\sum \text{Input DET's}) / \text{AIDE} + (\sum \text{Output DET's}) / \text{AODX} + (\sum \text{ER's}) / \text{AERP} \quad (4)$$

For the results of this component size conversion method I', see Table 9 below.

This component size conversion method was further refined (to 'component size conversion method II') when it was found that for the 13 SITA software items the values of the ratios:

   IDE = (∑ Input DET's) / ∑ E's, and

   ODE = (∑ Output DET's) / ∑ X's,

vary with MkII FP size, as shown in Figure 9 (The value of ERP, i.e. the ratio of ∑ ER's / (∑ R's + ∑ W's) hardly varied at all with MkII FP size.)

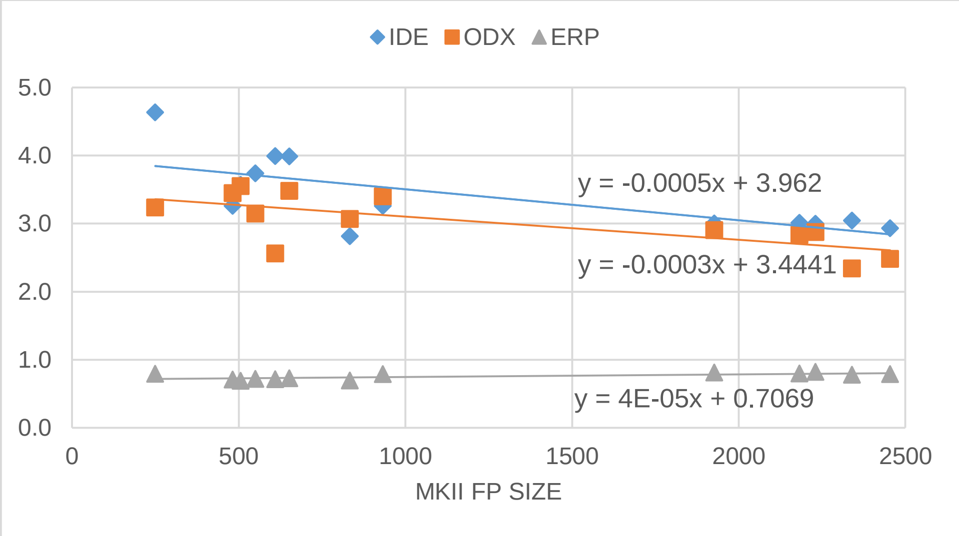


Figure 9. Variations of IDE, ODE and ERP with MkII FP size (13 SITA software items)

Consequently, 'smoothed' values of IDE, ODX and ERP were taken from the OLS-fitted straight lines in Figure 9 for each individual MkII FP size to be converted (instead of the fixed average values AIDE, AODE and AERP as in formula (4) for all MkII FP sizes). Formula (5) (a refinement of formula (4)), was then used to produce CFP sizes according to the 'component size conversion method II'.

Component converted CFP = (∑ Input DET's) / IDE
+ (∑ Output DET's) / ODX
+ (∑ ER's) / AERP (5)

Table 10 shows the results of

- using the OLS-fitted straight line of Appendix A.2, Figure 18 to convert the MkII to COSMIC sizes for the 13 SITA systems, i.e. using 'total size conversion',
- and of applying the component size conversion method I and II to the conversion of the MkII to COSMIC sizes for the same 13 SITA Information Systems.

Table 10 - Comparison of the accuracy of COSMIC FP sizes converted from MkII FP sizes for 13 SITA Information Systems by OLS total size conversion and by component size conversion methods I and II

Conversion Method	MMRE on 13 converted vs measured CFP sizes	Number of under-sized software items	Number of over-sized software items	The three highest % errors on converted vs measured CFP sizes
OLS-fitted straight line	6.3%	4	9	+28%, +14%, +7.3%
Component size conversion I	6.4%	6	7	+18%, +11%, -11%
Component size conversion II	3.8%	6	7	-11%, 8.9%, -6,4%

The results show clearly that component size conversion method I produces more accurately-converted individual CFP sizes than the total size conversion process, and that component size conversion method II produces more accurately-converted sizes than component size conversion method I.

In particular, it is worth noting that although total size conversion may produce reasonably accurate converted CFP sizes (the Mean Magnitude Relative Error (MMRE) is 6% for this SITA dataset):

- the OLS-converted CFP sizes are biased towards over-sizing, whereas the component size converted CFP sizes are balanced between over- and under-sizing;
- individual converted CFP sizes obtained from the total size conversion process are significantly less accurate than those obtained by the component size conversion methods I and II.

As a further test, the total size conversion process and the component size conversion methods I and II were applied to the four Control Systems projects whose measured sizes are shown in Figure 17, and whose IPO profile is shown in Figure 8. The results are shown in Table 11.

As there are only four pairs of measurements in this dataset, the results have very limited statistical significance. However, the improvement in accuracy of the converted sizes obtained by using the component size conversion methods I and II over the accuracy obtained by using the total size conversion process is similar in magnitude to the results obtained above for the 13 SITA Information Systems.

Table 11 - Comparison of the accuracy of COSMIC FP sizes converted from MkII FP sizes for four Control Systems by OLS total size conversion and by component size conversion methods I and II

Conversion Method	MMRE on 4 converted vs measured CFP sizes	Number of under-sized software items	Number of over-sized software items	The two highest % errors on converted vs measured CFP sizes
OLS fitted straight line	9.7%	1	3	+25%, +11%
Component size Conversion I	8.6%	2	2	+17%, -8.4%
Component size Conversion II	6.6%	2	2	+12%, -9.9%

3.3.3. Summary description of the component size conversion method

Generalising this method so that it can be applied to a large population of software items for which MkII sizes have been measured, that must be converted to COSMIC sizes, involves the following steps.

- Use general knowledge of the whole population to group the software items into homogeneous datasets, as in section 2.2, step 1.
- For each resulting dataset, select a sample of software items that you believe to be representative of the dataset and re-measure their sizes using the COSMIC method, as in section 2.2, steps 2 and 3.
- Determine the IPO size profile according to the MkII and COSMIC methods for all software items in the sample, as in section 3.3.1. Use the profile to determine if the sample is truly homogeneous. If necessary, adjust the membership of the sample to obtain homogeneity.
- Follow steps a) to d) of the component size conversion method to derive formula (4) for the sample, and then formula (5) if using the latter results in more accurate converted CFP sizes.
- Examine the IPO profile according to the MkII method of the remaining members of the dataset to check that their profile is in line with that of the sample. Remove software items from the dataset whose IPO profile is obviously out of line with that of the bulk of members of the dataset and of the sample. (Add the removed software items to another group, or convert their MkII sizes directly.)
- Apply the formula for the chosen component size conversion method to convert the MkII sizes of the remaining members of the dataset.

3.4. Adapting the component size conversion method for IFPUG/Nesma to COSMIC conversion

Successfully adapting the method for IFPUG/Nesma to COSMIC size conversion depends on making some of the same assumptions as were made for component size conversion method in section 3.3. The method:

- will only be valid for functional sizes measured by these two methods of 'whole' business applications or of enhancements to existing applications that consist of the addition of whole chunks of functionality – together called 'software items'. (The process is not valid for conversion of the size of a component of an application, nor for an enhancement to an existing application that comprises a mix of additions, modifications and deletions.);
- requires the basic raw data of the IFPUG sizes and the artefacts used as input to sizing to be available.

For IFPUG/Nesma measurements, if the DET and FTR counts have been recorded for the software items only in sufficient detail to determine the simple, average or complex classification of EI's, EO's and EQ's, then the Measurer will have to explore how to set average values of these counts on the basis of the data that has been recorded.

It will also be helpful if some expertise on the software items whose sizes are to be converted is available to assist with deciding on the equivalence between the BFC types on the two methods and how to proceed if the detailed DET and FTR counts have not been recorded.

The three assumptions made in section 3.3 must be re-examined to check if they are still valid for IFPUG/Nesma to COSMIC size conversion.

The first assumption concerns the equivalence of IFPUG/Nesma elementary processes to COSMIC functional processes. As mentioned in section 3.2 in the context of 'direct' conversion, some functionality such as the maintenance of code tables is not measured by the IFPUG/Nesma methods. The COSMIC size of this functionality will have to be measured directly.

