ISMA 4 Conference

September 13-16, 2009 , Chicago, United States

Juan J. Cuadrado Gallego

CuBIT Laboratory, www.cc.uah.es/cubit

AEMES, Spanish Software Metrics Association

University of Alcalá, Madrid, Spain

jjcg@uah.es

Horizontal Measurement Dispersion of Functional Size with IFPUG

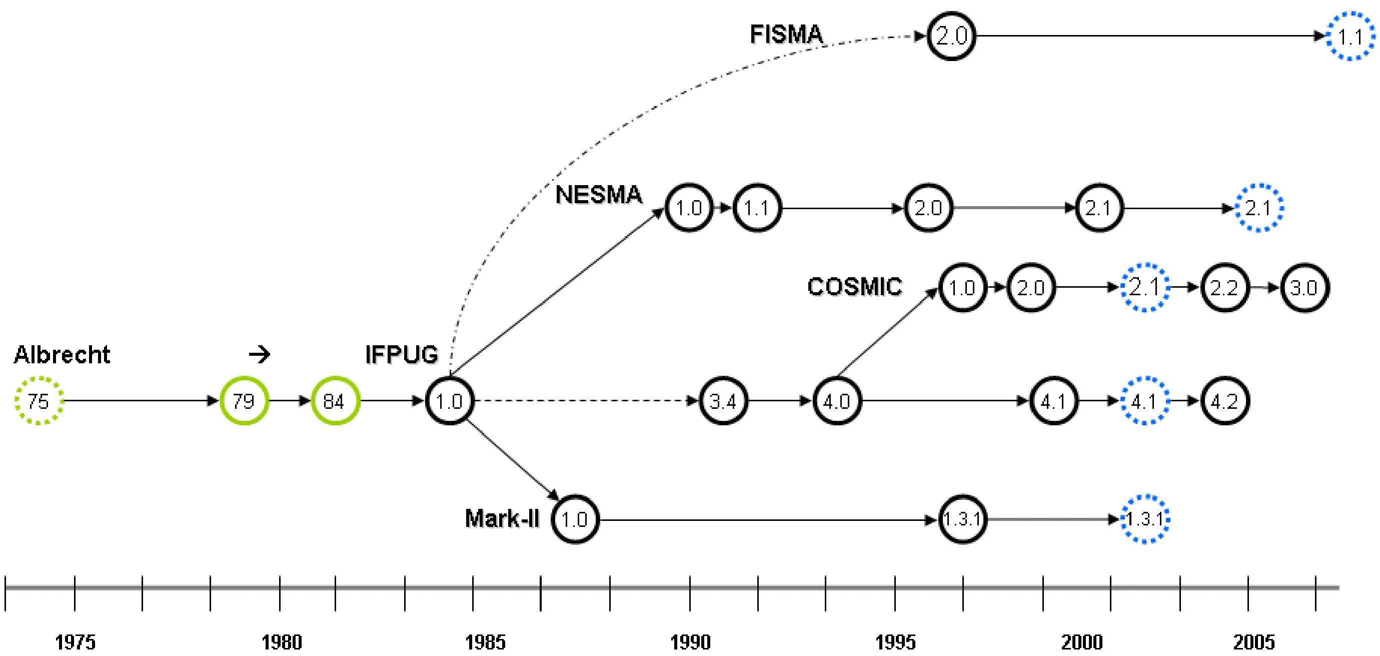
Juan J. Cuadrado Gallego
CuBIT Laboratory, www.cc.uah.es/cubit
University of Alcalá, Madrid, Spain
jjcg@uah.es

Introducción I

- Derivable from the projects' functional user requirements (FUR), it is possible to estimate the amount needed of required human and material resources, time and costs, and that is a main aspect for the project development success.
- One of the most significant figures for managing a software project is its functional size.
- A new measurement unit, based on the software functional size, called function points and the related measurement method.
- The method was developed to measure the amount of functionality to be delivered to end users as perceived from their viewpoint.

Introducción II

- An evolutionary view of Function Software Measurement (FSM) units.



Introducción III

- The research is organized as follows:
  - Analyzes main difficulties in obtaining the needed data for realizing this type of investigation.
  - To describe the methodology used in this work.
  - To present results obtained applying IFPUG analysis on those data.
  - To show the conclusions for this work and outlines future research issues.

