RESEARCH PAPERS OF WROCŁAW UNIVERSITY OF ECONOMICS
No. 85 2009
Advanced Information Technologies for Management – AITM 2009

Beata Czarnacka-Chrobot
Warsaw School of Economics, Warsaw, Poland
e-mail: bczarn@sgh.waw.pl

STANDARDIZATION OF SOFTWARE FUNCTIONAL SIZE MEASUREMENT METHODS

Abstract: Execution of application software development and enhancement projects encounters many problems, which may be proved by its unsatisfactory effectiveness, leading to considerable financial losses. One of the key reasons behind this status quo is lack of clear-cut measure for such projects product size, which may be regarded as one of the fundamental problems of software engineering. As a result of many years' verification of various approaches to software size measurement reliability and objectivity, this is only the concept called software Functional Size Measurement (FSM) that so far has been standardized by ISO and IEC. Set of rules for such measurement was included in the 6-part norm ISO/IEC 14143. Being compliant with this norm, five of the FSM methods have also been standardized by these organizations. In this paper the author attempts to make synthetic comparison of two most popular normalized FSM methods, dedicated especially for business software systems, namely IFPUG and COSMIC methods.

1. Introduction

In practice, execution of application software development and enhancement projects encounters many problems, which may be proved by its unsatisfactory effectiveness, leading to considerable financial losses [PCG 2008, pp. 1-2; Standish Group 2009, p. 1]. Low effectiveness of such projects is especially of significance with reference to Business Software Systems (BSS) as they are one of the most costly investments: spending on them may considerably exceed the expense of building offices occupied by companies commissioning these systems, and in extreme cases, even 50-storey skyscraper, roofed stadium, or cruising ship with a displacement of 70,000 tons [Jones 1999, p. 3]. One of the key reasons behind this *status quo* is lack of clear-cut measure for such projects product size, meanwhile the size of their product determines effort, cost and time frame for their execution. This situation manifests itself in the difference in costs spent by various organisations on very similar applications that may be even 15 fold [State Government of Victoria 2000, p. 1]. Hence lack of such measure may be regarded as one of the fundamental problems of software engineering.

As a result of many years' reliability and objectivity verification of various approaches towards software size measurement, having been proposed from several decades, this is only the concept called software Functional Size Measurement (FSM) that so far has been normalized by ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission). Set of rules for such measurement was included in the 6-part norm **ISO/IEC 14143** [ISO/IEC 14143 2007]. First of all, this standard specifies definition of functional size, which is understood as "size of the software derived by quantifying the Functional User Requirements" [ISO/IEC 14143 2007, Part 1, p. 2]. While Functional User Requirements (FUR) stand for the "sub-set of the User Requirements describing what the software does, in terms of tasks and services" [ISO/IEC 14143 2007, Part 1, p. 2]. The elementary unit of FUR used for measurement purposes is called Base Functional Component (BFC).[1] Furthermore, the ISO/IEC 14143 norm among others[2]:

- gives guidance on how to use FSM concept to support software development, enhancement and maintenance projects management,
- provides key definitions, characteristics and requirements for FSM,
- defines Functional Size Measurement Method (FSMM) as a specific FSM implementation defined by a set of rules, which conforms to the mandatory features of such measurement,
- allows users to choose FSMM which is best tailored to their needs,
- features description of software classes (e.g. business software systems, real-time systems, scientific software), the so-called functional domains, for which possibility of using FSMM is declared,
- recommends steps of the FSMM usage process,
- comprising the rules of selecting, among FSMM standardized by the ISO/IEC, the method that would be suitable for given functional domain.