The second assumption on the equivalence of IFPUG/Nesma logical files to COSMIC objects of interest for persistently stored data is unlikely to be true. However, provided the measurements show that the ratio of the count of FTR's to the count of Reads-plus-Writes summed over all functional processes is reasonably constant, this mis-match should not matter. The validity of this assumption can only be checked by examining actual measurement data.

The third assumption, that IFPUG elementary processes can be represented by the IPO formula may be sufficiently accurate if we allocate:

- the count of DET's recorded for an External Input (EI) all or mostly to the Input component of this functional process;
- the count of DET's recorded for an External Output (EO) or an External Enquiry (EQ) all or mostly to the Output component of this functional process.

Again, the Measurer will need to check whether this assumption for IFPUG/Nesma measurements is approximately true for the software under study and adjust the allocations of DET's as necessary.

Table 12 then shows the equivalence of the IFPUG/Nesma and COSMIC measures of the Input, Process and Output components summed over all IFPUG/Nesma elementary processes and all COSMIC functional process respectively.

Table 12 - The counts that are needed to measure the functional size of the Input, Process and Output components according to the IFPUG/NESMA and COSMIC methods

FSM Method	Input component	Process component	Output component
IFPUG/Nesma	# Input DET's	# FTR's	# Output DET's
COSMIC FSM	# Entries	# (Reads + Writes)	# Exits

The component size conversion method applied to convert IFPUG/Nesma to COSMIC sizes should then proceed as described in sections 3.3.2 and 3.3.3, noting that:

- the IPO profiles, that is the percentage contributions of the IPO components to total size from the IFPUG measurements, could be calculated using an arbitrary set of weights similar or identical to those developed for the MkII method (as in Table 6);
- for the final IFPUG/Nesma to COSMIC size conversion, the same process should be followed as in section 3.3.2, (steps a) to d), at least as far as formula (4)), but substituting the sums of IFPUG/Nesma FTR's for the sums of MkII ER's for each software item.

As mentioned earlier, this method for converting IFPUG/Nesma to COSMIC sizes is an adaptation of the method that has been demonstrated to work successfully for MkII to COSMIC size conversion. The method is an idea that has not yet been tested with real measurement data.

3.5 Conclusions on direct and component size conversion methods

'Traditional' total size (statistical) conversion methods are the easiest to apply but suffer the disadvantage that the accuracy of converted sizes will be unknown except for the sample of sizes used to establish the conversion formula. 'Direct' and 'component size conversion' methods should therefore be considered when large amounts of first generation sizes must be converted.

Direct conversion of a first generation functional size to a COSMIC size as described in section 3.2 should produce the most accurate converted sizes, but this approach requires the availability of detailed data and the most effort. It may also be worthwhile applying direct size conversion for sizes of small software items in preference to total size conversion as the latter may produce inaccurately converted small sizes,

The component size conversion methods described in section 3.3 produce more accurate converted sizes than by applying total size conversion, at least as has been demonstrated for MkII to COSMIC size conversion of two small datasets,.

We believe that the component size conversion method could be adapted for IFPUG/Nesma to COSMIC size conversion if the detailed DET and FTR counts are available at the Elementary Process level, at least for the sizes of whole applications or of enhancements to existing applications that consist of the addition of whole chunks of functionality. Such a method should be more accurate than conversion by a total size (statistical) conversion process and would require much less effort than a direct conversion process. However, there is no evidence yet to support this belief.

We encourage researchers to explore this application of component size conversion.

REFERENCES

NOTE. All COSMIC publications are available for free download from www.cosmic-sizing.org

[1] The COSMIC Functional Size Measurement Method v4.0.1: Measurement Manual. (The COSMIC Implementation Guide for ISO/IEC 19761:2011), April 2015.

[2] The COSMIC Functional Size Measurement Method v3.0: Advanced and Related Topics, December 2007

[3] IFPUG Function Point Counting Practices Manual, Release 4.3.1, International Function Point User Group, January 2010. URL: www.ifpug.org

[4] MkII FPA Counting Practices Manual Version 1.3.1, United Kingdom Software Metrics Association, 1998

[5] ISO/IEC 14143-1: 2007 Software and Systems Engineering – Software measurement – Functional size measurement – Definition of concepts.

[6] ISO 14143-5:2004 Technical Report 'Information technology -- Software measurement -- Functional size measurement -- Part 5: Determination of functional domains for use with functional size measurement

[7] T. Fetcke, The warehouse software portfolio: A case study in functional size measurement. Technische Universität Berlin, Fachbereich 13, Informatik, 1999.

[8] C. Symons, "Conversion between IFPUG 4.0 and MKII Function points," Software Measurement Services Ltd., Version, vol. 3, 1999.

[9] J-M Desharnais, P. Morris, "Validation Process in Software Engineering: An example with Function Point", Software Metrics, Research and Practice in Software Measurement, Information Engineering und IV Controlling, Springer, 1997

[10] F. Vogelezang and A. Lesterhuis, "Applicability of COSMIC Full Function Points in an administrative environment: Experiences of an early adopter," in Proceedings of the 13th International Workshop on Software Measurement–IWSM 2003, 2003.

[11] A. Abran, J. M. Desharnais, and F. Aziz, "Measurement convertibility-from function points to COSMIC-FFP," Delta, vol. 4, no. 3, p. 2.

[12] J. M. Desharnais, A. Abran, and J. Cuadrado, "Convertibility of Function Points to COSMIC-FFP: Identification and Analysis of Functional Outliers," IWSM/MENSURA, p. 190, 2006.

[13] J. J. Cuadrado-Gallaego, D. Rodríguez, F. Machado, and A. Abran, "Convertibility between IFPUG and COSMIC functional size measurements," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2007, vol. 4589, pp. 73-283.

[14] C. Gencel and O. Demirors, "Conceptual Differences Among Functional Size Measurement Methods," in Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on, 2007, pp. 305-313.

[15] H. van Heeringen, "Changing from FPA to COSMIC-A transition framework," in Software Measurement European Forum, 2007.

[16] J. Cuadrado-Gallaego, L. Buglione, R. Rejas-Muslera, and F. Machado-Piriz, "IFPUG-COSMIC Statistical Conversion," Proc. of The 34th Euromicro Conference On Software Engineering, pp. 427-432, 2008.

[17] J. J. Cuadrado-Gallaego, F. Machado-Piriz, and J. Aroba-Paez, "On the conversion between IFPUG and COSMIC software functional size units: A theoretical and empirical study," Journal of Systems and Software, vol. 81, no. 5, pp. 661-672, 2008.

[18] C. Gencel and O. Demirors, "Functional size measurement revisited," ACM Transactions on Software Engineering and Methodology, vol. 17, no. 3, p. 15 (36 pp.), Jun. 2008.

[19] O. Demirors and C. Gencel, "Conceptual association of functional size measurement methods," IEEE Software, vol. 26, no. 3, pp. 71-8, May. 2009.

[20] J. J. Cuadrado-Gallaego, L. Buglione, M. J. Domínguez-Alda, M. F. d Sevilla, J. Antonio Gutierrez de Mesa, and O. Demirors, “An experimental study on the conversion between IFPUG and COSMIC functional size measurement units,” Information and Software Technology, vol. 52, no. 3, pp. 347-357, 2010.

[21] P. Efe, C. Gencel, and O. Demirors, “Mapping Concepts of Functional Size Measurement Methods,” in Cosmic Function Points: Theory and Advanced Practices, CRC Press, 2010.

[22] L. Lavazza, “A systematic approach to the analysis of function point COSMIC convertibility,” presented at the 20th International Workshop on Software Measurement, IWSM/Mensura, Stuttgart, 2010.

[23] L. Lavazza and S. Morasca, “Convertibility of Function Points into COSMIC Function Points: A study using Piecewise Linear Regression,” 2011.

[24] C. Gencel, C. Bideau, "Exploring the Convertibility between IFPUG and COSMIC Function Points: Preliminary Findings", IWSM/Mensura 2012 Conference, IEEE CS, pp. 170-177, 2012.

[25] J.M. Amiri, V.V.Kumar, ”A Comprehensive Evaluation of Conversion Approaches for Different Function Points”, MSc Thesis, Blekinge Institute of Technology, Sweden, September 2011.

[26] A. Dasgupta, C Gencel, C.R. Symons. “A process to improve the accuracy of MkII FP to COSMIC size conversions: insights into the COSMIC method design assumptions”, IWSM/Mensura, Krakow, October 2015

[27] A. Abran, “Software Project Estimation”, IEEE Press/Wiley, 2015 (section 5.4.3)

[28] F.E. Grubbs, “Procedures for detecting outlying observations in samples”, Technometrics 11, pp.1-2.