Data gathering

- Data gathering methodology and problems.
  - The data gathering is one of the main problems, which comes from data gathering costs.
  - Data gathering in industrial environments is practically impossible.
  - In order to solve this problem there is basically only one solution: the researchers, coming from other institutions, usually a university or research centre, do the measures by themselves, using a set of software specifications provided by companies under agreements subscribed between both organizations with a group of students.
  - In order to maintain the highest data quality possible the following steps were taken.
    - Samples generation
    - Measurement of a real world application with the IFPUG method.
    - Selection of the students participating in the measurements.
  - 77 students were selected to participate in the IFPUG measurement process.

Data Analysis I

- The first step of the analysis consisted of the definition of variables in IFPUG that were going to be analyzed.

- Defined IFPUG variable:
  - FP, IFPUG unadjusted Function Points
  - EI, External Input
  - EO, External Output
  - EQ, External Inquiry
  - ILF, Internal Logic File
  - EIF, External Interface File

Data Analysis II

- Statistical analysis of variable
- Results:

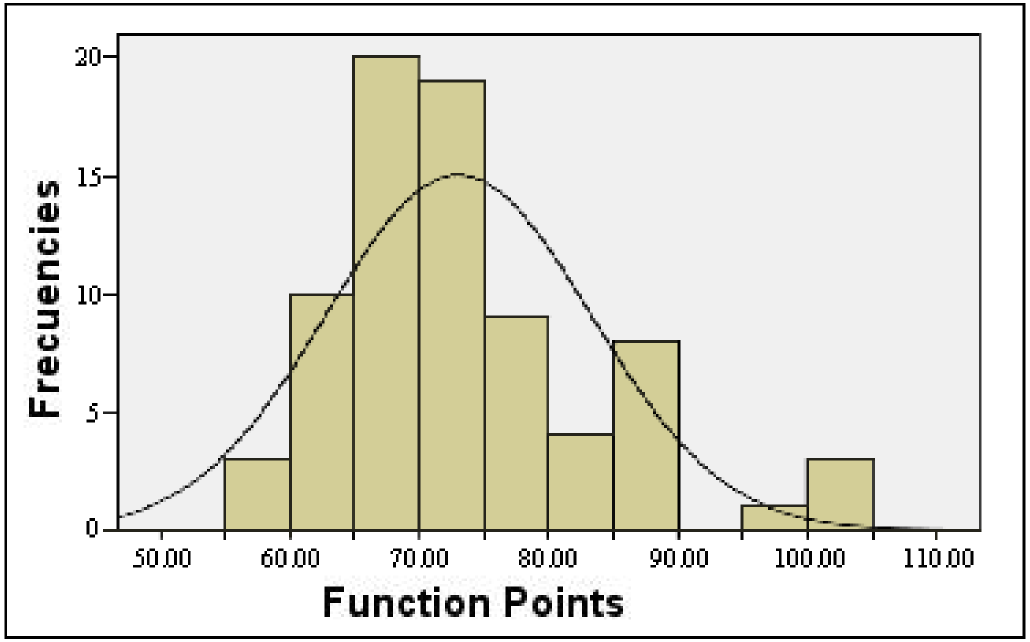
	ILF	EIF	EI	EO	EQ	UFP
Avg	2,29	2,23	6,03	1,2	5,01	73,01
Med	2	2	6	1	5	71
Mod	2	2	6	1	5	68
S	0,6	0,78	0,73	0,87	2,36	10,22
Min	1	1	3	0	1	57
Max	5	7	9	6	12	104
AV	2	2	6	1	7	74

Data Analysis III

- It can be observed that the standard deviation to the variables ILF, EIF, EI and EO has a value close to zero, which indicates that most of the measures were well executed and close to the real value, with the exception of the EQ
- It can be observed that the measures of central tendency are practically identical to almost all the variables, and also coincide with the real value.
- It would show that identifying these variables is an important source of horizontal dispersion.
- The variable EQ, statisticians of central tendency agree among themselves, but on the other side, they move away from real value.
- It can be observed that the maximum and minimum values are separated very significantly from the real value.

Data Analysis IV

- The measures of dispersion quantified the separation, the dispersion and variability of the values of the distribution over to the central value, and this is the main objective of this research.
- To analyze the shape of the dispersion of function data total points more in-depth.



