2013 Joint Conference of the 23nd International Workshop on Software Measurement (IWSM) and the Eighth International Conference on Software Process and Product Measurement (Mensura)

Approximate COSMIC Functional Size

Guideline for approximate COSMIC Functional Size Measurement

Frank Vogelezang (editor)
Ordina, Pricing Office
The Netherlands
frank.vogelezang@ordina.nl

Charles Symons
COSMIC, Measurement Practices Committee
United Kingdom
mpc-chair@cosmicon.com

Arlan Lesterhuis
COSMIC, Measurement Practices Committee
The Netherlands
arlanlesterhuis@cosmicon.com

Roberto Meli
DPO
Italy
roberto.meli@dpo.it

Maya Daneva
University of Twente, Information Systems Group
The Netherlands
m.daneva@utwente.nl

***Abstract*—The COSMIC method provides a standardized way of measuring the functional size of software from the functional domains commonly referred to as 'business application' or 'Management Information Systems' (MIS) and 'real-time' software, and hybrids of these.**

In practice it is often sufficient to measure a functional size approximately. Typical situations where such a need arises are early in the life of a project, before the functional user requirements ('FUR') have been specified down to the level of detail where the precise size measurement is possible or when a measurement is needed, but there is insufficient time or no need to measure the required size using the standard method.

The guideline describes the current state of the art with regard to approximate COSMIC functional size measurement. All proposed COSMIC approximation methods rely on determining some average of the size(s) and/or number(s) of functional processes. The fact that the size of a single functional process has no upper finite limit is probably the reason why multiple COSMIC approximation methods have been developed for different types of software. Therefore the guideline describes a number of approximation methods with their pros and cons, their recommended area of application and their validity, rather than document a single COSMIC approximation method.

Keywords—COSMIC; approximation; scaling; classification; accuracy, software measurement

I. INTRODUCTION

The COSMIC functional size measurement method requires that measurements of the functional user requirements ('FUR') of a piece of software must be made at a standard level of granularity, known as the 'functional process level of granularity'. The rules for this standard level of granularity, as defined in the Measurement Manual, are as follows:

A. *Functional size measurement should be made at the functional process level of granularity*

B. *where a functional size measurement is needed of some FUR that have not yet evolved to the level where all the functional processes have been identified and all the details of their data movements have been defined, measurements should be made of the functionality that has been defined, and then scaled to the level of granularity of functional processes.*

The guideline [1] describes several methods to implement rule (B). This rule states that when the full details of the FUR are not (yet) available, we should find a way of approximating the size that we would measure when the full details of the FUR become available, based on what we know about the FUR on a higher level of granularity.

In practice there are two main reasons to measure a functional size approximately. First, when a measurement is needed early in the life of a project, for example as input to a project estimating process, before the FUR have been specified down to the level of detail where the precise size measurement is possible. This is known as 'early sizing'. Second, when a measurement is needed but there is insufficient time or it would be uneconomic to measure the required size using the standard method and an approximate size is acceptable – if it can be measured faster than is possible

978-0-7695-5078-7/13 $26.00 © 2013 IEEE
DOI 10.1109/IWSM-Mensura.2013.14

with the standard method. This is known as ‘rapid sizing’ which can be valuable when a very large piece of software or, say, a whole software portfolio needs to be measured.

In the rest of this paper, we first summarize the COSMIC functional size measurement method for the purpose of providing clarity and context to those readers who are less familiar with it. We then present the General Principles of Approximating Size and the five approximation methods along with information about their scope of applicability and the assumptions behind them. We conclude with some implications that our analysis has for practitioners and researchers.