After about 30 years of improving various software FSMM five of them (out of over 20) have been now acknowledged by the ISO/IEC as conforming to the rules laid down in the ISO/IEC 14143 norm, namely:

- Function Point Method developed by the International Function Point Users Group (IFPUG) – standardized in the **ISO/IEC 20926** norm [ISO/IEC 20926 2003],
- Function Point Method in the Mk II version proposed by the United Kingdom Software Metrics Association (UKSMA) – normalized in the **ISO/IEC 20968** standard [ISO/IEC 20968 2002],
- Function Point Method in the version developed by the Netherlands Software Metrics Association (NESMA) – standardized in the **ISO/IEC 24570** norm [ISO/IEC 24570 2005],

[1] Example: A FUR could be "Maintain Customers" which may consist of the following BFC: "Add a new customer", "Change customer details" and "Delete a customer" [ISO/IEC 14143 2007, Part 1, p. 1].

[2] More details on this norm were presented in [Czarnacka-Chrobot 2009b].

– Full Function Points Method proposed by the Common Software Measurement International Consortium (COSMIC) – normalized in the **ISO/IEC 19761** standard [ISO/IEC 19761 2003],
– Functional Size Measurement Method developed by the Finnish Software Metrics Association (FiSMA) – standardized in the **ISO/IEC 29881** norm [ISO/IEC 29881 2008].

The first three methods listed above are accepted by the ISO/IEC not in full versions, as proposed by the organizations developing them, but in part, however in the most important part as it concerns the software functional size – that is why they are called the first-generation FSMM. On the other hand the COSMIC and FiSMA methods were recognized as international standard entirely – that is why they are called the second-generation FSMM. In this paper the author attempts to make synthetic comparison of two most popular normalized FSM methods dedicated for BSS, namely IFPUG (first-generation) and COSMIC (second-generation) methods.

2. ISO/IEC 20926 standard: IFPUG Functional Size Measurement Method

In the ISO/IEC 20926 standard, normalizing FSMM developed by IFPUG, there are no functional domains constraints indicated, which means that the accepted part of this method is adequate for all software classes, although it was originally intended for the business software systems.

In IFPUG method, the unit of software functional size measurement is the so-called Unadjusted Function Point (UFP). Appropriate number of UFP is assigned to particular software functional elements: functions and data indispensable to functions execution – depending on their type and level of complexity. Number of UFP is "the measure of the functionality provided to the user by the project or application" [IFPUG 2004, Part 4, p. G-7]. In the approach proposed by IFPUG, the discussed method comprises also determining of the value adjustment factor, which is calculated using 14 predefined so-called general system characteristics. It is aimed to adjust software functional size to the environment of specific project by taking into account the impact of technical and quality requirements on the development and implementation process. However, this part of the IFPUG method has not been approved by the ISO and IEC as the assumptions adopted by these organisations do not allow for the situation where FSM depends on the requirements of technical and quality nature.

According to the normalized version of the discussed method, the process of determining the software functional size in UFP proceeds through the following basic steps:

1) defining type of calculation process,

2) identifying scope of analysis and defining measurement boundaries that determine user functions to be measured as well as data being indispensable to their execution,

3) determining number of UFP for data,

4) determining number of UFP for functions,

5) determining total number of UFP, i.e., functional size of the application measured.

Total number of UFP provides a basis for determining key attributes of software projects: effort, personnel costs, completion time and productivity of project-related activities.

According to the guidelines, calculation process may regard:

- software development project, where the subject of calculation is the size of newly-developed application being measured on the basis of FUR,
- software enhancement project, consisting in adding, changing and/or removing functions, where the subject of calculation is the size of modifications being introduced to the existing application,
- implemented/installed application, where the result of calculation reflects functionality being currently delivered by the application to a user.

Measurement boundaries define elements being of internal and external character to the measured application thus providing basis for identification of the IFPUG BFC, that is:

- data maintained within the boundaries of the measured application: the so-called Internal Logical Files (ILF),
- data used by the measured application yet maintained outside it: the so-called External Interface Files (EIF),
- transactions (functions), thanks to which the data they are processing go beyond the set boundaries: the so-called External Inputs (EI), External Outputs (EO) and External inQuiries (EQ).