[29] Lavazza, Luigi. "Convertibility of functional size measurements: new insights and methodological issues." Proceedings of the 5th International Conference on Predictor Models in Software Engineering. ACM, 2009.

[30] Lavazza, Luigi. "An evaluation of the statistical convertibility of Function Points into COSMIC Function Points." Empirical Software Engineering 19.4 (2014): 1075-1110.

[31] S. Di Martino, F. Ferrucci, C. Gravino, F. Sarro, "Web Effort Estimation: Function Points Analysis vs. COSMIC", Information and Software Technology (2016), pp. 90-109, DOI: 10.1016/j.infsof.2015.12.001.

[32] F. Ferrucci, C. Gravino, F. Sarro, "Conversion from IFPUG FPA to COSMIC: within- vs without-company equations", in Proceedings of the 40th Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2014), pp. 293-300.

[33] A. Corazza, S. Di Martino, F. Ferrucci, C. Gravino, F. Sarro, "From Function Points to COSMIC - A Transfer Learning Approach for Effort Estimation", 16th International Conference on Product-Focused Software Process Improvement (PROFES 2015), pp. 251-267.

[34] Cook R., Weisberg S., (1982) ‘Residuals and Influence in Regression’, vol 5, Chapman and Hall, New York.

[35] ‘Introduction to the COSMIC method of measuring software, v4.0.1’, v1.1, January 2016.

APPENDIX A – REVIEW OF SIZE CONVERSION STUDIES

This Appendix presents a summary of existing total size conversion studies that have sought to establish a relationship between sizes obtained by 1G methods to corresponding COSMIC sizes.

The aim of showing the results of existing conversion studies is to illustrate the variety of results that have been obtained, so that we can discuss the factors that may cause the variations in the results. We strongly recommend that Measurers gather data to create their own conversion formulas. We do NOT recommend anyone to use the results shown here.

(In the following we refer to 'project sizes', which is the term used in all the studies cited, rather than to 'software sizes delivered by projects', which is what is really meant.)

A.1 IFPUG/Nesma to COSMIC conversion formulas in the literature

A.1.1 Converting IFPUG/Nesma total (i.e. transaction PLUS data) sizes

In the literature, there are 8 published datasets (DS) (see Appendix B for details) in total consisting of 135 projects where total sizes have been measured of the same applications by both the IFPUG / Nesma and COSMIC methods [25]. All measurements are of whole business application software systems. None of the measurements are of enhancements and/or of components of whole applications,

Apart from the Van Heeringen dataset (DS5) and Cuadrado Gallego et al. datasets (DS6, DS7, DS8), all other datasets include measurements collected almost entirely within a single organization. The scatter plot diagrams for some example datasets such as DS2 (see Figure 10), DS4 (see Figure 11) and DS5 (see Figure 12) are given below.

These datasets have been studied in the literature using various statistical analysis techniques such as ordinary least squares (OLS), log-log regression[6], least median of squares (LMS), and piecewise linear regression. These studies in total resulted in 19 different conversion formulas (CM) (for details see Appendix B, Table 1).

Most of the resulting regression lines of the form y = ax+ b had slopes between 0.73-1.22 and passed through a point above zero on the IFPUG/Nesma scale for a size of zero on the COSMIC scale. (This corresponds to a negative value 'b', as discussed in section 2.2, Step 4-c, and as expected.)

The follow-up studies aimed to improve the conversion formulas by paying more attention to mainly 'Type 2' outlier data points in the datasets (see Appendix B, Table 1), i.e. data points that are dispersed from the OLS-fitted straight line. However, as far as we can tell, none of the datasets studied in the literature were tested for functional homogeneity before being analysed to find a conversion formula. Indeed, some of the datasets are known to include data from several organizations. They also comprise a large functional size range.

(The dataset DS2 is the set analysed in Section 2.3. This dataset could be analysed in detail to detect outliers as the authors of this Guideline had more information regarding the projects in this dataset.)

[6] Readers should be aware of the risks of applying regression analysis to data plotted on a log-log scale. The fitted regression line may appear to be reasonable and yield a high R-squared but the graph may hide problems. For example, when plotted on a log-log scale, the existence of type 1 outliers may not be as obvious as when the same data are plotted on a linear-linear scale. Also, the 'break-point' phenomenon (see below) would not show up so appreciably. For more see [27].

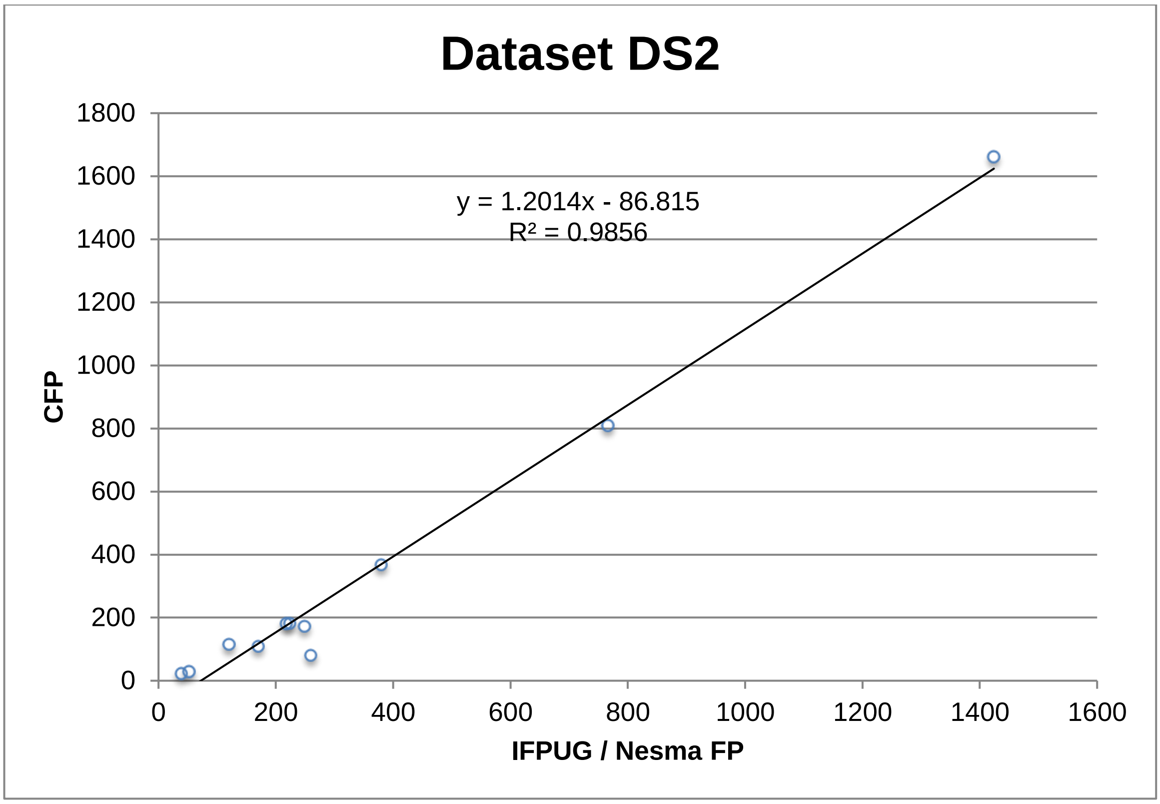


Figure 10 - The Relationship between IFPUG/Nesma FP and COSMIC CFP - Dataset DS2 [10]