Data Analysis V

- The distribution of the totals function points in the sample conformed to a normal distribution, which allows contrasting the shape of the distribution of each sample in a Normal distribution, of Poisson, Uniform and Exponentially.

N	77
Z de Kolmogorov-Smirov	1,3179071 2
Sig. asintot. (bilateral)	0,0619987 5

- The null hypothesis is rejected when the P-value is less than the level of significance set, which in this case has been p = 0,05. In the case of the sample being analyzed the p-value is 0,06 so that you cannot reject the null hypothesis and can therefore assume that the total data function points presented a Normal distribution.

Conclusions I

- The fundamental objective raised in this research was determine the margin of error that could be considered that is introduced by default on the measurements made with this unit. That is the analysis of the horizontal dispersion of the function points on an application measured by different measurers with the measurement units IFPUG.
- Key findings:
  - The horizontal dispersion of the measurements using the unit of measurement IFPUG conforms to a normal disperse, in such a way that in the interval
  - are located th $x_{av} \pm 0.18 x_{av}$ s taken. And in the interval
  - are located th $x_{av} \pm 0.23 x_{av}$ taken.

Conclusions II

- It is logical to think that if the measures were undertaken with measurers with more experience the margins of error would be lower.
- The main sources of error in the measures. We have found that it comes from the identification of the external queries, EQ and could come from the determination of the complexities of the functions identified.

Future Research

- Implementation of new analysis on a sample obtained with experts measurers to experimentally test the conclusion that the dispersion that occurs in such sample is lower than that produced in a sample obtained with low expert measurers.

- 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.

Questions?

www.cc.uah.es/cubit

Impact of measurer in IFPUG functional size measurements

Juan J. Cuadrado-Gallego[a,b,c], Isaac Sánchez[a], Miguel Ángel Herranz[a], José Luis Cuadrado[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*
[c] *AEMES, Asociación Española de Métricas de Sistemas Informáticos*

Abstract

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.

In the last years, because of the growing use, done by the software development companies of the software functional size measurement as the main variable to carry out the effort and time needed to perform a new software project, has led that the interest in improving the way of taking that measures.

The error introduced by the person or people who measures the software, through the subjectivity that can be introduced in the interpretation of the unit application rules. It is one of the main impact aspects that could have on the measurements and that has not been enough studied. Such error could take to a measurement dispersion that is defined in this work Horizontal Dispersion; which is one that could be introduced by the fact that two or more different people counted the same application at the same moment in the project development.

The aim of this research is to draw conclusions from software functional size data statistical analysis about which may be the horizontal dispersion degree that could be introduced in measurements taken with IFPUG unit.

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

1. Introduction

Derivable from the projects' functional user requirements (FUR), it is possible to estimate the amount needed of required human and material resources, time and costs, and that is a main aspect for the project development success.

The first unit used to measure the size of software products was given by the amount of source lines of code (SLOC). SLOC suffers from the fact that it can only be measured once the software has been built. Although this unit is useful when is used to analyze different aspects such as error ratios or team productivity ratios, from a software project management point of view. For that reason, the definition of a magnitude able to measure software, for management issues, early in the project lifecycle became essential.

One of the most significant figures for managing a software project is its functional size.

A solution to this problem was defining a new measurement unit, based on the software functional size, called *function points* and the related measurement method. That was given by IBM's researcher Allan Albrecht, first by himself [1] and then working with his collaborator John Gaffney [2]. This measurement unit can be applied when the documentation is available during project's early phases, such as the software requirements specification and analysis phases. The enacting need for a software measurement unit as the one proposed by Albrecht, able to solve software projects management issues, together with the success derived from its first applications, were the reasons for the foundation, in 1986, of the *International Function Point Users Group* (IFPUG), whose main goals is to promote the usage of this measure and to control the evolution of the measurement standard definition. The method was developed to measure the amount of functionality to be delivered to end users as perceived from their viewpoint.

Since its appearance in 1979, several variants were produced during the years [22]. This changed the name from Albrecht's Function Points to IFPUG Function Points Analysis (FPA). Since then, several versions of IFPUG FPA have been published [9-15].