Once all BFC are identified, the number of UFP is calculated – first for the data (ILF, EIF), next for transactions (EI, EO, EQ) – on the basis of complexity level, defined for each of these elements, and with the use of rules being precisely described by the IFPUG (see [IFPUG 2004]). As a result of summing up the number of UFP for data and transactions the functional size of application is delivered.

According to the IFPUG method, it is best to calculate software functional size:

- at the analysis stage, where initial estimation of the newly-developed or modified application final size is being made,
- after completing analytical activities at the development stage, where detailed estimation of the newly-developed or modified application final size is being made,
- after completing the project – in order to evaluate the actually delivered functionality versus required functionality.

3. ISO/IEC 19761 Standard: COSMIC Functional Size Measurement Method

Full Function Points Method developed by the Common Software Measurement International Consortium is entirely normalized in the ISO/IEC 19761 standard. The fact of this method being accepted in full by ISO and IEC makes a contribution to recognizing it – unlike the IFPUG approach – as a second-generation FSMM. Nonetheless the ISO/IEC 19761 norm reads that although this method is adequate for BSS, real-time systems and hybrid solutions combining the two, there are, however, functional domains constraints for software with complex mathematical algorithms or with other specialised and complex rules (e.g. expert, simulation, self-learning, weather forecasting systems) and for software processing continuous variables (e.g. computer games or musical instruments software).

The unit of software functional size in this method is the so-called COSMIC Function Point (CFP), whereas 1 CFP stands for the size of a single so-called data movement. Data movement, in the COSMIC method being an equivalent of BFC, may hold a character of:

– entry – moves a data group from a user towards software,
– exit – moves a data group from a software to the user,
– read – moves a data group from persistent storage to the software,
– write – moves a data group from software to persistent storage.

In the COSMIC method, the process of FSM proceeds in the following phases.

1. Working out measurement strategy, executed prior to actual measurement on the basis of purposes analysis and the so-called COSMIC software context model, being specific to the discussed method, which results in setting the purpose and scope of the measurement.

2. Mapping FUR for the measured software – taking into consideration the so-called COSMIC generic software model, being specific to the discussed method, as well as the measurement purpose and scope – in FUR being expressed in the form of generic software model.

3. Actual measurement based on FUR expressed in the form of generic software model, as a result delivering functional size of software measured.

The COSMIC method is an approach fully independent of technical and quality requirements. Thus the measurement within its framework is based on FUR for the specified part of software. These requirements are object of reference for the set of models, rules and processes, the use of which delivers numerical value representing functional size of particular software parts in CFP (see [COSMIC 2007a, b]).

Prior to the phase of mapping FUR for the measured software in FUR in the form of generic software model, which should be done before applying the rules and procedures of actual measurement, one should identify these requirements. FUR may be identified on the basis of information resulting from the data modelling or from the function decomposition, which allows for determining functional size of software before its implementation, or on the basis of information coming from the installed software (screens, reports, implemented data movements).

4. IFPUG Method versus COSMIC Method

All methods of software functional size measurement originate in the concept developed by A. Albrecht [Albrecht 1979] therefore there are certain similarities between them. In the case of IFPUG and COSMIC methods, they most of all include:

– common FSM concept, based on similar understanding of the measurement purpose and scope as well as definition of boundaries of the measured application;
– the rules of both methods are based on similar (yet not identical) meaning of the terms related to data. What also is convergent is the concept of data transformation as well as of users perceived as recipients of the measured software functionality;
– occurrence of two phases of measurement: identification of elements, on the basis of which the functional size is determined, and actual measurement, in which they are mapped into this numerically-expressed size. In the IFPUG method, the first of these phases is not described *explicite*, yet it assumes that the measurement is based on the FUR; data models, functions/processes models or windows, screens, forms and reports designs may also be used for this purpose. In the phase of actual measurement, the explicitly described rules of this method are employed towards these elements. In the COSMIC method, on the other hand, the measurement phase proceeds solely on the basis of FUR;
– similar way of expressing functional user requirements. In both methods, FUR are expressed by means of base functional components. In the approach developed by IFPUG there are 5 types of BFC being singled out: ILF, EIF, EI, EO, and EQ, whereas in the COSMIC method there are 4 types: entry, exit, read, and write. However, there is no simple analogy between them as in the COSMIC method data are not measured *explicite* and they are not distinguished as a type of BFC;
– both approaches, although in a different way, meet the requirements being imposed on FSM methods in the ISO/IEC 14143 norm therefore both were recognised as international standards of such measurement (the IFPUG method not in full version).

