An overview of COSMIC-FFP field trial results

Abran A., Symons C., Oligny S.,

November 2000

Abstract

The Common Software Measurement International Consortium (COSMIC) was formed in 1998 to design and bring to market a new generation of software measurement methods. The COSMIC group reviewed existing functional size measurement methods, studied their commonalities, and proposed the basic principles on which a new generation of software functional size measurement methods could be based. In November of 1999, the group published version 2.0 of COSMIC-FFP, a measurement method implementing these principles, and put its measurement manual on the Web for public access. Over the past year, industrial organizations have contributed data in the context of COSMIC field trials. This report present an overview of the field trial results, including an analysis of the relationship of effort with respect to the software functional size, measured in COSMIC-FFP size units. The data set is described, as well as the constraints for the interpretation of the results.

1. Context

Quantifying the size of software is generally recognized as one of the keys to adequate estimation of effort, cost and schedule of software projects. Source lines of code was the first generally accepted measure for this purpose and is still used extensively, as demonstrated by the many estimation models that include this measure as a key parameter [1, 2]. As a measure of software size though, the source code measure carries some inherent limitations, and this has been recognized by software engineering practitioners and researchers alike [3]. Among the practitioners, Allan Albrecht was the first to propose, over 20 years ago, a new way of quantifying software size based on the user's view of the software [4]. Albrecht's 1979 method, now referred to as the IFPUG method, is still used today and provides useful results in many organizations, but it also has some limitations and these have been well documented over the past 15 years. One of these limitations is the difficulty of applying such a method outside the MIS domain, as documented by [3, 5, 6, 7, 9, 10, 11, 12, 13, 14]. In 1996, the industry sponsored the development of an IFPUG extension for real-time and embedded software, which was put into the public domain under the name of Full Function Points [8, 15, 16].

Building on the strengths of this work and with the support of the industry, the Common Software Measurement International Consortium (COSMIC) was formed in 1998 to design and bring to market a new generation of software measurement methods. The first functional sizing method should be designed to work equally well for 'data-rich' business/MIS software and for 'control-rich' or 'real-time' software, that is software which is to be found in telecoms, avionics, process control, embedded and operating systems. The method should be able to size such software in any layer or peer-item of a multi-layer and/or multi-tier architecture. It is not designed to measure functionality of software which is 'algorithm-rich', that is software which is dominated by complex mathematics.

The COSMIC group reviewed existing methods (IFPUG, MarkII [17], NESMA [18] and version 1.0 of the Full Function Point methods [8]), studied their commonalties, and proposed the basic principles on which a new generation of software functional size measurement method could be based [19, 20, 21]. In November of 1999, the group published version 2.0 of COSMIC-FFP [23], a measurement method implementing these principles, and put its measurement manual on the Web for public access. Overall, close to 40 people from 8 countries participated in the design of this measurement method. The Measurement Manual, describing the method, is available in English, French, Japanese and Spanish. The purpose of this paper is to introduce the COSMIC-FFP functional size measurement method and to present some results of the field trials, gathered over the past year.

2. The measurement method

2.1. COSMIC-FFP measurement systems

The COSMIC-FFP measurement method explicitly defines a measurement system which includes a measurement principle, base functional components, a standard unit of measure and an aggregation function.

- *Measurement principle*: based on the COSMIC-FFP generic software model (Figure 1), the method, through a defined set of rules and procedures, produces a numerical figure to represent the functional size of this model. These rules implement the principle that the functional size of a piece of software is directly proportional to the number of its data-moving sub-processes. By convention, this numerical figure is then extended to represent the COSMIC-FFP functional size of the software itself.

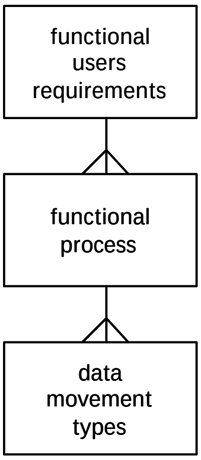

Figure 1: COSMIC-FFP generic software model

- *Base functional components*: version 2.0 of the COSMIC-FFP measurement method uses only four types of data-moving sub-processes, referred to as base functional components in ISO/IEC 14143-1 [22]: entry, exit, read and write; each one is defined in [23].

