Determining the Standard Deviation for COSMIC Software Functional Size Measurement

Pablo Rodríguez-Soria[1], Borja Martín-Herrera, Marian Fernández de Sevilla, José J. Martínez-Herráiz, María J. Domínguez-Alda, Alberto Lucendo Patiño

CuBIT, Software Measurement Lab., Department of Computer Science, University of Alcalá, Madrid, Spain

[1]Contact Author: pablo.rsoria@uah.es / Conference: **SERP'09**

Abstract – *Software Functional Size has become over the years in the main variable to carry out the effort and time needed to perform a software project. This growth has led to the interest shown in the study and development of these units of measurement as well as the optimization of them. IFPUG Function Points has been since its definition the more widely used model. However, has recently emerged a new method called COSMIC that has brought new features and benefits being proposed as a second generation unit for functional software measurement.*

The aim of this research paper is the study of the error introduced in the interpretation of the unit application rules, focusing on COSMIC unit. This error can lead to measurement dispersion due to the subjectivity when the application is measured. Thus, conclusions about the dispersion degree generated will be drawn according to the software functional size data statistical analysis of its main variables.

Keywords: Software Engineering, Software Measurement, Functional Size Measurement, IFPUG, COSMIC.

1 Introduction

In order to be able to correctly analyze the basis of Software Measurement we have certainly to talk about Software Functional Size, which has become a key aspect in managing software projects by estimating the effort and time required (amount of personnel, time and cost, resources, etc.).

Following this line, the first method used in the measurement of software functional size was the SLOC (Source Lines of Code), which consists into calculate the amount of source lines of code and then apply some equation to consider the effort estimated. This unit was showed efficient in the estimates of different aspect such as error ratios or team productivity ratios but had the inconvenient that you cannot measure the functional size of the given software project until the application was completely built.

Different lines of research were tried until Allan Albrecht [1] proposed in 1979 a new and revolutionary measurement unit called Function Points. He defined with the collaboration of his colleague John Gaffney [2] this new method based on the functional size of the different applications. The main novelty of this method consists in that you can apply it when the documentation is available during early phases in the project and is not necessary to wait until the application is built to measure the amount of functionality of each application.

Then, in 1986 the *International Function Points User Group* (IFPUG) was founded in order to solve software project management issues and to standardize and to properly define the operation of this unit and its project application. Thus, this measurement unit changed its name from Albrecht´s Function Points to IFPUG Function Points Analysis (IFPUG FPA).

Several versions of IFPUG FPA have been developed since then [9-15] (the actual version is v4.2), and consequently several measurement methods or units have been produced on the basis of Function Points [8]. These methods present some advantages compared with their predecessor. Next we are going to list the most relevant and the ones that have been recognized as the *International Organization for Standardization* (ISO) *de jure* standards:

- IFPUG v.4.1, 1998. Standard ISO/IEC 20926 [17]
- NESMA v.2.1, Standard ISO/IEC 24570 [19]
- MK II v.1.3.1, Standard ISO/IEC 20968 [20]
- COSMIC v2.2. Standard ISO/IEC 19761 [18]
- FISMA FPA v1.1 Standard ISO/IEC 29881 [7]

After the analysis made along this article the following key findings were obtained:

- The dispersion of the measurements using the unit of measurement COSMIC conforms to a normal disperse, in such a way that in an interval of 60% around the average (AV) are located the 95% of the measures taken.

  And in an interval of 50% around the average (AV) are located the 90 % of the measures taken.

  In this case, for the 95% of the data for COSMIC, the 60% of the values around the average could be considered that these would be the maximum limits of the horizontal dispersion of measurements. The intervals become narrower with values of 50% around the average for the 90% of the data. Since they have done with measurers with low experience it would be logical to think that if the measures were undertaken with measurers with more experience the margins of error would be lower.

- The second conclusion would relate to the identification of the main sources of error in the performance of the measures. Thus, for COSMIC we have identified the variable W as the main source of error or dispersal.

In addition, as part of this research the problem of the data collection has also been addressed for this kind of studies and, in that sense, it has been proposed a repeatable and contrasted procedure to the obtaining of reliable data in an academic environment.

As future work in the scope of this research are proposed the following:

- Implementation of new analysis on new sets of data with the objective of verifying and scrub the results obtained in this study, in particular those relating to the sources of dispersion.
- Implementation of new analysis on a sample obtained with expert measurers to experimentally test the conclusion that the dispersion that is produced in such sample is lower than that produced in a sample obtained with low expert measurers.

Acknowledgement

We would like to thank the University of Alcalá for supporting this research (Ph.DC researchers support programme).

References

[1] Albrecht A. J., "Measuring application development productivity," en Proc. Joint SHARE, GUIDE, and IBM Application Development Symposium, IBM, pp. 83-92.

[2] Albrecht A. J. & Gaffney J. E., "Software function, source lines of code, and development effort prediction: A software science validation," IEEE Trans. Software Eng., vol. 9, no. 6, pp. 639-647.

[3] Common Software Measurement International Consortium, “COSMIC-FFP Measurement Manual 2.0”

[4] Common Software Measurement International Consortium, “COSMIC-FFP Measurement Manual 2.1”

[5] Common Software Measurement International Consortium, “COSMIC-FFP Measurement Manual 2.2”

[6] Common Software Measurement International Consortium, “COSMIC-FFP Measurement Manual 3.0”

[7] FISMA, “PAS Submission to ISO/IEC JTC1/SC7 – Information Technology – Software and Systems Engineering – FISMA v1.1 Functional Size Measurement Method”, 2006, Finnish Software Metrics Association, URL: www.fisma.fi/wp-content/uploads/2007/02/fisma_fsmm_11_iso-final-1.pdf

[8] Gencel C., Demirors, O,, “Functional Size Measurement Revisited”, Scheduled for publication in ACM Transactions on Software Engineering and Methodology, July 2008, URL: http://tosem.acm.org/

[9] International function points users group, “Function points counting practices manual 2.0”

[10] International function points users group, “Function points counting practices manual 3.0”

[11] International function points users group, “Function points counting practices manual 4.0”

[12] International function points users group, “Function points counting practices manual 4.1”