Differences between the discussed methods mostly concern the following.

– Rules of measurement. Fundamental difference in this respect is the fact that the IFPUG method includes general system characteristics, representing the influence of technical and quality requirements on functional size. This is the reason why this approach has not been approved by ISO/IEC entirely, however taking them into account in calculations is not necessary. Characteristics of this type do not exist in the COSMIC method where measurement is based solely on FUR.

– Size boundaries for processes/functions. In the IFPUG method, the size of all five BFC is arbitrarily limited thus the software size depends on their number. While in the COSMIC approach there is no upper limit for the process size as it is determined by the number of data movements. On the other hand, the size of COSMIC data movement is 1 CFP and does not depend on data to which it pertains, which is the case of processes in the IFPUG method.
– Data inclusion manner. In the IFPUG method, data are included in calculations in a twofold way: separately as internal/external logical files and as file type referenced affecting the process size. In the case of COSMIC method, data are included with each data movement of read or write type. Thus the use of IFPUG method requires constructing of data model, which in the COSMIC approach is not indispensable however proves useful. In the IFPUG method, data model also provides basis for early estimates while in the COSMIC approach this is process model that is employed for the approximation purposes.
– Benchmarking data resources. Current version of the largest repository with benchmarking data concerning software functional size measurement, that is repository of International Software Benchmarking Standards Group [ISBSG 2007], includes data in nearly 85% pertaining to the software products being measured with the use of IFPUG method while in only 3% to those measured with the use of COSMIC method.

Moreover, in the subject literature, however in most cases being supported by COSMIC, the following features of this method are pointed out as deciding on its advantage over IFPUG method.

– Broader range of application. The IFPUG method was developed in order to measure BSS, however in its current version no constraints with regard to the measurement of other functional domains were imposed by ISO and IEC. Meanwhile it is often argued that this method does not prove useful in the case of real-time systems size measurement – unlike COSMIC method [Xunmei et al. 2006]. According to the author, such conclusion goes too far both from theoretical and practical points of view although measurement of this type of software using IFPUG method undoubtedly is more complicated as compared to the COSMIC method and therefore it may be less accurate. In publications on the IFPUG method one may find not only the rules but also the examples of employing it in the measurement of real-time systems size[3].
– Compliance with object-oriented analysis and programming. In this case it is argued that if the COSMIC method was developed much later than IFPUG method it then takes into account modern techniques of FUR description and systems construction, paying attention mostly to the object-oriented approach[4]. However, this in no way proves that there is no possibility to calculate functional size using

[3] See for example: http://www.ifpug.org/publications/case.htm, Case 4 (9.06.2009).
[4] http://www.cosmicon.com/advantagecs.asp (8.06.2009).

object-oriented approach to the development based on the IFPUG method – rules of the method and practical examples do indicate such a possibility exists[5].