Among those methods, five ones have been recognized by the International Organization for Standardization (ISO) *de jure* standards:

- IFPUG v.4.1, 1998. Standard ISO/IEC 20926 [18]

- NESMA v.2.1, Standard ISO/IEC 24570 [24]

The origin of the NESMA FPA is presented on a manual published by the *Netherlands Software Metrics*

Association (NESMA), where it explains how to apply the IFPUG measurement unit, in particular to software developed during maintenance projects. This manual, of which five versions has been published [26-30], has had a large impact on the software industry; that is the reason why the NESMA's Function Points are considered both a measurement unit and an international standard. NESMA FPA represents a minor variation from the IFPUG method and therefore it is possible to consider the two related functional size units as equivalent.

- MK II v.1.3.1, Standard ISO/IEC 20968 [25]

Inspired in the IFPUG FPA, but with some foundations introducing noteworthy differences from them, MK II's Function Points were published by Charles Symons [31] as new unit proposal for software functional size measurement. The MK II unit obtained spread reach, mainly in the nineties, not only in the United Kingdom from where they belong, but also in many other countries. The reasons behind its success lay in the belief that this unit improves from IFPUG FPA in the way to consider the internal complexities on data handling, a key aspect on business. All these reasons led to promote this measurement unit as an international standard. Nevertheless, the facts that Charles Symons participated also in the development of COSMIC unit, and is currently working in its development and sponsorship, should introduce certain scepticism about the future of the acceptance and usage of the MK II unit.

- COSMIC v2.2. Standard ISO/IEC 19761[19]

To organize the execution of the tasks that lead to the definition of the new measure, some experts established in 1998 the Common Software Measurement International Consortium, COSMIC, which first outcome was the definition of the measure in 1999 [3]. Since its first publication, the interest in both the academic community and the industry by the new unit was enormous, reaching vast diffusion and utilization in very short time, with three new versions published later [4-6], including the one which has been standardized. COSMIC Function Points – that represent a 2nd generation FSM method - are the result of the pursuit of the international group of experts in software functional size measurement, to find a measurement unit able to be successfully applied to the greatest possible number of software types and, specially, to real time software, where the application of the IFPUG unit is really hard.

The growing interest in industrial organization created by COSMIC can be verified by the growing number of projects included in the ISBSG repositories (one of the most important global repositories of data about software projects) grown from less than 50 projects in the 8th edition [16] up to 110 ones in the latest one (the 10th edition, January 2007) [17], with a growing rate higher more than 100% in five years; it can be also noted by the growing number of measurement experts certified in COSMIC, that also has grown more than 200% in the last 2 years; all these facts reflects the relevance of research activities on it.

This situation is the result of the contribution of three factors: first, its wide scope of applicability, since the unit can be used to measure very different kinds of software; second, the clearness of its concepts, making the unit easy to use and to learn how to use it; and finally, the low cost resulting from using this unit.

- FISMA FPA v1.1 Standard ISO/IEC 29881 [23]

Figure 1 presents an evolutionary view on the four afore-mentioned Function Software Measurement (FSM) units. A fifth method, FISMA FPA [23], will have been recently standardized.

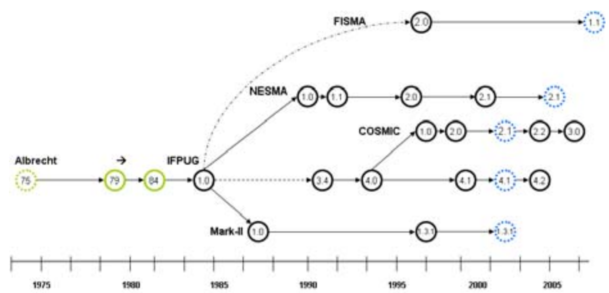


Fig. 1 – ***FSM Methods Evolution***

The paper is organized as follows: Section 2 analyzes main difficulties in obtaining the needed data for realizing this type of investigation, specifying the sources used for obtaining data in related works, and finally it describes the methodology used in this work in order to do it. Section 3 presents results obtained applying IFPUG analysis on those data. Section 4 shows the conclusions for this work and outlines future research issues. Section 5 proposes the list of references used. Moreover, an annex present the experimental data gathered and used in this work.

2. Data Gathering

In this section the different aspects of the Data gathering are treated in different subsections: The first one describes the data gathering methodology is stated and the main problems that the practitioners find when they try to obtain this kind of data; and the last section describes the new databases specifically obtained in this research.