II. The COSMIC method

The COSMIC method [2] has been developed by the Common Software Measurement International Consortium (COSMIC) and is now an international standard (ISO/IEC 19761). The method focuses on the “user view” of functional requirements, and is applicable throughout the development life cycle, from the requirements phase right through to the implementation and maintenance phases. The process of measuring software functional size using COSMIC implies that the software functional processes and their triggering events be identified. In COSMIC, the basic functional components are data movements. The unit of measure is a COSMIC Function Point (CFP) which refers to a movement of the data attributes belonging to a single data group. Data movements can be of four types: Entry, Exit, Read or Write. The functional process is an elementary component of a set of functional user requirements triggered by an event via an actor – the ‘functional user’. The triggering event is an event occurring outside the boundary of the measured software that causes a functional user to initiate a functional process. A functional process comprises at least two data movements: an Entry plus at least either an Exit or a Write. There is no upper limit to the number of data movements of a functional process.

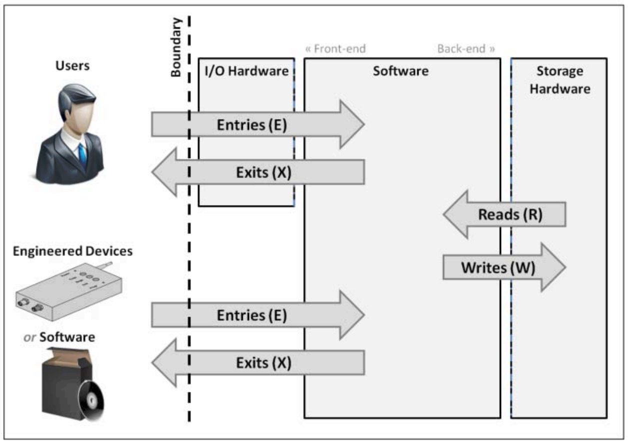

Fig. 1. Generic flow of data attributes through software from a functional perspective

An Entry moves a data group, which is a set of data attributes, from a user across the boundary into the functional process, while an Exit moves a data group from a functional process across the boundary to the user requiring it. A Write moves a data group lying inside the functional process to persistent storage, and a Read moves a data group from persistent storage to the functional process. Figure 1 illustrates the generic flow of data attributes through software from a functional perspective.

III. General principles of approximate sizing

The approximation methods described in the Guideline rely on one common principle, namely that the only precisely-defined level of granularity of functional user requirements is the functional process level of granularity. The methods are generally based on two approximation principles or a combination of these:

Scaling: a linear transformation to enlarge the size of an object to the desired dimension. Count objects at a higher level of granularity than the functional process and multiply the count by a scaling factor to represent their size at the functional process level of granularity.

Classification: classify an object at a given level of granularity and assign a size to it that represents the size at the functional process level of granularity for that class of objects.

In practice, a size measurement may be needed when requirements exist at varying levels of detail. For example, an early size approximation may be needed when some requirements exist only at a global level whilst other requirements have been elaborated in more detail. In this situation a mix of approximate sizing methods may be used for the requirements at each of the different levels of granularity.

Approximate COSMIC sizing methods described here may be used in projects that must develop new functionality or that enhance existing software with wholly new functionality, or for sizing existing software. The numbers we present in the description of the methods may not be applicable for sizing the functionality to be dealt with in projects that must make changes and deletions to existing software, without proper calibration.

IV. Approximation methods

The guideline summarizes five documented approximation methods for the COSMIC method:

- Average functional process approximation
- Fixed size classification approximation
- Equal size bands approximation
- Average use case approximation
- Early & Quick COSMIC approximation

For each method the origin and principles are briefly described along with the reported use and claimed applicability domain. For each method we have analyzed the strengths and weaknesses and based on that analysis we give guidance on the recommended area of application.

Where relevant we describe research areas related to the method. This can be research that is already published, that is under way or proposed research.

In the next sections we describe the approximation methods in more detail.