- Broader measurement perspective. With the use of IFPUG method, functional size is measured from the perspective of end user while with the use of COSMIC method – from the point of view of the so-called functional user that next to an end user includes also developers, who perceive other applications and devices interacting with the measured software [Xunmei et al. 2006]. Perspective limited to an end user only carries some danger of skipping in the calculations of such functionality, which is imperceptible to an end user, however on condition that it is assumed that only a user being a person can be a recipient of functionality. Meanwhile, recognising the IFPUG method as complying with the ISO/IEC 14143 standard means that definition of user it currently employs is consistent with this notion's definition included in this norm, wherein a user is understood not only as a person but also as a thing (e.g. other applications, devices, hardware) that interacts with the software being measured [ISO/IEC 14143 2007, Part 1, p. 3].
- Possibility of faster delivery of results. COSMIC method happens to be regarded as more intuitive, more concise and simpler than IFPUG approach, which should result in quicker delivery of the measurement outcome. Yet this has not been confirmed by the surveys, which indicated that there are no significant differences in the quickness of measurement made with the use of both methods [Heeringen 2007]. What is more, even authors of the COSMIC method admit that in case one needs quick measurement with low-quality user requirements, it is simpler (and therefore quicker) to employ IFPUG method – which results from the limited scope of its BFC size, which are easier to be predicted correctly[6]. In this situation using the COSMIC method would require an expert in order to obtain result on the same level of reliability, while this would increase the effort of measurement process. It is worth noting that it also applies to the possibility of employing both methods for the estimation purposes.

5. Concluding remarks

The ISO/IEC standards for the FSM methods, like the ISO/IEC 14143 norm for the FSM concept, adhere to the **ISO/IEC 15939** standard [ISO/IEC 15939 2007], which determines the general procedures for software measurement process in compliance with the **ISO/IEC 15288** norm [ISO/IEC 15288 2008] which, on the other hand, defines processes of the system's life cycle. The ISO/IEC 15939 norm identifies the process supporting defining of the appropriate set of measures pertaining to the specific informational needs as well as it determines activities and tasks being

[5] See for example: https://www.ifpug.org/publications/case.htm, Case 3 (9.06.2009).

[6] http://www.cosmicon.com/advantagecs.asp (8.06.2009).

indispensable to the effective identification, definition, selection, use and improvement of the measurement process within its general structure on the level of the organization or project. This standard also defines terms related to the measurement, used in the software engineering. The ISO/IEC 15939 norm contains also description of the procedure of selecting software product measure, which requires three steps: characteristics of organizational units, identification of their informational needs, and as a result selection of applicable measure. This way it offers help in defining the set of measures being adequate to the specific informational needs yet it neither provides the list of such measures nor it recommends specific set of measures for the software development/enhancement/maintenance project. Therefore one may find the opinion that although employing rules described in the discussed standard is necessary for the measurement process implementation in the organization, these rules *per se*, however, are not sufficient for this purpose [Bégnoche et al., p. 111]. Thus this standard should be linked with other normalized measurement approaches, e.g. IFPUG or COSMIC method.

As indicated by the above, it is hard to unequivocally decide on the advantage of the COSMIC method over the IFPUG method – both have strengths and drawbacks, coming up in the specified problem areas, both have supporters and adversaries. Most probably, COSMIC approach will not totally replace the IFPUG method in the nearest future as this first-generation method has proved being sufficiently objective and reliable approach, at least with regard to the business software systems.

Regardless of differences between the methods being compared, standardization of software functional size measurement methods supports:

- project management by enabling to:
  - make early prognosis on resources necessary for project execution,
  - monitor progress in project execution,
  - manage the changes in the required project product size,
  - determine degree to which the supplied product meets FUR,
  - make post-execution project analysis and compare its attributes with other projects;
- software performance management by:
  - software development, enhancement and maintenance productivity management,
  - quality management, especially of software reliability,
  - organizational processes maturity and capability management,
  - determining organizational software asset value in order to estimate cost of its potential replacement, reengineering, or outsourcing,
  - making prognosis on budget necessary to maintain software,
  - software supply contracts management.

Moreover, setting ISO/IEC standards for software functional size measurement methods brings, among others, the following benefits:

- positive verification of objectivity and reliability of the accepted FSM methods,
- normalization of the rules of particular FSM methods, thanks to which the consistence and coherence in using them get increased,
- stronger acceptance for FSM methods,
- facilitation of the FSM methods automation,
- strict control of changes in FSM methods.

In further works attention should be mostly paid to the possibility of working out the rules of conversion between the results gained with the use of various FSM methods, especially two most popular methods dedicated for business software systems, namely IFPUG and COSMIC methods (see [Czarnacka-Chrobot 2009a]).

