Advances in Engineering Software 41 (2010) 262–269

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Horizontal dispersion of software functional size with IFPUG and COSMIC units

Juan J. Cuadrado-Gallego [a,b], Luigi Buglione [a,b,*], Marian Fernández de Sevilla [a], Pablo Rodríguez-Soria [a], Maria J. Dominguez [a]

[a] Universidad de Alcalá, Departamento de Ciencias de la Computación, 28805 Alcalá de Henares, Madrid, Spain
[b] Ecole de Téchnologie Superieure (ETS) – Université du Québec à Montreal/Nexen (Engineering Group), 00148 Rome, Italy

ARTICLE INFO

Article history:
Received 13 February 2009
Received in revised form 9 April 2009
Accepted 17 July 2009
Available online 19 August 2009

Keywords:
Software engineering
Software measurement
Functional Size Measurement (FSM)
IFPUG
COSMIC

ABSTRACT

Software development companies today are widely using software functional size measurement (FSM) as the main variable to assess the effort and time needed to perform a new software project. In the recent years, this has led to a grown interest in improving the way the measures are taken.

In such sense, one of the main aspects that could have impact on measurements and that has not been enough studied is the error introduced by the measurer of the software application, through the subjectivity that can be introduced in the interpretation of the unit application rules. Such error could be evident in a measurement dispersion, defined in this paper in two possible ways: (a) Horizontal Dispersion, where the error could be introduced by the fact that two or more different people counted the same application at the same moment in the project development; and (b) Vertical Dispersion, where the error could be introduced by same measurer that count the same application at different times during the development.

Since its definition by Albrecht in 1979 and its subsequent change of name in 1986, IFPUG function points have been the functional software measurement unit mostly applied, despite the definition and standardization of other variants such as NESMA, Mk-II, or more recently FiSMA. However in recent years a new method has been introduced called COSMIC that has been defined as a 2nd-generation FSM method, attracting the interest of the international software measurement community.

The aim of this research is to draw some preliminary conclusions from statistical analysis of the software functional size data in which the horizontal dispersion degree could have been introduced in measurements taken into account IFPUG and COSMIC methods.

1. Introduction

Software development companies today are widely using software functional size measurement (FSM) as the main variable to assess the effort and time needed to perform a new software project. In the recent years, this has led to a grown interest in improving the way the measures are taken.

Since its definition by Albrecht in 1979 and its subsequent change of name in 1986, the function points IFPUG have been the functional software measurement unit more widely used, despite the definition and standardization of other variants such as NESMA, MkII, or more recently, FiSMA. However in recent years a new method called COSMIC has been introduced and it has been defined as a 2nd-generation method for functional software measurement, with a diffused interest from the international community. Therefore a huge number of worldwide companies that solely used the IFPUG FPA method until now would start to measure also with COSMIC [21]. The results is that nowadays IFPUG and COSMIC are the most important FSM methods for sizing software functionalities. This is the rationale for a study on the dispersion of these two methods when applied to the same application and, in that sense, this research paper attempts to analyze the error introduced in the measures done with IFPUG and COSMIC methods.

In such sense, one of the main aspects that could have impact on the measurements and that has not been enough studied, is the error introduced by the people who measures the software, through the subjectivity that can be introduced in the interpretation of the unit application rules. That subjectivity could originate fundamentally from two sources, different personality or different expertise, for instance a Certified Function Points Specialist (CFPS) measure vs. a non-CFPS one. In such case, only the first source has been explored, leaving the other for future works. Such error could take to a measurement dispersion that is defined in this work in two ways: (a) *Horizontal Dispersion*, which is one that could be

* Corresponding author. Address: Ecole de Téchnologie Superieure (ETS) – Université du Québec à Montreal/Nexen (Engineering Group), 00148 Rome, Italy. Tel.: +39 335 1214813; fax: +39 06 8307 4200.
E-mail addresses: luigi.buglione@nexen.it, luigi.buglione@computer.org (L. Buglione).