- *Standard unit of measurement*: the standard unit of measurement, that is, 1 *Cfsu*, is defined by convention as equivalent to one single data movement type at the sub-process level.

- *Aggregation function*: The functional size of the data movement base functional components can be combined to obtain the size of any higher-level functional structures such as that of functional processes, components within layers or whole applications. This is performed by arithmetically adding together the functional sizes of the constituent functional structures according to the purpose of the measurement.

The measurement system proposed by the COSMIC-FFP measurement method offers a scalable result, which means that the functional size figure can be constructed at the desired level of abstraction. Furthermore, as demonstrated by Fetcke in [24], the COSMIC-FFP measurement system meets the dominance and the monotonicity properties; two distinctive and desirable measurement properties, violated by some other functional size measurement methods, and contributing theoretically to a) better predictability of effort estimates (in the case of dominance) [24, pp.150] and b) better predictability of functional size itself (in the case of monotonicity) [24, pp. 152].

2.2. COSMIC-FFP field trials

The field trials of the COSMIC-FFP method were organized:

- to test that the COSMIC FFP V.2 documentation is understandable and interpretable in a common and repeatable way across software developers working in different domains, with different methods of expressing requirements and with different development technologies, and can be applied with acceptable effort;
- to test that the functional size measures obtained properly reflect the functionality of the related software requirements as perceived by experts in the requirements and/or that the functional size measures correlate with development effort;
- to enable a full transfer of the COSMIC FFP method to the organisations participating in the field trials, such that they regard it as 'implemented';
- to establish initial benchmark performance levels according to the COSMIC FFP method, against which the participating organisations can compare their own performance.

A protocol was documented to collect specific and standardized data from each participating industrial partner. This protocol included the subset of compulsory data attributes specified in the data collection protocol of the International Software Benchmarking Standards Group (ISBSG) [25].

Specific reports have been prepared for each of the participating field trials organizations addressing how each of the above field trials aims has been met for their own organization; in addition, specific benchmarking of organizational performance against the full trial data set was delivered to each participant. This report presents an overview of the field trial results, including an analysis of the relationship of effort with respect to the software functional size, measured in COSMIC-FFP size units (Cfsu). The data set is described, as well as the constraints for the interpretation of the results. Further specific reports discussing each of the other field trials aims will be prepared and published over the next few months.

3. General acceptance of the method in the field trials

During the field trials, the COSMIC FFP method was applied to software in different layers and peer items of multi-layer, multi-tier architectures from the application areas of telecommunications (switches and mobile devices), avionics and other defence control software, process control, and banking software. The projects were either new developments, or enhancements to existing systems. As with any measurement process carried out in any field of sciences and engineering, each organisation has to specify its measurement procedure for the application of a measurement method in their specific context. This is necessary to implement the generic functional size measurement rules in the context of the specific organisational means of documenting requirements or functional specifications such as UML Use Cases or SADT diagrams. In all cases, measurement procedures were developed to implement the general COSMIC FFP principle and rules and which satisfied the local need for unambiguous and repeatable interpretation.

Two specific trials were conducted to test repeatability of sizing of the same specifications by different measuring engineers. In the first trial, the documentation of some avionics software was available to a very high standard and the five engineers who each sized two sets of specifications were all familiar with the domain. Near-perfect repeatability was obtained.

In the second trial on specifications of process control software, engineers who were familiar with the domain obtained results of reasonable repeatability. However, junior engineers with limited experience of the domain showed poor repeatability.

Our conclusion is that experience of both the COSMIC FFP sizing method and of the domain are essential to ensure repeatability of sizing. Specific measurement procedures for implementing the method in the context of local documentation standards are also necessary.

The general consensus of those participating in the trials was that the COSMIC FFP functional size method is ‘easy’ or ‘reasonably easy’ to apply. The measurement effort is similar to that needed for established functional sizing methods.