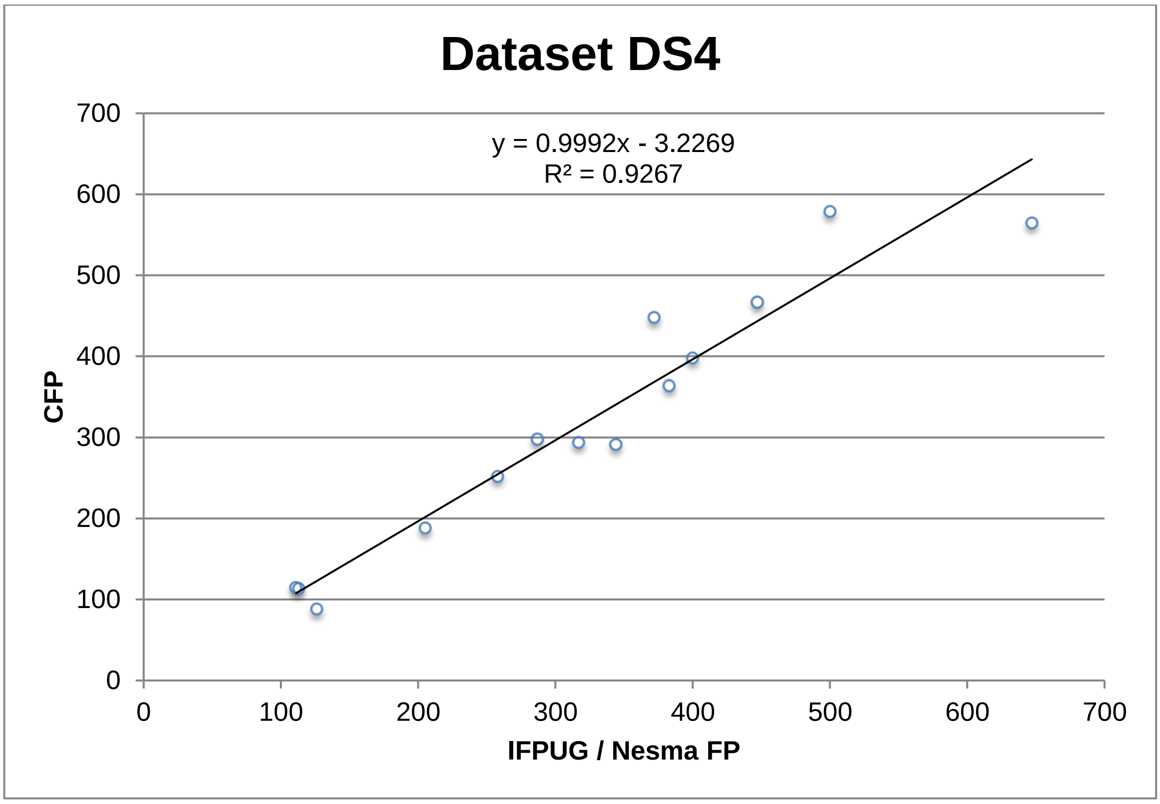


Figure 11 - The Relationship between IFPUG/Nesma FP and COSMIC CFP - Dataset DS4 [12]

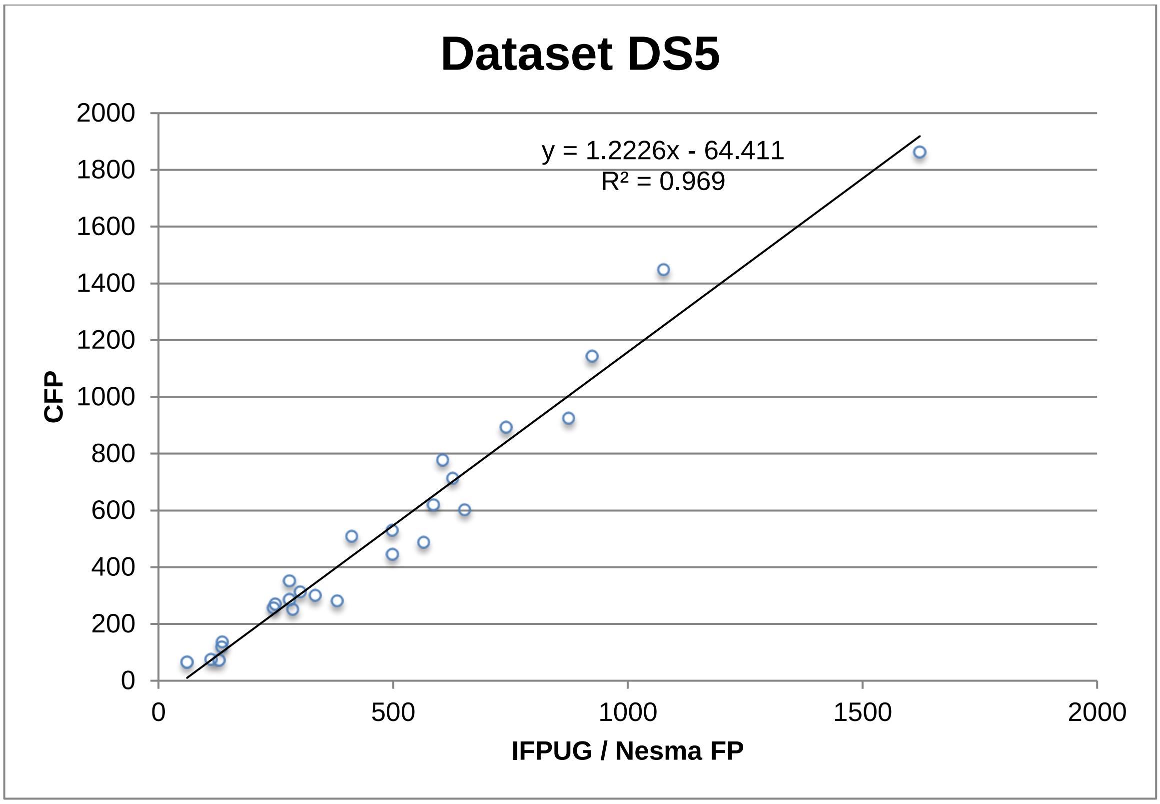


Figure 12 - The Relationship between IFPUG/Nesma FP and COSMIC CFP - DS5 [14]

Figure 13 below provides an overall picture of the nature of the relationship between IFPUG/Nesma FP and CFP sizes based on the data pairs of all 135 projects from the 8 datasets (without distinguishing between the IFPUG or Nesma measurements). The R-squared value seems to be high.

However, if we look at the data closely, it appears to hide information, which is revealed by splitting the data between Figure 14 showing the large number of projects that delivered sizes up to 400 FP and Figure 15 showing the data for the projects between 400-2000 FP.

These graphs show clearly how much error one can make in a conversion exercise without considering the functional homogeneity of the data. Basically no statistical test for outliers could help, as the data pairs are too widely spread, indicating significant systematic differences between projects.

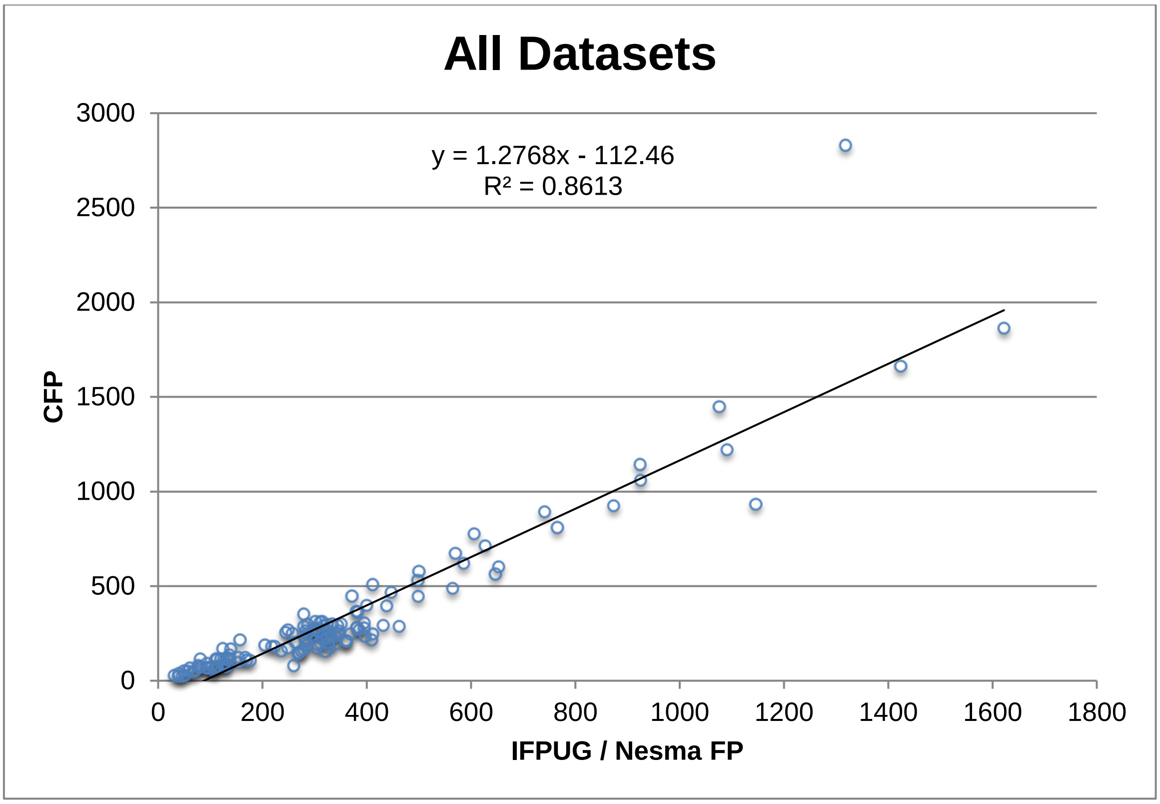


Figure 13 - The Relationship between IFPUG/Nesma FP and COSMIC CFP for all datasets merged (135 projects)