2.1 Data gathering methodology and problems

The data gathering is one of the main problems, which comes from data gathering costs, since the wages of the specialists for this matter are usually high, up to 200 €/h in Europe. However, companies that already selected a specific measurement unit for their projects cannot easily see the return of investment derived from analyzing all the measures on the same software once again with a different measurer, considering the costs involved, and the problem multiplies if several measurers must to do the same measure. As consequence, data gathering in industrial environments is practically impossible.

The procedure used in to solve the problem related with its costs was applied; the data were derived as a final assignment in a measurement course on IFPUG on real, industrial applications.

In order to solve this problem there is basically only one solution: the researchers, coming from other institutions, usually a university or research centre, do the measures by themselves, using a set of software specifications provided by companies under agreements subscribed between both organizations with a group of students. Once they finalize their courses on software measurement studying these units, are asked to do a final project performing the required measurements with the units studied.

But in order to maintain the highest data quality possible the following steps were taken.

2.2. Samples generation

These steps were got during one academic year with the following results:

1. *Generation of the ifpug06 sample*

(measurement results are given in the Annex under ifpug06)

a. Starting from a total of 97 students, divided into 4 groups, 3 of them with 25 students per group and one with 22 students.
b. The final number of students with at least a 90% attendance was 89.
c. From these, 89 students enabled to participate in the measurement process; 86 passed the IFPUG part of the assignment, with the following distribution: 12 with more than 5/10 points, 52 with more than 7/10 points, 21 with more than 8/10 points, and 4 with more than 9/10 points.

As a consequence, 77 students were selected to participate in the IFPUG measurement process.

2. *Measurement of a real world application with the IFPUG method.*

Every selected student received the specifications of the same software application. The measure of the application was done by IFPUG certified measurers and was known by the researchers.

Each student should then proceed to measure the assigned application with both measurement units. And the students spent between one and one and a half hours for each labor day with a total between 10 and 15 hours to count the application.

3. Selection of the students participating in the measurements.

To participate in the measurement process it was required:

a. Each question had four possibilities with only one correct. If one question had a wrong solution this question did not count to the overall qualification but it had not a negative valor. Questions were corrected separately and the students were required to obtain a grade of 8 out of 10 in each part. The time that the students had to fill the task was 60 minutes.
b. To have a grade better than 7 over 10 in a written test of knowledge performed on the 11th week of classes. This test was applied to all students. Undergraduate students of the Systems Planning and Management course, given as part of Technical Engineering on Management Informatics degree, were trained in IFPUG 4.1 method. The versions used are the versions standardized by ISO. Even if they do not participate in the measurements.
c. To attend at least 90% of the classes, that is, students can only be absent in one class.

3. Data Analysis

Once generated the sample of the data collected, the first step of the analysis consisted of the definition of variables in IFPUG that were going to be analyzed, and the second one was the estimation of its statistics, the generation of frequency tables, the obtaining of frequency histograms and the calculation of the distribution that follows the sample to be able to do a quantitative analysis of the dispersion.

3.1 Defined IFPUG variable

The six main variables of IFPUG on which would take place the statistical study were indentified; ILF, EIF, EI, EO, EQ and the Total Unadjusted Function Points (UFP). Then, each of them are briefly described: [18]

a. IFPUG unadjusted Function Points, FP. The IFPUG unadjusted function points is the sum of the ILFs, EIFs, EIs, EOs and EQs, taking into account for each one of them its own function points based on its complexity.
b. External Input, EI. An external input (EI) is an elementary process that processes data or control

information that comes from outside the application's boundary. The primary intent of an EI is to maintain one or more ILFs and/or to alter the behavior of the system.