V. AVERAGE FUNCTIONAL PROCESS APPROXIMATION

The average functional process approximation was first introduced in the COSMIC-FFP Measurement Manual version 2.2. [2]. With this method, all measurements are made at the functional process level of granularity. This simplest process for obtaining an approximate size of a new piece of software is therefore as follows. Given a new piece of software, execute:

A. Sampling and calculation of the size of an average functional process

1. Identify a sample of FUR with characteristics similar to the new FUR
2. Identify the functional processes of these other FUR
3. Measure the sizes of the functional processes of these other FUR accurately using the standard COSMIC method
4. Determine the average size, in CFP, of the functional processes of these other FUR (e.g. average size = 8 CFP). This the scaling factor for this method.

B. Approximation using the calculated average of the sample

1. Identify and count all the functional processes of the new FUR (e.g. = 40)
2. Based on the sample, the approximate size of the new FUR is estimated to be (number of functional processes x average size from sample) = 40 x 8 = 320 CFP

In organizations that have established a COSMIC measurement practice this method is used to produce a first ball-park approximation of the size.

In 2005 Vogelezang [4] reported that in different industry sectors different sizes were measured for an average functional process. This suggests that the use of this approximation method should always be calibrated locally. For example, the average size for a functional process in avionics software is much larger than the example of 8 CFP reported for MIS software. The most probable cause for the larger functional processes is that avionics software is designed to achieve the isolation and protection from faults and failures of (multiple) components [5].

This approximation is valid as long as there is sufficient reason to assume that the sample on which the size of the average functional process is calculated is representative for the software of which the functional size is approximated.

VI. FIXED SIZE CLASSIFICATION APPROXIMATION

The average functional process approximation was first introduced in the 'Advanced and Related Topics' document that was separated from the Measurement Manual in version 3.0 of the COSMIC method [6].

The method depends on identifying a typical size classification of the functional processes in a given piece of software. A corresponding size is then assigned to each functional process. This approximation was based on experience with a large business organization in the Netherlands and has been extensively used there.

A requirements specification must be analyzed to identify the functional processes and to classify each of them according to their size in one of three or more size classes called, for instance Small, Medium and Large. To force the Measurer to make a deliberate choice of size, the step size between the classes is taken to be fairly wide, at 5 CFP. The Small class was thus assigned 5 CFP, the Medium class 10 CFP and the Large class 15 CFP, which are the scaling factors for this method.

Although the method was used successfully within the large Dutch business organization, there is no public information on the accuracy of this approximation.

The numbers used in the text above suggest that this method is only applicable to software with small, relatively simple functional processes of limited size range. This approximation is valid as long as there is sufficient reason to assume that the assigned size classification is representative for the software of which the functional size is approximated. For software with larger functional processes in a more extended size range other classification numbers will apply with a different accuracy. If this approximation would have been used for the avionics software in the next section, the step-size would have been 8, instead of 5.

VII. EQUAL SIZE BANDS APPROXIMATION

The average functional process approximation (discussed in Section V) was first introduced in the COSMIC-FFP Measurement Manual version 2.2. [2].

This approximation method can be refined in order to improve the accuracy of the counting results if sufficient size data are available for an accurate calibration. In the 'Equal Size Bands' method, the functional processes are classified into a small number of size bands. The boundaries of the bands are chosen in the calibration process so that the total size of all the functional processes in each band is the same for each band. So if, for example, the choice is to have three bands, then the total size of all the functional processes in each band will contribute 33% to the total size of the software being measured.

Vogelezang and Prins [7] have reported on using this method for early sizing, having carried out a calibration using measurements on 37 business application developments, each of total size greater than 100 CFP. In this empirical study [7] it was decided to use four size bands. The average sizes of each band (the scaling factors for this approach) when the 2,427 functional processes of the 37 applications were distributed over the four bands evenly are:

TABLE I. EQUAL SIZE BANDS FOR BUSINESS APPLICATIONS

Band	Average FP size	% of size	% of #FP
Small	4.8 CFP	25%	40%
Medium	7.7 CFP	25%	26%
Large	10.7 CFP	25%	19%
Very large	16.4 CFP	25%	15%