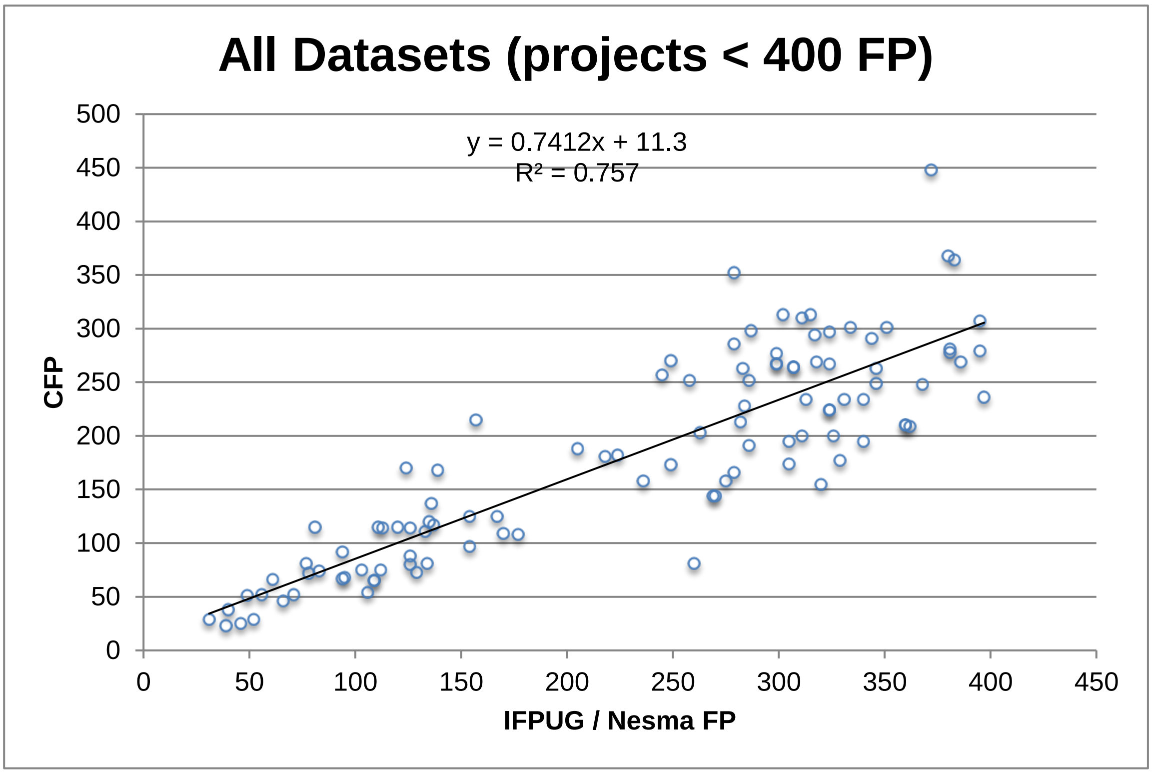


Figure 14 - The Relationship between IFPUG/Nesma FP and COSMIC CFP for all datasets (Sizes up to 400 IFPUG FP)

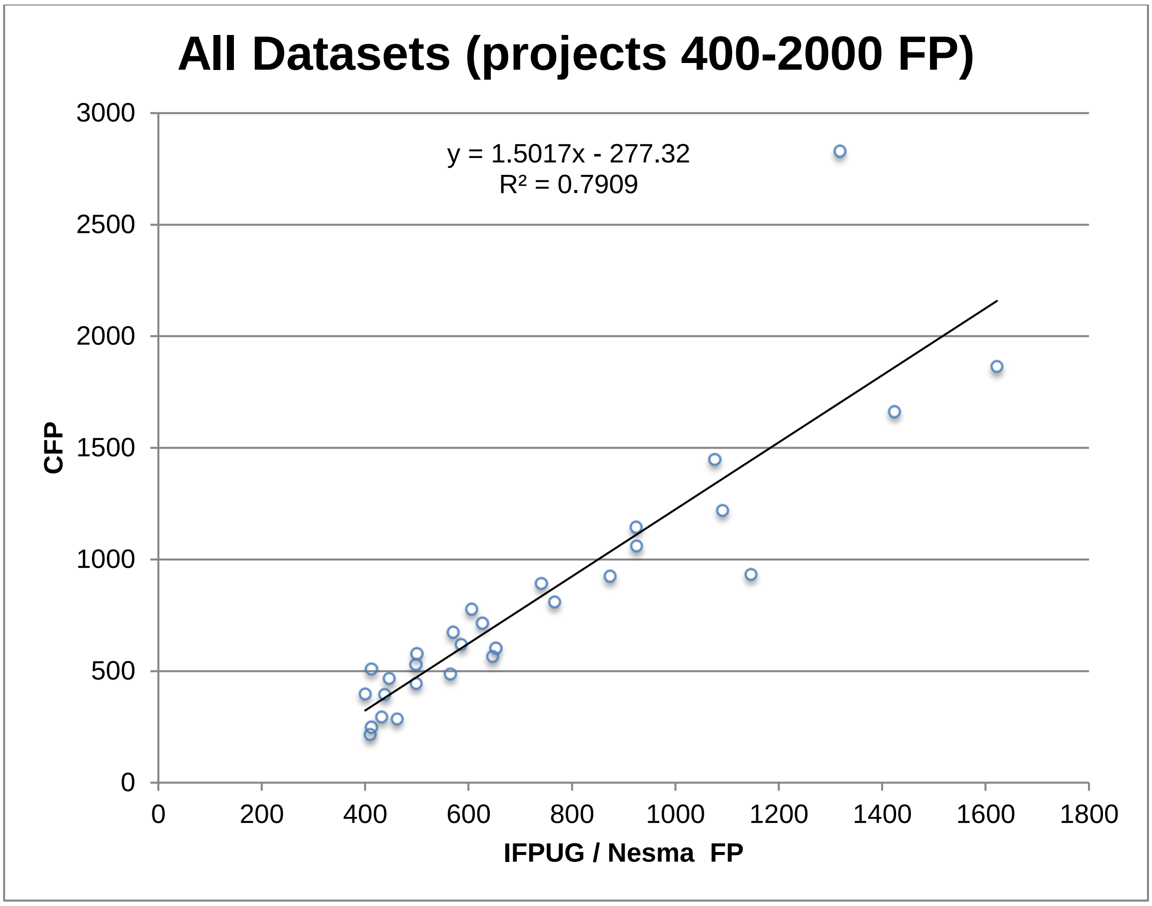


Figure 15 - The Relationship between IFPUG/Nesma FP and COSMIC CFP for all datasets (projects with sizes between 400-2000 IFPUG FP)

Some studies have examined whether 'break-points' exist in the linear relationship between IFPUG/Nesma and COSMIC sizes by dividing measurements into size segments of different size ranges (as was done above in dividing the whole 135 projects into sizes up to 400 FP (Figure 14) and above 400 FP (Figure 15).

First, Abran et al. [11] identified that the intercept in their formula is relatively high which might be due to one big project in the Vogelezang and Lesterhuis dataset. So, they split the datasets into two parts: projects a) less than 200 FP and b) larger than 200 FP. Later, Lavazza & Morasca [22] applied piecewise linear regression to identify break-point(s) in a more systematic way (see CM10, CM14, CM16 in Appendix B, Table 1). However, it appeared that for different datasets, there exist different break-points (e.g. 317 FP, 279 FP, 606 FP). Hence, it was not possible to derive one formula for sizes below some particular point and another one for the sizes above. Lavazza & Morasca [22] also stated that they could find no statistically significant evidence that one particular formula can represent a dataset better than other formulas. When all datasets are considered, it is also unclear from statistical analysis whether break-points are a statistically-significant phenomenon or if they arise by chance for the particular available data.