Comments from individual organisations participating in the trials include the following:

- ‘Project Teams were able to grasp the elements of the method easily and were enthusiastic about the method’
- ‘Documentation and effort needed is similar to that for applying the IFPUG method, though there is an extra step to identify layers’
- ‘The Z-unit has now also continued COSMIC FFP measurements with new starting projects and decided to implement this measurement technique as a standard procedure in their development process ’

Finally, organisations that have tried the COSMIC FFP method have realised the value of producing requirements which are *measurable*, (as well understandable, traceable, testable, etc.) In two cases, errors were discovered in the documentation of software during the field trials as a result of the rigour of applying the COSMIC FFP sizing method.

4. Relationship with project effort

Field trial data were received from a variety of sources. The data analysed here have been selected because they belong to the same functional domain: they are all for projects developed for the domain of highly-constrained real-time software.

4.1. Presentation of the data sample

The data used in the analysis below are the 12 software projects from the real time domain; these projects were developed and measured by three different organizations from three countries. Training on the COSMIC-FFP method was organized at these three organizations using the same training material. All projects were completed either in 1999 or early 2000. The total size of the 12 software is 13 961 *Cfsu*; there were therefore 13 961 elementary data movements identified at the functional level. The data sample is presented in Table 1 below. Of the 12 projects, ten projects were for the development of new software applications while the other two were projects implementing functional enhancements to existing software applications. In terms of functional size, four projects delivered under 100 *Cfsu*, six between 142 and 810 *Cfsu* and two projects were much larger, delivering 3004 and 8251 *Cfsu* respectively.

While these organizations measured the functional size using the same measurement standard, that is COSMIC-FFP 2,0, it can be observed from Table 1 that these organizations had different time recording systems, some with details for three project phases (Specify, Build, Test) while others had information only for one or two of these project phases. Effort is reported in hours, as specified in the data collection protocol defined by the International Software Benchmarking Standards Group (ISBSG) [25].

Project	Type	Size (Cfsu)	Specify effort (person-hours)	Build effort (person-hours)	Test effort (person-hours)

A	Development	32	n.a.	252	n.a.
B	Development	76	381	1457	401
C	Development	56	68	487	335
D	Development	142	136	643	n.a.
E	Development	8	115	116	n.a.
F	Enhancement	142	2 060	1 487	5 055
G	Development	332	468	11 382	254
H	Enhancement	624	n.a.	15 815	1 372
I	Development	810	1 304	10 903	4 548
J	Development	484	n.a.	20 808	6 772
K	Development	8251	49 000	66 000	93 000
L	Development	3004	32 000	27 000	20 000
TOTAL		13961			

Table 1 – COSMIC-FFP data sample for selected field trial results

4.2. Relationship between software functional size and effort

The analysis of the data for the sample represented some difficulties for the following reasons:

- Size of the data sample: even though 12 data points is within the usual size of samples in publications (e.g. of the 21 empirical duration models analyzed in [8], 12 of them are derived from samples containing 20 data points or fewer), it is still a small sample for statistical analysis; care must thus be exercised in the interpretations of the results.
- Non-homogeneity of the development processes. Data was received from three different organisations each with their own development processes. Even though their processes could have some similarities within the respective organisations due to the implementation of internal organisational standards, such a level of standardisation cannot be expected across organisations.
- Non-homogeneity in the range of some of the key variables: for example, in this sample, the two larger projects have a functional size of an order of magnitude greater than the other ten projects. In this context, a simple regression model cannot be used since one of its key requirements is not met, that is that the variables must have a gaussian distribution (e.g. a normal distribution). A mathematical transformation is then required: in this specific instance, a log transformation was used to normalise the data set. This allowed then for a proper use of the regression models, while making data interpretation more difficult for practitioners not familiar with interpretation of figures with logarithmic scales. These comments also apply to the other key variable, the project effort (and for the same two projects with the largest effort).
- Non-homogeneity in the standards across organisations for data collection of the effort variable: this is directly illustrated in the data set by the different abilities of the organisation of recording effort by project phases (Specify, Build and Test phases), and indirectly since each organisation had their own practices in defining project phases, even though the reporting was based in their best effort to match the ISBSG definition of project phase.