This same 'Equal Size Bands' method was used to calibrate one component of a major real-time avionics system [7] (of total size 10,875 CFP), giving the following results for the four bands:

TABLE II. EQUAL SIZE BANDS FOR AVIONICS

Band	Average FP size	% of size	% of #FP
Small	5.5 CFP	25%	49%
Medium	10.8 CFP	25%	26%
Large	18.1 CFP	25%	16%
Very large	38.8 CFP	25%	7%

Note the similarity between the *numbers* of functional processes in each of the four bands in spite of the totally different types of software. This might indicate that there is a general distribution pattern over the size bands. The samples used for this research are too small to draw firm conclusions on this point. However, the average size of the functional processes in each band is quite different, especially for the large and very large bands. This emphasizes the need for local size calibration of scaling factors.

To size a new piece of software the functional processes of the new piece are identified, and then they are classified as 'Small', 'Medium', 'Large' or 'Very Large'. In the next step, the average sizes of each band (such as listed above but preferably calibrated locally, because a Measurer may not have knowledge about the data sets used to produce the classification ranges) are then used to multiply the number of functional processes of the new piece of software, in each band respectively to get the total estimated approximate size.

The advantage of this method is that at the end of a new sizing, the Measurer can check if the contribution to the total size of the functional processes in each size band is close to the 25% assumed by this 'Equal Size Band' method. If so, the used bands will have been suitable for this new measurement. If not, the Measurer should consider whether the used bands are suited for this approximation.

The accuracy of any calibration of classification sizes for this method is critically important for accurate sizing since functional processes typically exhibit a skewed size distribution, as illustrated by both sets of data given above. In other words, software systems typically have many functional processes of small size and few of larger sizes. It follows that relatively more attention must be paid to correct identification (and accurate average sizing for scaling purposes) of the few 'large' and the even fewer 'very large' functional processes to get an accurate total size.

This method is recommended for approximate sizing of software that has a skewed distribution of the size of functional processes. For the business application software as reported in [7] this method has little added value over the average functional process method (Section V) or the fixed size classification method (Section VI). For the avionics software there is added value to use this method. Additional research is needed to find predictors for a skewed distribution. The presence of functional processes that exceed the size of an average functional process in that domain (say two- or threefold) is currently the best indicator.

In 2012 the results reported by Vogelezang and Prins [7] were tested using a fuzzy logic model to estimate the functional size of the C-registration system Case Study by Valdes Souto and Abran [8]. This test showed that the equal size bands approximation was a better approximation than the fuzzy logic model used in the experiment.

VIII. AVERAGE USE CASE APPROXIMATION

The average functional process approximation was first introduced in the 'Advanced and Related Topics document that was separated from the Measurement Manual in version 3.0 of the COSMIC method [6]. The principle of the approximation is similar to the average functional process approximation from section II, but at the level of granularity of the use case, which is typically higher than that of the functional process.

Local calibration might determine that a (locally-defined) use case comprises for example, on average, 3.5 functional processes, each of average size 8 CFP (as in section V). This approximation thus makes use of an average size of a functional process and a scaling factor for the average number of functional processes per use case. Hence the average size of a use case according to this local definition, is 3.5 x 8 = 28 CFP per use case.

For a new project with 12 use cases, the software size would be 12 x 28 = 236 CFP.

A recent study by Gencel [9], however, of practices in a very large software house showed that different parts of the software house had quite different ideas on what is a Use Case. In one part there was a fairly consistent ratio of functional processes per Use Case. In another part, this ratio varied widely. Obviously this method requires consistent practice on interpreting what is a Use Case.

Thus, with this calibration, identifying the number of use cases early in a development project's life will provide a basis for making a preliminary estimate of software size in units of CFP. The uncertainty on this approximate size will be greater than that with the method discussed in e.g. Section V. This is because the scale factor 28 is the product of two other factors (8 and 3.5) which are themselves estimated. The result might therefore be better expressed as, for example, 240 plus or minus x%, where the ‘x%’ has been obtained by appropriate analysis).