In fact, if we draw on our knowledge of how the IFPUG/Nesma method measures functional size, compared to the COSMIC method, we can hypothesize why a break-point might *appear* to exist in a given dataset, and why the position of the break-point may vary with datasets of measurements from different portfolios. Two factors must have an effect, as already discussed in relations to Figure 3.

- The IFPUG/Nesma size of a software item is the sum of the size of its Elementary Processes (EP's) and the size of its Logical Files (LF's, i.e. the sum of the sizes of its ILF's and EIF's). Small software items tend to have a high contribution from the LF sizes relative to the contribution from EP sizes because the whole size of every LF that is referenced must be counted, even though there may be only a few EP's making the references. For increasingly large software items, the proportion of the size contribution of the LF's to the total size decreases because more and more EP's are referencing the same LF's, and the LF sizes are only counted once in the total size of the software item being measured. This is confirmed by Lavazza in a study of data in the ISBSG database [30]. For the largest items in a software portfolio, a point may be reached where *all* the LF's have been accounted for and size increases due only to more and more EP's referencing those LF's.

- The size of IFPUG/Nesma EP's is limited by their upper size bounds.

In contrast, for a functionally homogeneous software portfolio measured using the COSMIC method:

- the size of software items increases only with the number of their functional processes. We would not expect the contribution of Entries, Exits, Reads and Writes as a proportion of total size of an item in the portfolio to change as the number of its functional processes increases.
- COSMIC functional processes typically have similar sizes to IFPUG/Nesma EP's up to the latter's maxima. Above these maxima, COSMIC functional processes moving many DET's (i.e. probably with many Entries and Exits) and with many FTR's (i.e. many Reads and Writes) will result in sizes in CFP far exceeding the 7 FP maximum of an IFPUG/Nesma. EP (see section 1.2.2).

Thus we would expect that a graph of COSMIC sizes (y-axis) versus IFPUG/Nesma sizes (x-axis) with a very large number of data pairs would appear as a curve that rises slowly upwards (as already pointed out in Step 1 of the 'total size conversion method' in section 2.2).

The data in Figure 13 give the impression that the relationship may be curved**.** In fact, the existence of a quadratic model (i.e., $CFP = a + b\ FP^2$) is documented in [23], for the dataset by Cuadrado-Gallego. Furthermore, in [30], Lavazza tested the goodness of fit of various linear and non-linear models to four datasets. He concluded that 'non-linear models are generally applicable' and that a straight-line 'cannot describe with reasonable accuracy the CFP/FP relationship' (over the size range for which we have data).

In reality, there are no fixed 'break-points' (implying a discontinuity) in the overall IFPUG/COSMIC size relationship. But for any given dataset, the relationship will probably exhibit a 'zone' where the slope increases appreciably.

Consequently, it is perfectly possible that a small dataset, which typically has several measurements of small software items and only a few larger software items will appear to show a 'break-point', i.e. a point where the slope appears to increase perceptibly. Where this 'break-point' appears will obviously vary with the individual dataset. Even for the dataset of 135 projects available for this Guideline, a break-point *appears* around 400 FP, which we have exploited to fit different straight lines above and below 400 FP. But this is a chance result of the data that happens to have been collected.

A.1.2 Converting IFPUG/Nesma transaction sizes

As discussed in section 1.2, a major difference between IFPUG/Nesma and COSMIC is that the IFPUG method measures data files (ILF and EIF) and transaction functions (EI, EO, and EQ) separately whereas COSMIC does not. Consequently various authors have studied the relationship between the sizes (or numbers) of IFPUG/Nesma BFC's and COSMIC total sizes. We are aware of the following studies.

- Abran et al. [11] proposed that there is a possibility of deriving better conversion formulas by considering only the size of IFPUG/Nesma transactions (FP-TX) instead of total FP sizes when converting to CFP.
- Desharnais et al. [12] proposed a similar conversion formula using another dataset.
- Lavazza has published various studies (22], [23] and [29]. In [30] he concluded **that 'in no cases the model based on transaction functions was significantly less precise than the models based on the full count of FP'.**

Table 2 in Appendix B shows the results of the conversion formulas obtained based on the IFPUG/Nesma FP-TX size and total COSMIC size. The results showed higher correlation coefficient values than the methods based on total (transaction + data) sizes discussed in Appendix A.1 section A.1.1.

According to Desharnais and Morris [9] and Vogelezang & Lesterhuis [10], 30% to 40% of IFPUG functional sizes are due to logical files. Different application types may have different (data size) / (transaction size) ratios as measured by the IFPUG/Nesma methods. However, if this ratio is not showing too much variation in particular contexts, then this factor may not influence the converted sizes based on either IFPUG/Nesma total FP or only FP-TX sizes.

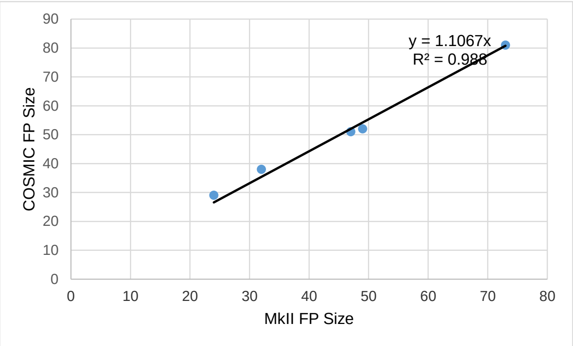
As there have been no further studies performed, we do not have sufficient evidence at present to strongly prefer using this approach rather than using total sizes when developing conversion formulas. Still, we strongly recommend investigating the relationship between the sizes of transactions as well as the relationship between the data size and transaction size of IFPUG/Nesma measurements. This may help in identifying outlier software items and derive more accurate conversion formulas.

A.2 MkII to COSMIC total size conversion formulas in the literature

Three studies have provided datasets of projects whose software sizes were measured both by MkII and COSMIC (see Appendix B, Table 3), which makes it possible to investigate the relationship between MkII and COSMIC sizes.

Figure 16 shows the relationship between COSMIC and MkII sizes based on Fetcke 1999 Dataset – DS1 (Warehouse management) [7].

Figure 17 shows the relationship between COSMIC and MkII sizes for four Command and Control software systems from a Gencel and Demirors dataset [18].



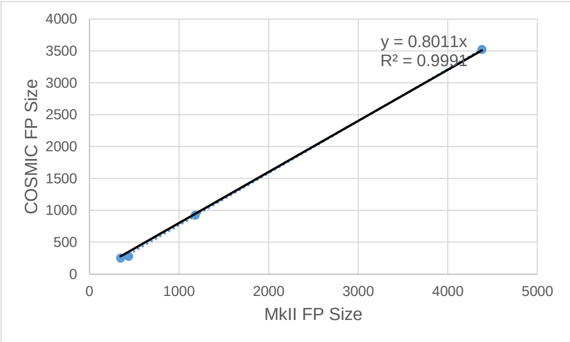


Figure 16 and Figure 17 - Relationships between COSMIC and MkII sizes for two datasets

Figure 18 shows the COSMIC versus MkII FP sizes for 13 Information Systems from a single organization (SITA), reported by Dasgupta, Gencel and Symons [26]. These systems support operations in the air transportation industry.

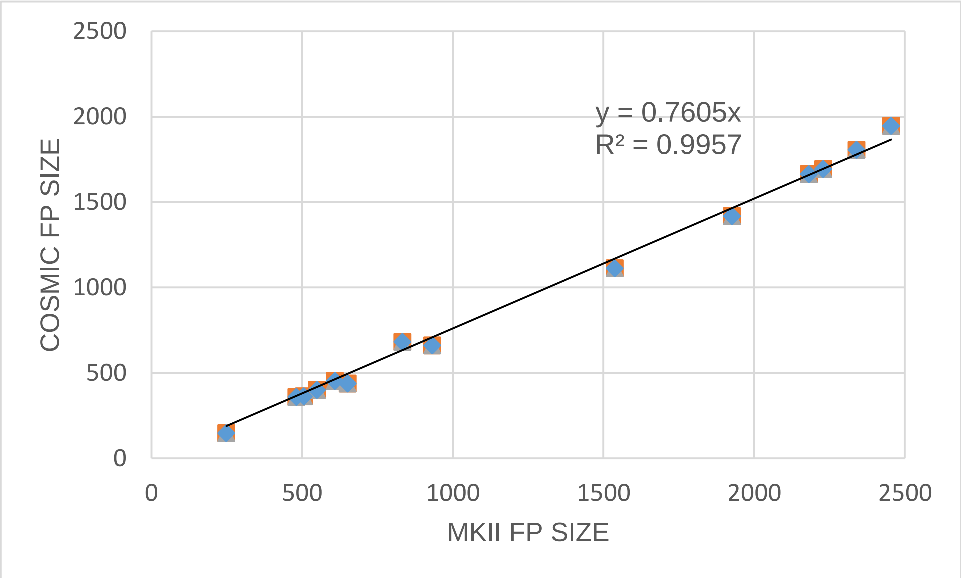


Figure 18 - Relationship between COSMIC and MkII sizes – Dasgupta, Gencel, & Symons [26]

Note: as the minimum size of a MkII Logical Transaction is 2.5 MkII FP and that of a COSMIC functional process is 2 CFP, the OLS-fitted straight lines for all three datasets were constrained to pass through the origin at (0,0).