c. External Output, EO. An external output (EO) is an elementary process that sends data or control information outside the application's boundary. The primary intent of an external output is to present information to a user through processing logic other than or in addition to the retrieval of data or control information. The processing logic must contain at least one mathematical formula or calculation, or create derived data. An external output may also maintain one or more ILFs and/or alter the behavior of the system.
d. External Inquiry, EQ. An external inquiry (EQ) is an elementary process that sends data or control information outside the application boundary. The primary intent of an external inquiry is to present information to a user through the retrieval of data or control information. The processing logic contains no mathematical formula or calculation, and creates no derived data. No ILF is maintained during the processing, nor is the behavior of the system altered.
e. Internal Logic File, ILF. An internal logical file (ILF) is a user identifiable group of logically related data or control information maintained within the boundary of the application. The primary intent of an ILF is to hold data maintained through one or more elementary processes of the application being counted.
f. External Interface File, EIF. An external interface file (EIF) is a user identifiable group of logically related data or control information referenced by the application, but maintained within the boundary of another application. The primary intent of an EIF is to hold data referenced through one or more elementary processes within the boundary of the application counted. This means an EIF counted for an application must be in an ILF in another application.

3.2 Statistical analysis of variable

The statistics values obtained for IFPUG variable in the above sample were the average (Avg), median (Med), mode (Mod), standard deviation (S) and the minimum (MIN) and maximum (Max) values. It also gave the real value (AV) of each of the measured variables, and was obtained from the consensus of 3 certified IFPUG measurers. Then, the results for each of the samples are listed and analyzed.

	ILF	EIF	EI	EO	EQ	UFP
Avg	2,29	2,23	6,03	1,2	5,01	73,01
Med	2	2	6	1	5	71
Mod	2	2	6	1	5	68
S	0,6	0,78	0,73	0,87	2,36	10,22
Min	1	1	3	0	1	57
Max	5	7	9	6	12	104
AV	2	2	6	1	7	74

Table 1. Stats of central tendency and dispersal measurements from ifpug06 sample

It can be observed that the standard deviation to the variables ILF, EIF, EI and EO has a value close to zero, which indicates that most of the measures were well executed and close to the real value, with the exception, already known of the EQ. However, it can be observed that the maximum and minimum values are separated very significantly from the real value.

Sample ifpug06

If these obtained values of central tendency [*Table1*] are analyzed, it can be observed that the measures of central tendency (average, mode and median) are practically identical to almost all the variables (ILF, EIF, EI, EO), with the exception of external queries (EQ), and also coincide with the real value (AV). It would show that identifying these variables is an important source of horizontal dispersion. Against the variable EQ, statisticians of central tendency agree among themselves, but on the other side, they move away from real value. This could indicate a greater difficulty in identifying EQs with regard to the rest of variables.

Once analyzed the statisticians of central tendency of the variables, measures of dispersion [*Table1*] were analyzed, because if the measures of central tendency let to synthesize the data in a representative value, the measures of dispersion show to what extent these measures are representative as a synthesis of the information. The measures of dispersion quantified the separation, the dispersion and variability of the values of the distribution over to the central value, and this is the main objective of this research.

On the other hand, it analyzes the statistics of the total function points. It was observed that the value of the average is very close to the real value, but the mode shows that most of the measurers have a tendency to rely fewer points of that really are. This may be, on one hand to the impact of the incorrect measurement of EQs already identified; and on the other to an incorrect allocation of complexities to the identified variables.

It is interesting to know the distribution that follows the sample would provide a clear idea about how the data is distributed. In many probability distributions of quantitative variables is a trend of the values around the average and less values once upon it reaches the ends of the range. If the number of observations is more than 30, as happens in this study, according to the central theorem of the limit, the distribution takes approximately the shape of the Gauss´s Bell known as Normal distribution.

In order to analyze the shape of the dispersion of function data total points more in-depth, so that they could establish confidence intervals of the measures, once obtained the results specified above, the type of distribution that remained that variable in the samples.

In order to bring this, there were taken two steps, in the first one took place the histogram of frequencies, the result of which, with the normal curve, can be appreciated in *Figure2*. As it shows, the value of the total function points varies between 65 and 75 function points for most of the measures. However it can be observed, although in a minority share of the sample, values very disparate of real value of 74 totals function points as are the falling between 95 and 105 or less than 60.