4.2.1. Effort to specify

The effort variable is available for the Specify phase for nine projects. The log-log representation of this subset is presented in Figure 2. The coefficient of regression R^2 is 0,7824 for the linear regression model:

$$\text{Specify Effort} = 4{,}0342 \text{ x } (\text{Size})^{0{,}9903}$$

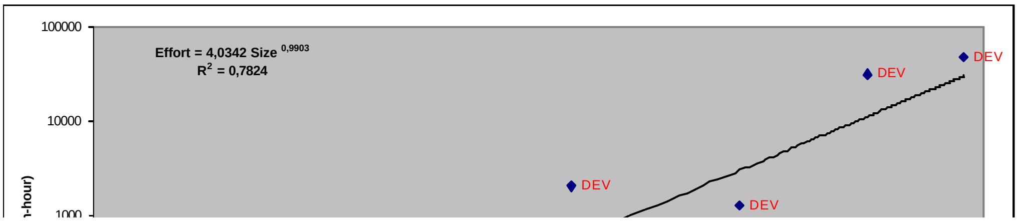

Figure 2 – Specify phase – Regression model on Log-log plot (n = 9, R^2 = 0,7824)

4.2.2. Effort to build

The effort variable is available for the Build phase of the full set of 12 projects. The regression model for these 12 is presented in Figure 3. The coefficient of regression R^2 is 0,8971 for the linear regression model:

$$\text{Build Effort} = 12{,}313 \text{ x } (\text{Size})^{1{,}015}$$

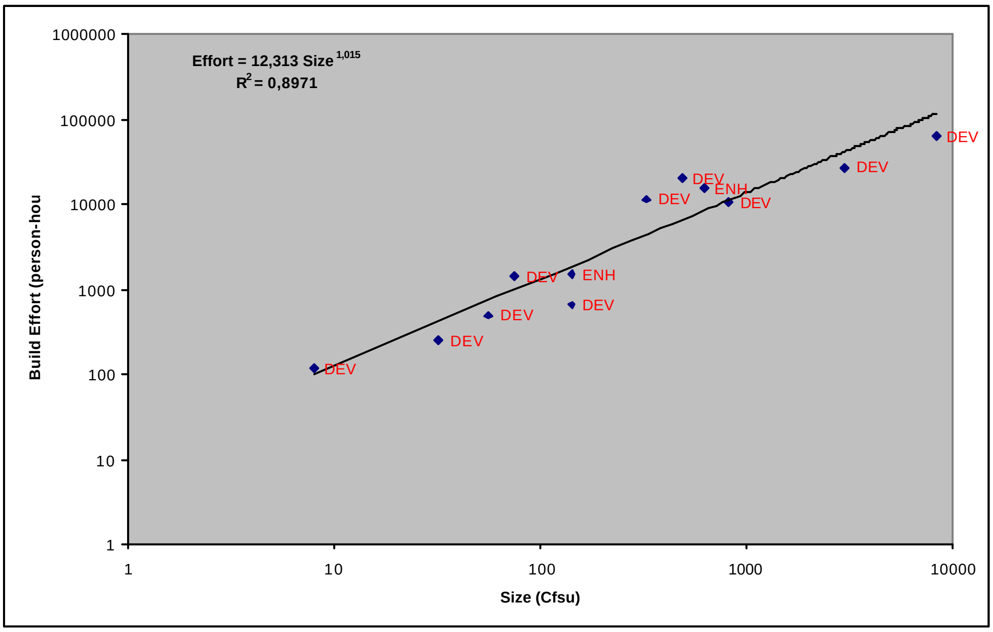

Figure 3 – Build phase regression model on Log-log scale (n = 12, R^2 = 0,8971)

4.2.3. Effort to test

The effort variable is available for the Test phase for 9 projects. The regression model for these 9 projects is presented in Figure 4. The coefficient of regression R^2 is 0,7019 for the linear regression model:

Test Effort = 5,2124 x $(\text{Size})^{1,024}$

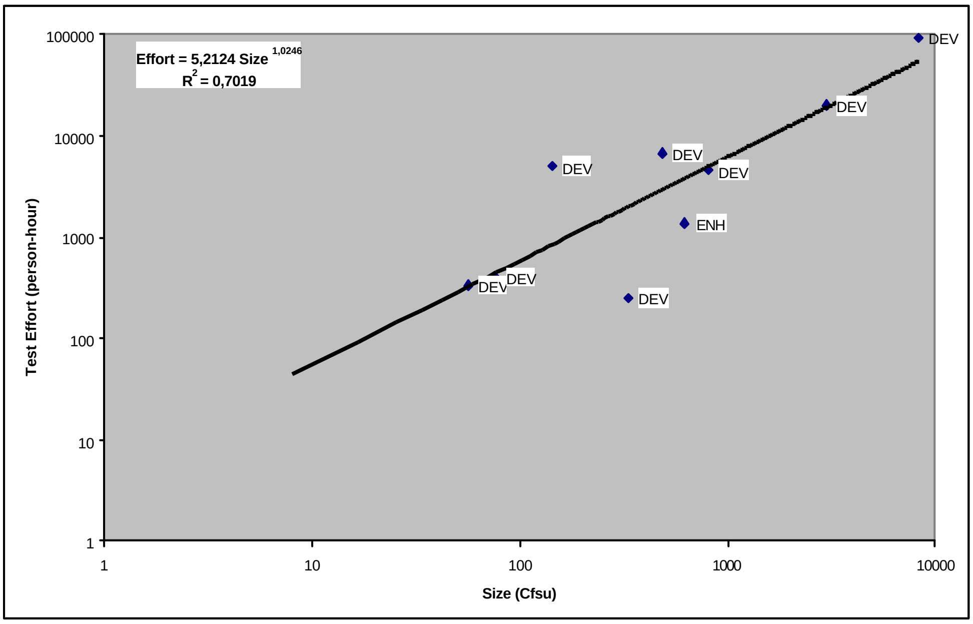

Figure 4 – Test phase regression model on Log-log scale
($n = 9$, $R^2 = 0,7019$)

5. Concluding remarks.

This paper has presented the highlights of the analysis of the relationships of project effort with respect to the functional size variable as measured with COSMIC-FFP. Taking into account the significant non-homogeneity of the origin of the data sample (three organisations from three different countries with different development methodologies and platforms), we would have expected much less consistency on the relationship across both variables. For all of the three project phases for which data was available, the coefficients of determination between size and effort, on a log scale, varied between 0,7019 and 0,8971 depending on the project phase. These results are of course of a preliminary nature, and should be considered only as indicative due to the nature of the data set.

On the other hand, such results have been highly valued within the organisation which provided the largest number of projects within this sample. Using the measurements from the projects contributed to the COSMIC field trials, they have built their own productivity and estimation models, using COSMIC-FFP as the standard for functional size measurement and adding other in-house variable to take into account such productivity factors as programming platform and levels of reuse. Of course such in-house estimation models can be expected to perform much better than models built from non homogeneous multi-organisational data sets.

The COSMIC FFP method is the first Functional Sizing method to

- be designed by an international group of experts on a sound theoretical basis
- draw on the practical experience of all the main existing FP methods
- be designed to conform to the ISO standard on 'Functional Size Measurement' (ISO/IEC 14143-1)
- be designed to work across MIS and real-time domains, for software in any layer or peer item
- be tested in field trials before being finalised

6. Acknowledgements

The authors of this paper wish to acknowledge the data collection contributions of the field trials organizations who for competitive reasons have selected to remain anonymous.

7. Bibliography

[1] B.W. Boehm, R.W. Wolverton, "Software cost modelling: some lessons learned", *Journal of System and Software*, 1:195-201, 1980.

[2] V. Côté, P. Bourque, S. Oligny, N. Rivard, "Software metrics: an overview of recent results", *Journal of System and Software*, 8:121-131, 1988.

[3] C. Jones, *Applied software measurement - Assuring productivity and quality,* 2nd Edition. New York, NY: McGraw-Hill Inc., 1996.

[4] A.J. Albrecht, "Measuring Application Development Productivity," presented at *IBM Applications Development Symposium*, Monterey, CA, 1979.

[5] S.A. Whitmire, "3D Function Points: Scientific and Real-Time Extensions to Function Points," presented at *Pacific Northwest Software Quality Conference*, 1992.

[6] D.J. Reifer, "Asset-R: A Function Point Sizing Tool for Scientific and Real-Time Systems," *Journal of Systems Software*, Vol. 11, p. 159-171, 1990.