A main conclusion from all three datasets is that MkII FP and COSMIC sizes correlate very well. The R-squared values of all three datasets (from different organizations and software domains) are all at least 0.99.

APPENDIX B – DETAILED RESULTS OF STATISTICAL CONVERSION STUDIES

(In the tables below, R^2 is the determination coefficient. and expresses the proportion of variation in the COSMIC size (in CFP) that is explained by a change in the FP size (either IFPUG or Nesma).

Note that some of the studies reported below had less than 10 data points which means their OLS fitted lines are not very statistically significant. The results have been included because the data from all these studies has been included in the analysis of the total of 135 projects (see Appendix A, section A.1.1) and used in the general discussions in this Guideline.

Table 1 - Conversion Formulas (from IFPUG /Nesma FP to COSMIC CFP) reported in the Literature

Dataset	No. of projects	Version of FSM method	Range of Sizes (CFP: IFPUG/Nesma FP)	Convertibility Study	Statistical analysis technique used	Conversion Formula (CF)	R^2
DS1. Fetcke 1999 Dataset (Warehouse management) [7]	5	COSMIC 2.0 IFPUG 4.1	IFPUG FP: 31 – 77 CFP: 29- 81	Vogelezang & Lesterhuis (2003) [9]	OLS	CM1: CFP = 1.1 x FP – 6.2	0.99
DS2. Vogelezang & Lesterhuis 2003 Dataset (Rabobank) [10]	11	COSMIC 2.2 Nesma 2.0	Nesma FP: 39 – 1424 CFP: 23 - 1662	Vogelezang & Lesterhuis (2003) [10]	OLS	CM2: CFP = 1.2 x FP - 87	0.99
				Lavazza [22]	OLS (outliers removed)	CM3: CFP = 0.78 x FP - 4	0.92
				Lavazza [22]	log-log regression and LMS	CM4: CFP = 0.81 x FP - 12	0.67
				Abran, Desharnais, Azziz (2005) [11]	Piecewise linear regression	CM5: CFP= 0.75 x FP – 2.6 (FP <= 200) CFP= 1.2 x FP – 108 (FP >200)	0.85 0.99
DS3. Abran et al. 2005 Dataset (government projects) [11]	6	COSMIC 2.2 IFPUG 4.1	IFPUG FP: 103 – 1146 CFP: 75 - 934	Abran, Desharnais, Azziz (2005) [11]	OLS	CM6: CFP = 0.84 x FP + 18	0.91

DS4. Desharnais and Abran Dataset 2006 (MIS projects) [12]	14	COSMIC 2.2 IFPUG 4.1	IFPUG FP: 111 - 647 CFP: 88 – 579	Desharnais & Abran (2006) [12]	OLS	CM7: CFP = 1.0 x FP - 3	0.93
				Lavazza [22]	OLS (outliers removed)	CM8: CFP = 0.97 x FP - 6	0.96
				Lavazza [22]	log-log regression and LMS	CM9: CFP = 0.98 x FP - 5	0.84
				Lavazza [23]	Piecewise linear regression (outliers removed)	CM10: CFP = 1.0 x FP - 11 (FP <= 317) CFP = 1.2 x FP - 79 (FP > 317)	0-96 0.85
DS5. Van Heeringen Sogeti Dataset (banking, insurance, government projects) [15]	26	COSMIC 2.2 Nesma 2.0	Nesma FP: 61 – 1622 CFP: 66 - 1864	Van Heeringen (2007) [15]	OLS	CM11: CFP = 1.22 x FP - 64	0.97
				Lavazza [22]	OLS (outliers removed)	CM12: CFP = 1.05 x FP - 18	0.94
				Lavazza [22]	log-log regression and LMS	CM13: CFP = 1.09 x FP - 34	0.81
				Lavazza & Morasca [23]	Piecewise linear regression (outliers removed)	CM14: CFP = 1.06 x FP - 18 (FP <= 606) CFP = 1.67 x FP - 389 (FP > 606)	0.91 0.90

DS6.Cuadrado-Gallaego et al. Dataset 2007 [12]	33	COSMIC 2.2 IFPUG 4.1	IFPUG FP: 78 - 462 CFP: 65 - 313	Lavazza [22]	OLS (outliers removed)	N/A	N/A
				Lavazza [22]	log-log regression and LMS	CM15: CFP = 0.54 x FP +15	0.50
				Lavazza & Morasca [23]	Piecewise linear regression (outliers removed)	CM16: CFP = 0.56 x FP + 17 (FP <= 279) CFP = 1.15 x FP - 148 (FP > 279)	0.93 0.54
DS7. Cuadrado-Gallaego et al. jj06 [15]	21	COSMIC 2.2 IFPUG 4.1	IFPUG FP: 109 - 438 CFP: 65 - 396	Cuadrado-Gallaego et al. [16]	OLS	CM17: CFP= 0.83 x FP – 36.6	0.7
DS8. Cuadrado-Gallaego et al. jj07 [17]	14	COSMIC 2.2 IFPUG 4.1	IFPUG FP:66 - 351 CFP: 46 - 301	Cuadrado-Gallaego et al. [17]	OLS	CM18: CFP=0.85 x FP + 0.19	0.86
DS7-DS8 Merged Cuadrado-Gallaego et al. [20]	35	COSMIC 2.2 IFPUG 4.1	IFPUG FP:66 – 438 CFP: 46 - 396	Cuadrado-Gallaego et al. [20]	OLS	CM19: CFP= 0.73 x FP – 4.45	0.9

Table 2 - Conversion Formulas (from IFPUG Transactional Size (FP-TX) to COSMIC total CFP Size)

Dataset (DS)	No. of projects	Version of FSM methods	Range of Sizes (CFP: IFPUG FP)	Study	Statistical analysis technique used	Conversion Formula (CM)	R^2
DS3. Abran et al 2005 Dataset (government projects) [11]	6	COSMIC 2.2 IFPUG 4.1	IFPUG FP: 103 - 1146 CFP: 75 - 934	Abran et al.[11]	OLS	CM20: CFP=1.35 x (FP-TX) + 5.5	0.98
DS4. Desharnais et al. Dataset 2006 (MIS projects) [12]	14	COSMIC 2.2 IFPUG 4.1	IFPUG FP: 111 - 647 CFP: 88 - 579	Desharnais et al. [12]	OLS	CM21: CFP=1.36 x (FP-TX)	0.98

Table 3 - Conversion Formulas (from MkII FP total size to COSMIC total CFP Size)

Dataset (DS)	No. of projects	Version of FSM methods	Range of Sizes (CFP: MkII FP)	Study	Statistical analysis technique used	Conversion Formula (CM)	R^2
Fetke1999 Dataset (Warehouse management) [7]	5	MkII v1.3.1 COSMIC 2.0	MkII: 24 – 73 COSMIC 29 - 81	This Guideline	OLS	CFP = 1.107 x MkII FP or CFP = 1.043 x MkII FP + 3.259	0.988 or 0.992
Gencel, Demirors [18]	4 (Control Systems)	MkII 1.3.1 COSMIC 4.0	MkII: 348-4380 COSMIC: 251-3524	Gencel, Demirors [18]	OLS	CFP = 0.801 x MkII FP (Computed for this Guideline)	0.999
Dasgupta (SITA) [26]	13 Information Systems	MkII 1.3.1 COSMIC 4.0	MkII: 249 – 2988 COSMIC: 148 - 1947	Dasgupta, Gencel, Symons 2015 [26]	OLS	CFP = 0.7605 x MkII FP	0.996

APPENDIX C – FPA TO COSMIC MIGRATION IN ORDER TO ESTIMATE WEB APPLICATION DEVELOPMENT EFFORT

This note on recent research was provided by Federica Sarro.(see Acknowledgements).