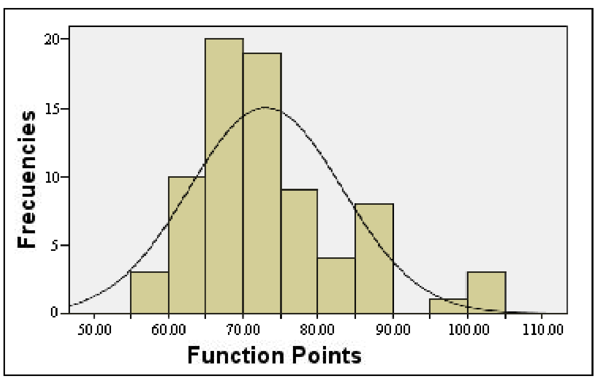


Fig. 2 –*Histogram of the frequencies from ifpug06 sample*

Thus, in a second step, to conclude whether, as can be seen visually in the histogram, the distribution of the totals function points in the ifpug06 sample conformed to a normal distribution, was conducted the test of Kolmogorov-Smirnov, which allows contrasting the shape of the distribution of each sample in a Normal distribution, of Poisson, Uniform and Exponentially. This will compare the role of accumulated observed data with the normal distribution, measuring the distance between the two curves. Table 3 shows the results obtained in the test.

The null hypothesis is rejected when the P-value is less than the level of significance set, which in this case has been p = 0,05. In the case of the sample being analyzed the p-value is 0,06 so that you cannot reject the null hypothesis and can therefore assume that the total data function points presented a Normal distribution.

N	77
Z de Kolmogorov-Smirov	1,3179071 2
Sig. asintot. (bilateral)	0,0619987 5

Table 3. Kolmogorov-Smirnov in ifpug06 sample

4. Conclusions and Future Work

The fundamental objective raised in this research was determine the margin of error that could be considered that is introduced by default on the measurements made with this unit. That is the analysis of the horizontal dispersion of the function points on an application measured by different measurers with the measurement units IFPUG.

After the analysis made along this article the following key findings were obtained, in chronological order:

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.

- The conclusion would relate to the identification of the main sources of error in the performance of the measures. Thus, for IFPUG unit we have found that it comes from the identification of the external queries, EQ and could come from the determination of the complexities of the functions identified.

- The horizontal dispersion of the measurements using the unit of measurement IFPUG conforms to a normal disperse, in such a way that in the interval

$$x_{av} \pm 0.18x_{av} \quad (2)$$

are located the 90 % of the measures taken. And in the interval

$$x_{av} \pm 0.23x_{av} \quad (3)$$

are located the 95 % of the measures taken.

Since they have done with measurers with low experience and it is logical to think that if the measures were undertaken with measurers with more experience the margins of error would be lower. The intervals become narrower with values of $x_{av} \pm 18\%$ for IFPUG for the 90% of the data. It can be appreciated that, for the 95% of the data, $x_{av} \pm 33\%$ for IFPUG could be considered that these would be the maximum limits of the horizontal dispersion of measurement.

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

- Implementation of new analysis on a sample obtained with experts measurers to experimentally test the conclusion that the dispersion that occurs in such sample is lower than that produced in a sample obtained with low expert measurers.
- 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.

References

[1] Albrecht A. J., "Measuring application development productivity," en Proc. Joint SHARE, GUIDE, and IBM Application Development Symp., 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 Meaurement Manual 2.0"

[4] Common Software Measurement International Consortium, "COSMIC-FFP Meaurement Manual 2.1"

[5] Common Software Measurement International Consortium, "COSMIC-FFP Meaurement Manual 2.2"

[6] Common Software Measurement International Consortium, "COSMIC-FFP Meaurement Manual 3.0"

[7] 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.

[8] Ho, V.T.; Abran, A.; Fetcke, T., "A Comparative Study Case of COSMICFFP, Full Function Point and IFPUG Methods," Département d'informatique, Université du Québec à Montréal, Canada.

[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"

[13] International function points users group, "Function points counting practices manual 4.1.1"

[14] International function points users group, "Function points counting practices manual 4.2"

[15] International function points users group, "Function points counting practices manual 4.2.1"

[16] International Software Benchmarking Standards Group repository, Release 8. http://www.isbsg.org/

[17] International Software Benchmarking Standards Group repository, Release 10. http://www.isbsg.org

[18] ISO/IEC 20926: 2003, Software engineering IFPUG 4.1 Unadjusted functional size measurement Method. Counting practices manual International Standardization Organization, ISO, Ginebra, 2003.

[19] ISO/IEC19761:2003, Software Engineering COSMIC-FFP. A Functional Size Measurement Method, International Standardiz ation Organization, ISO, Ginebra, 2003.