This approximation is valid as long as there is sufficient reason to assume that the assigned size classification of an average use case is representative for the software of which the functional size is approximated.

IX. Early & Quick COSMIC approximation

The Early & Quick COSMIC approximation method is an adaptation of the Early & Quick Function Points method developed by the Italian company DPO that was originally proposed in 1997 [10] for IFPUG Function Point Analysis, in order to help practitioners in sizing software systems starting from non-detailed information, typically with reduced sizing effort and time with respect to the standard measurements. After many years of extensive usage in Italian companies and public administrations it was published as a publicly available method [11]. In 2000 a first version of the COSMIC approximation method was presented [12]. In 2004, the Early & Quick method was improved both in the IFPUG and, experimentally, to the COSMIC measurement method [13] taking advantage of enhancement opportunities derived from local or global measurements data sets, like the ISBSG benchmarking data base and others.

The Early & Quick COSMIC approximation method combines a number of techniques in order to estimate a software system functional size. It makes use of both analogical comparison and structured classification of functions. Moreover, it permits the use of different levels of granularity for different branches of the system (multilevel approach). The overall accuracy in the size approximation (which is a 3-point estimate of a minimum, most likely, and maximum size) depends upon the weighted sum of the individual components’ accuracy. Finally, the method provides its estimates through tables of values which DPO claims have been ‘statistically and analytically validated’. The method is based on the following principles:

- Classification by analogy
- Structured aggregation
- Multilevel approach
- Use of a reference table of sizes at each level

The Early & Quick COSMIC approximation method is based on the capability of the Measurer to classify a part of the FUR as belonging to a particular functional category. Each part of the FUR is to be classified, in order of increasing magnitude and number of composing elements at one of four levels, as a Functional Process, Typical Process, General Process, or Macro-Process. The reference table then allows the estimator to assign a CFP average value for that part of the FUR (this is applied for each identified level separately). Note that each level is built up on the basis of the lower one except for the Typical Process, which is off-line from the hierarchical structure outlined. The Typical Process is just the set of the four frequently used operations that arise in the business application domain, which are: Create, Retrieve, Update and Delete (CRUD) information in a relevant data group.

Assigning each part of the FUR to a specific category higher than the Functional Process level is quite subjective. The detailed description of the method gives guidance on assigning the proper category [11]. When applied at the General Process level, however, this subjectivity disappears. The precision of the method is thus strongly dependent on the training and capability of the practitioners who use it to understand the categories at the higher levels of granularity.

This method is most suited when (a part of) the FUR is not detailed enough to identify functional processes.

X. Conclusions and Implications for Practitioners and Researchers

Methods to approximate sizing can be made to work and are valuable for use early in a new software project‘s life and/or can save time and effort compared with sizing accurately using the standard COSMIC measurement method.

But approximate sizing methods need to be used with care. When there is a need for approximate sizing, the Measurer should:

- choose a method which is optimal for the purpose of the measurement, given the availability of data for the calibration, the time available for the measurement and the accuracy required of the approximate size;
- calibrate the method using accurately-measured local data on software that is comparable to that for which the approximate sizes must be measured;
- pay particular attention to identifying the large functional processes and to determining good average sizes and/or scaling factors for them, since they are few in number, but make a large contribution to the total size;
- consider whether an allowance should be made for ‘scope creep’ when publishing an approximate size and also for non-functional requirements which (partly) may result in functional processes or additional data movements (i.e. more CFP).
- estimate and report the plus or minus uncertainty on the approximate size, mentioning any correction that has been made for scope creep; estimating the uncertainty on an approximate size is especially important in contractual situations.