This Guideline has focused on how to build size conversion equations in order to migrate from FPA to COSMIC. In most cases, the main use of software sizes will be as input for effort estimation models. The research described below concerns a method of adapting existing FP-based estimating formulae by substituting CFP sizes without the need to first establish a FP/CFP size correlation.

Di Martino et al. [31] were the first to assess the effectiveness of the COSMIC sizes obtained using both 'internal' (measured locally) and 'external' (taken from the literature) conversion equations for effort estimation purposes. In particular, they investigated whether the size in terms of COSMIC is more informative than the size in terms of FPs when the aim is to predict the development effort of Web applications.

The subject of their empirical study was a set of 25 Web applications provided by an Italian medium-sized software company, whose core business is the development of enterprise information systems, mainly for local and central government. The set includes e-government, e- banking, Web portals, and Intranet applications. All the projects were developed with SUN J2EE or Microsoft .NET technologies. Oracle was the most commonly adopted DBMS, but also SQL Server, Access and MySQL were employed in some of these projects. This was hence a very heterogeneous dataset.

The results revealed that COSMIC outperformed Function Points by providing significantly better estimation of development effort. This applied when using two different estimation techniques (Simple Linear Regression and Case Based Reasoning). Thus, the authors concluded that the software company involved in the study should migrate from FPA to COSMIC to improve the quality of its effort estimations.

In the same study the authors also addressed the question on how to manage such a transition [31] by investigating the use of a two-step estimation process (named 2SEP) that first exploits historical FPA data and a conversion equation to estimate COSMIC sizes and then uses them to predict development effort, until enough COSMIC data has been collected. The results also highlighted that internal conversion equations provided more accurate effort predictions than those provided by external conversion equations [32]. However, it was observed that employing internal conversion equations requires re-measuring a sample of previously-developed applications, which means further effort/cost by managers of the company.

In order to reduce this cost, Corazza et al. [33] proposed an alternative approach based on transfer learning, that is able to estimate the effort of new projects by adjusting the information gathered about past projects over time. This approach, named CFP-TL, builds adaptive regression models based on the combined use of data on past projects sized with the previous measurement method (source domain) and the data about new incoming projects sized with the new measurement method (target domain), i.e., it adaptively exploits the knowledge acquired from the source domain until there are enough points in the target domain.

The approach was empirically validated on the same set of Web Applications used in previous work [33]. *The results revealed that during the migration from Function Points to COSMIC the company involved in the study could dismiss the FP based model in favor of the CFP-TL effort estimation model only after two projects had been measured in COSMIC*.

The empirical evidence that COSMIC is more effective than FPA for tasks such as software development effort estimation could motivate those organizations that usually employ FPA to migrate to COSMIC. However, although the results discussed in this section are very promising they might not apply to other contexts. To this end more studies based on industrial experiences should be conducted.

APPENDIX D – ABBREVIATIONS AND GLOSSARY OF TERMS

Abbreviations used in this Guideline

BFC = Base Functional Component

CFP = COSMIC Function Point

DET's = Data Element Types

E = Entry

EI = External Input

EIF = External Interface File

EO = External Output

EQ = External Query

ER's = Entity References

FP = Function Points (as per IFPUG/Nesma)

FP-TX = the FP size of IFPUG/Nesma 'transaction types' (or Elementary Processes)

FSM = Functional Size Measurement

FTR's = File Types Referenced

FUR = Functional User Requirements

ILF = Internal Logical File

IPO = Input/Process/Output

LF = Logical File (ILF's + EIF's)

OLS = Ordinary Least Squares

R = Read

RET's = Record Element Types

W – Write

X = Exit

Glossary of Terms specific to this Guideline

This Glossary contains terms that are specific to this Guideline.

The Measurement Manual [1] contains the main Glossary of terms of the COSMIC method, For terms of the IFPUG, MkII and Nesma methods, see their respective manuals.

In the definitions given below:

- terms are shown in **bold**.
- terms that are defined elsewhere in this Glossary are under-lined, for ease of cross-reference.

Homogeneous dataset. A set of software items whose functional requirements are similar according to a given objective test and therefore whose functional sizes measured by one FSM method may be converted to functional sizes according to another FSM method by a common conversion method. An example of an objective test is that measurements indicate a similar (i.e. more or less equal) IPO Profile for all software items in the dataset.

Input/Process/Output model. A model in which each functional process is divided into an Input, a Process, and an Output component.

IPO Profile. The percentages of the contributions to total size of the Input, Process, and Output components of the functional processes measured.

Log-log representation. A two-dimensional representation of numerical data that uses logarithmic scales on both the horizontal and vertical axes.

Mean Magnitude Relative Error (MMRE). The mean of a number of Magnitude Relative Error (MRE) estimates, where for example a MRE of estimated effort is defined as: (| actual effort – estimated effort |) / actual effort.

Outlier. An observation (or 'data point') that is substantially different from (or 'out of line with') the other observations in the same set.

Regression analysis (simple linear). Fits a straight line through a set of N points in such a way that the vertical distances between the points of the data set and the fitted line (the 'residuals') are minimized. The underlying technique minimizes the sum of the squared residuals; it is therefore also called the 'least squares method' (or 'Ordinary Least Squares' (OLS) method.

R-squared (the coefficient of correlation or determination), denoted R^2 or r^2 and pronounced '**R squared'**. A number that indicates how well data fit a statistical model – sometimes simply a linear-curve or a non-linear curve such as an exponential or a power curve. An R^2 of 1 indicates that the regression line perfectly fits the data, while an R^2 of 0 indicates that the line does not fit the data at all.

Software item. 'A whole' business application or an enhancement to an existing application that consists of the addition of whole chunks of functionality

Statistically significant. If the probability that an observed effect would have occurred just by chance is less than a chosen level (often e.g. 5%), then it is concluded that the observed effect actually reflects the characteristics of the population, i.e. the observed effect is 'statistically significant'. The 'p-value' is usually adopted to evaluate statistical significance.

APPENDIX E - COSMIC CHANGE REQUEST AND COMMENT PROCEDURE

The COSMIC Measurement Practices Committee (MPC) is very eager to receive feedback, comments and, if needed, Change Requests for this guideline. This Appendix sets out how to communicate with the COSMIC MPC.

All communications to the COSMIC MPC should be sent by e-mail to the following address:

mpc-chair@cosmic-sizing.org

Informal general feedback and comments

Informal comments and/or feedback concerning the guideline, such as any difficulties of understanding or applying the COSMIC method, suggestions for general improvement, etc. should be sent by e-mail to the above address. Messages will be logged and will generally be acknowledged within two weeks of receipt. The MPC cannot guarantee to action such general comments.

Formal change requests

Where the reader of the guideline believes there is a defect in the text, a need for clarification, or that some text needs enhancing, a formal Change Request ('CR') may be submitted. Formal CR's will be logged and acknowledged within two weeks of receipt. Each CR will then be allocated a serial number and it will be circulated to members of the COSMIC MPC, a world-wide group of experts in the COSMIC method. Their normal review cycle takes a minimum of one month and may take longer if the CR proves difficult to resolve. The outcome of the review may be that the CR will be accepted, or rejected, or 'held pending further discussion' (in the latter case, for example if there is a dependency on another CR), and the outcome will be communicated back to the Submitter as soon as practicable.

A formal CR will be accepted only if it is documented with all the following information.

- Name, position and organization of the person submitting the CR.
- Contact details for the person submitting the CR.
- Date of submission.
- General statement of the purpose of the CR (e.g. 'need to improve text…').
- Actual text that needs changing, replacing or deleting (or clear reference thereto).
- Proposed additional or replacement text.
- Full explanation of why the change is necessary.

A form for submitting a CR is available from the www.cosmic-sizing.org site. The decision of the COSMIC MPC on the outcome of a CR review and, if accepted, on which version the CR will be applied to, is final.

Questions on the application of the COSMIC method

The COSMIC MPC regrets that it is unable to answer questions related to the use or application of the COSMIC method.

You can use the forum on cosmic-sizing.org/forums to post your questions and receive answers from our world-wide community. The quality of any answers will depend on the knowledge and experience of the community member that writes the answer; the MPC cannot guarantee the correctness. Commercial organizations exist that can provide training and consultancy or tool support for the method. Please consult the 'cosmic-sizing' for further detail.