0965-9978/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2009.07.003

Appendix A.2. (*continued*)

N	*E*	*X*	*R*	*W*	FFP
45	6	4	3	6	19
46	19	10	13	12	54
47	24	19	12	6	61
48	9	14	13	6	42
49	23	19	14	6	62
50	11	13	13	9	46
51	6	11	14	5	36
52	9	9	11	6	35
53	10	13	9	6	38
54	3	16	21	7	47
55	11	16	6	7	40
56	12	31	19	6	68
57	12	10	7	5	34
58	3	15	20	7	45
59	11	13	13	7	44
60	8	16	13	6	43
61	22	15	15	4	56

Notes on tables: ***N***: Project identification; **ILF**: Number of function points per ILF; **EI**: Number of function points per EI; **EO**: Number of function points per EO; **EQ**: Number of function points per EQ; **UFP**: Total number of UFP; ***E***: Number of function points per *E*; ***X***: Number of function points per *X*; ***W***: Number of function points per *W*; ***R***: Number of function points per *R*; **FFP**: Total number of FFP.

References

[1] Albrecht AJ. Measuring application development productivity. In: Proceedings of joint SHARE, GUIDE, and IBM application development symposium; 1979. p. 83–92.

[2] Albrecht AJ, Gaffney JE. Software function, source lines of code, and development effort prediction: a software science validation. IEEE Trans Soft Eng 1983;9(6):639–47.

[3] IFPUG. Function point counting practices manual, release 4.2.1. International Function Point User Group; 2004. ISBN: 0-963-1742-9-0.

[4] Gencel C, Demirors O. Functional size measurement revisited. ACM Trans Soft Eng Method (TOSEM) 2008;17(3).

[5] Abran A, Robillard P. Function points: a study of their measurement process and scale transformations. J Syst Soft 1994;25:171–84. <www.bfpug.com.br/Artigos/Neutros/abran94function.pdf>.

[6] ISO/IEC 20926:2003. Software engineering – IFPUG 4.1 unadjusted functional size measurement method. Counting practices manual. Genève: International Organization for Standardization; 2003.

[7] ISO/IEC 24750:2005. Software engineering – NESMA functional size measurement method, version 2.1. Definitions and counting guidelines for the application of function point analysis. Genève: International Organization for Standardization; 2005.

[8] ISO/IEC 20968:2002. Software engineering – Mk-II function point analysis. Counting practices manual. Genève: International Organization for Standardization, ISO; 2002.

[9] ISO/IEC 19761:2003. Software engineering – COSMIC-FFP. A functional size measurement method. Genève: International Organization for Standardization; 2003.

[10] ISO/IEC 29881:2008. Information technology – software and systems engineering – FISMA v1.1 functional size measurement method. Genève: International Organization for Standardization; 2008.

[11] NESMA. Function point analysis for software enhancement – version 1.0. Netherlands Software Metrics Association; 2001. <http://www.nesma.org>.

[12] Symons C. Function point analysis: difficulties and improvements. IEEE Trans Soft Eng 1988;14(1):2–11.

[13] COSMIC. COSMIC measurement manual 3.0.1. Common Software Measurement International Consortium; May 2009.

[14] ISBSG. Repository data – release 8. International Software Benchmarking Standards Group; 2005. <http://www.isbsg.org/>.

[15] ISBSG. Repository data – release 11. International Software Benchmarking Standards Group; 2009. <http://www.isbsg.org/>.

[16] IEEE Std 830-1998. In: IEEE recommended practice for software requirements specifications. Institute of Electrical and Electronics Engineers; 1998.

[17] Fetcke T. The warehouse software portfolio. A case study in functional size measurement. Technical report no. 1999–20, Département d'informatique, Université du Quebec à Montréal, Canada. <http://www.fetcke.de/papers/Fetcke1999b.pdf>.

[18] Ho VT, Abran A, Fetcke T. A comparative study case of COSMIC-FFP, full function point and IFPUG methods. Département d'informatique, Université du Québec à Montréal, Canada. <www.lrgl.uqam.ca/publications/pdf/599.pdf>.

[19] Desharnais JM, Abran A, Cuadrado-Gallego JJ. Convertibility of function points to COSMIC: identification and analysis of functional outliers. In: Proceedings of the 17th international workshop on software measurement, Shaker-Verlag, Palma de Mallorca, Illes Balears, España, November 5–8, 2007. p. 130–46. <http://www.gelog.etsmtl.ca/publications/pdf/1035.pdf>.

[20] Abran A, Desharnais J-M, Azziz F. Measurement convertibility: from function points to COSMIC-FFP. In: Proceedings of the 15th international workshop on software measurement, Shaker-Verlag, Montréal, Canada, September 12–14, 2005. p. 227–40. <http://www.gelog.etsmtl.ca/publications/pdf/906.pdf>.

[21] Vogelezang F, Lesterhuis A. 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, Shaker-Verlag, Montréal, Canada, September 22–25, 2003. p. 232–43.