The guideline not only describes the different methods to approximate sizing, but also gives guidance on how to deal with a number of locally defined aspects [1]:

- Establishing a locally-defined approach
- Determining the accuracy of approximations
- Approximate sizing of changes
- Approximate sizing and scope creep

These aspects have not been investigated in full detail yet, so this paper can only announce a more detailed discussion in the guideline.

The work on the guideline has some implications not only for practitioners but also for researchers.

First, while presenting the five approximation methods, we acknowledge that research efforts are under way to collect and evaluate empirical evidence about the strengths of these methods in various contexts. An open question for industry-relevant research is how the methods compare and which method is better in what context.

Second, not all methods have been sufficiently investigated for accuracy. More data collected from a variety of organizations would be helpful to reason more convincingly about how our experiences of using the methods might be generalized.

Third, each method makes assumptions about the context in which it is supposed to bring results. In this paper, we documented these assumptions and acknowledged that they may not be realistic in all cases. We therefore consider that empirical studies on the assumptions and their proper justification would be the next logical research step to better understand the full potential of the presented methods. This research will help identify alternative methods to approximation in those cases where the assumptions behind the methods presented here, are not present.

Fourth, the methods described are quite often applied on FUR that have not reached their final maturity. Little is known how to check their level of granularity, scope coverage and proper decomposition. Statistical analysis of these aspects will help to improve the estimation accuracy of these approximate functional sizing methods.

References

[1] F.W. Vogelezang (ed.), C.R. Symons, A. Lesterhuis, M. Daneva, R. Meli and G. Ungerer, "Guideline for approximate COSMIC size measurement," *to be published on www.cosmicon.com*

[2] C.R. Symons and A. Lesterhuis, The COSMIC Functional Size Measurement Method Version 3.0.1 Measurement Manual, the COSMIC implementation guide for ISO/IEC 19761:2003, May 2009

[3] A. Abran, J-M. Desharnais, S. Oligny, D. Saint-Pierre and C.R. Symons, COSMIC-FFP Measurement Manual version 2.2, the COSMIC implementation guide for ISO/IEC 19761:2003

[4] F.W. Vogelezang, Early estimating using COSMIC-FFP, Proceedings of the 2nd Software Measurement European Forum, March 16-18, 2005, Rome.

[5] L. Sha, The Complexity Challenge in Modern Avionics Software, National Workshop on Aviation Software Systems, October 4-5, 2006, Alexandria (VA).

[6] A. Lesterhuis and C.R. Symons, The COSMIC Functional Size Measurement Method version 3.0 Advanced and Related topics, December 2007.

[7] F.W. Vogelezang and T.G. Prins, 'Approximate size measurement with the COSMIC method: Factors of influence', SMEF 2007 Conference, Rome, Italy.

[8] F. Valdes Souto and A. Abran, Case Study: COSMIC Approximate Sizing Approach Without Using Historical Data, 2012 Joint Conference of the 22nd International Workshop on Software Measurement and the 7th International Conference on Software Process and Product Measurement, Assisi, Italy

[9] C. Symons, C. Gencel, From Requirements to Project Effort Estimates – Work in Progress (still?). REFSQ Conference, Essen, Germany, April 2013.

[10] R. Meli, "Early Function Points : a new estimation method for software projects", ESCOM 97, May 1997 – Berlin (Germany).

[11] T. Iorio, R. Meli and F. Perna, "Early and Quick Function Points® v3.0: Enhancements for a Publicly Available Method", Software Measurement European Forum SMEF2007, May 2007, Roma, Italy.

[12] R. Meli, A. Abran, V.T. Ho, S. Oligny, "On the applicability of COSMIC-FFP for measuring software througout its life cycle, ESCOM-SCOPE 2000, Munich, Germany.

[13] M. Conte, T. Iorio, and L. Santillo, "E&Q: An Early & Quick Approach to Functional Size Measurement Methods," Software Measurement European Forum SMEF 2004, January 1-3, 2004, Italy.