References

Albrecht A.J. (1979), Measuring application development productivity, [in:] *Proceedings of IBM Application Development Symposium*, Monterey, CA, October 14-17, pp. 83-92.

Bégnoche L., Abran A., Buglione L. (2007), *A measurement approach integrating ISO* 15939, CMMI and ISBSG, [in:] *Proceedings of Software Measurement European Forum (SMEF) Conference*, Rome, pp. 111-130.

COSMIC (2007a), *The COSMIC Functional Size Measurement Method, Version 3.0, Measurement Manual*, Common Software Measurement International Consortium, Québec.

COSMIC (2007b), *The COSMIC Functional Size Measurement Method, Version 3.0, Method Overview*, Common Software Measurement International Consortium, Québec.

Czarnacka-Chrobot B. (2009a), Konwersja rezultatów kluczowych metod wymiarowania funkcjonalnego systemów oprogramowania wspomagających zarządzanie, [in:] *Systemy wspomagania organizacji 2009*, Prace Naukowe Akademii Ekonomicznej w Katowicach, AE, Katowice (forthcoming).

Czarnacka-Chrobot B. (2009b), Wymiarowanie funkcjonalne produktów programowych w standardach ISO/IEC, [in:] *Komputerowe systemy zarządzania*, Eds. W. Chmielarz, J. Turyna, Wydawnictwa Naukowe Wydziału Zarządzania Uniwersytetu Warszawskiego, Warsaw University (forthcoming).

Heeringen H. (2007), Changing from FPA to COSMIC. A transition framework, [in:] *Proceedings of Software Measurement European Forum (SMEF) Conference*, Roma, pp. 143-154.

IFPUG (2004), *IFPUG Function Point Counting Practices Manual*, Release 4.2, Part 1-4, International Function Point Users Group, Princeton Junction, NJ.

ISBSG (2007), *Release 10 Repository Demographics*, International Software Benchmarking Standards Group, Hawthorn VIC, Australia.

ISO/IEC 14143 (2007), *Information Technology – Software measurement – Functional size measurement – Part 1-6*, ISO, Geneva.

ISO/IEC 15288 (2008), *Systems and software engineering – System life cycle processes*, ISO, Geneva.

ISO/IEC 15939 (2007), *Systems and software engineering – Measurement process*, ISO, Geneva.

ISO/IEC 19761 (2003), *Software engineering – COSMIC-FFP – A functional size measurement method*, ISO, Geneva.

ISO/IEC 20926 (2003), *Software engineering – IFPUG 4.1 Unadjusted functional size measurement method – Counting practices manual*, ISO, Geneva.

ISO/IEC 20968 (2002), *Software engineering – Mk II Function Point Analysis – Counting practices manual*, ISO, Geneva.

ISO/IEC 24570 (2005), *Software engineering – NESMA functional size measurement method version 2.1 – Definitions and counting guidelines for the application of Function Point Analysis*, ISO, Geneva.

ISO/IEC 29881 (2008), *Information Technology – Software and systems engineering – FiSMA 1.1 functional size measurement method*, ISO, Geneva.

Jones T.C. (1999), *Software Project Management in the Twenty-first Century*, Software Productivity Research, Burlington.

PCG (2008), *2008 ERP Report, Topline Results*, Panorama Consulting Group, Denver.

Standish Group (2009), *CHAOS Summary 2009*, West Yarmouth, MA.

State Government of Victoria (2000), *southernSCOPE, Reference Manual*, Version 1, Government of Victoria, Melbourne, Australia.

Xunmei G., Guoxin S., Hong Z. (2006), The Comparison between FPA and COSMIC-FFP, [in:] *Proceedings of Software Measurement European Forum (SMEF) Conference*, Roma, pp. 109-117.

Websites

http://www.ifpug.org/publications/case.htm, Case 4 (9.06.2009).

http://www.cosmicon.com/advantagecs.asp (8.06.2009).

https://www.ifpug.org/publications/case.htm, Case 3 (9.06.2009).