[7] T. Mukhopadhyay and S. Kekre, "Software effort models for early estimation of process control applications," *IEEE Transactions on Software Engineering*, Vol. 18, p. 915-24, 1992.

[8] A. Abran, J.M. Desharnais, M. Maya, D. St-Pierre, P. Bourque, "Design of a functional size measurement for real-time software", Research report no. 13, Software Engineering Management Research Laboratory, Université du Québec à Montréal, Montreal, Canada, November 1998.

[9] S.D. Conte, H. E. Dunsmore, V. Y. Shen, *Software engineering metrics and models*. Menlo Park: The Benjamin/Cummings Publishing Company, Inc., 1986.

[10] R.B. Grady, *Practical software metrics for project management and process improvement*. Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1992.

[11] B. Hetzel, *Making Software Measurement Work - Building an Effective Measurement Program.* Boston: QED Software Evaluation Series, 1993.

[12] S.H. Kan, *Metrics and models in software quality engineering*. Readings, Massachusetts: Addison-Wesley Publishing Company, 1995.

[13] D.C. Ince, "History and industrial application", in N.E. Fenton, *Software metrics: a rigorous approach,* Chapman & Hall, UK, 337 pages, 1991.

[14] S. Galea, "The Boeing Company: 3D function point extensions, v. 2.0, release 1.0", Boeing Information and Support Services, Research and Technology Software Engineering, June 1995.

[15] A. Abran, M. Maya, J.M. Desharnais, D. St-Pierre, "Adapting Function Points to real-time software", *American Programmer*, Vol. 10, no. 11, p. 32-43, November 1997.

[16] D. St-Pierre, M. Maya, A. Abran, J.M. Desharnais, P. Bourque, "Full Function Points: Function Points Extension for Real-Time Software - Counting Practices Manual", Technical Report no. 1997-04, Software Engineering Management Research Laboratory, Université du Québec à Montréal, Montreal, Canada, September 1997. Downloadable at http://www.lrgl.uqam.ca/ffp.html.

[17] C.R. Symons, *Software sizing and estimating – MkII FPA (function point analysis)*, John Wiley & sons, Chichester, UK, 1991.

[18] The Netherlands Software Metrics Users Association (NESMA), "Definitions and counting guidelines for the application of function point analysis, version 2.0", 1997.

[19] C.R. Symons, P. G. Rule, “One size fits all – COSMIC aims, design principles and progress”, Proceedings of *ESCOM ’99*, p. 197-207, April 1999.

[20] A. Abran, “FFP Release 2.0: An Implementation of COSMIC Functional Size Measurement Concepts”, Proceedings of *FESMA ‘99*, Amsterdam, Oct. 1999.

[21] C.R. Symons, “ COSMIC aims, design principles and progress”, Proceedings of *IWSM ’99*, pp. 161-172, September 1999. Proceedings are downloadable at http://www.lrgl.uqam.ca/iwsm99/index2.html.

[22] International Organization for Standardization (ISO), “ISO/IEC 14143-1:1997 – Information technology – Software measurement – Functional size measurement – Definition of concepts”, October 1997.

[23] A. Abran, J.-M. Desharnais, S. Oligny, D. St-Pierre, C. Symons, “COSMIC-FFP Measurement Manual, version 2.0”, Ed. S. Oligny, Software Engineering Management Research Laboratory, Université du Québec à Montréal, Montreal, Canada, Oct. 1999. Downloadable at http://www.lrgl.uqam.ca/ffp.html

[24] T. Fetcke, “A Generalized Structure for Function Point Analysis”, Proc. of *the International Workshop on Software Measurement (IWSM ‘99)*, Lac Supérieur, Canada, Sept. 1999, Downloadable at http://www.lrgl.uqam.ca/ffp.html.

[25] International Software Benchmarking Standards Group (ISBSG), ISBSG. *Worldwide Software Development - The Benchmark*. Victoria, Australia International Software Benchmarking Standards Group, 1997. http://www.bs.monash.edu.au/asmavic/isbsg.htm.

[26] B.A.Kitchenham. Empirical studies of assumptions that underlie software cost-estimation models. *Information and Software Technology* 34(4):211-218 1992.