[20] C. Symons, "Function Point Analysis: Difficulties and Improvements," IEEE Transactions on Software Engineering, vol. 14, no. 1, pp 2-11.

[21] Vogelezang, F., Lesterhuis, A., "Applicability of COSMIC Full Function Points in an administrative environment: Experiences of an early dopter," en: Proccedings of the 13th International Workshop on Software Measurement, Shaker-Verlag, Montréal, Canada, Sept. 22-25, 2003, pp. 232-243.

[22] 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/

[23] 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

[24] 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", International Standardization Organization, Genève, 2005.

[25] ISO/IEC 20968:2002, Software engineering Mk II Function Point Analysis. Counting Practices Manual, International Standardization Organization, ISO, Genève, 2002.

[26] NESMA, "Definitions and counting guidelines for the application of function points analysis. A practical manual 1.0", Nederlandse Software Metrieken Associatie.

[27] NESMA, "Definitions and counting guidelines for the application of function points analysis. A practical manual 1.1", Nederlandse Software Metrieken Associatie

[28] NESMA, "Definitions and counting guidelines for the application of function points analysis. A practical

manual 2.0", Nederlandse Software Metrieken Associatie

[29] NESMA, "Definitions and counting guidelines for the application of function points analysis. A practical manual 2.1", Nederlandse Software Metrieken Associatie

[30] NESMA, "Definitions and counting guidelines for the application of function points analysis. A practical manual 2.2", Nederlandse Software Metrieken Associatie

[31] Symons C., "Function Point Analysis: Difficulties and Improvements," IEEE Transactions on Software Engineering, vol. 14, no. 1, pp 2-11.

[32] Desharnais, J.M., Abran, A. and Cuadrado-Gallego, J.J. "Convertibility of Function Points to COSMIC: Identification and analysis of functional outliers" Proceedings of the 17th International Workshop on Software Measurement, Shaker-Verlag, Palma de Mallorca, Illes Balears, España, Nov. 5-8, 2007, pp. 130-146.

[33] Abran, A., Desharnais, J.-M., Azziz, F., "Measurement Convertibility: From Function Points to COSMIC-FFP" Proceedings of the 15th International Workshop on Software Measurement, Shaker-Verlag, Montréal, Canada, Sept. 12-14, 2005, pp. 227-240, URL: http://www.gelog.etsmtl.ca/publications/pdf/906.pdf

Annex: Project's Data

Sample *ifpug06 (n=77)*

N	ILF	EIF	EI	E0	EQ	FP
1	3	2	6	1	9	87
2	3	1	3	3	4	72
3	3	1	5	1	2	58
4	2	2	6	0	6	67
5	2	3	6	1	2	72
6	2	2	6	1	5	66
7	3	2	6	0	3	104
8	2	2	6	1	8	74
9	2	2	6	2	2	57
10	2	3	6	1	4	74
11	2	2	6	1	5	62
12	4	2	6	3	10	104
13	2	2	6	1	5	67
14	2	2	6	1	5	63
15	3	2	7	1	3	69
16	2	2	7	1	4	66
17	2	2	6	1	8	75
18	2	2	6	1	7	71
19	2	3	6	0	3	65
20	2	3	6	1	2	72
21	2	2	6	1	6	70
22	2	3	6	1	7	75
23	2	2	7	1	2	61
24	3	2	6	0	5	80
25	3	2	6	0	5	80
26	2	2	6	1	5	69
27	2	2	7	1	3	63
28	3	2	6	1	7	85
29	2	3	6	1	3	84
30	5	7	5	0	4	97
31	2	2	6	1	5	74
32	2	2	6	1	5	63
33	2	2	6	1	5	69
34	2	2	6	1	2	61
35	2	2	5	1	7	69
36	3	3	6	1	2	85
37	2	2	6	2	5	68
38	2	2	6	1	5	68
39	2	2	6	0	6	65
40	2	2	6	1	7	78
41	2	2	8	2	4	76
42	2	2	6	1	2	57
43	2	2	6	1	6	71
44	2	2	6	1	4	62
45	2	2	6	2	2	63
46	1	2	6	1	5	61
47	2	2	6	1	7	72
48	2	2	6	1	5	66
49	2	2	6	1	4	71
50	2	2	6	1	5	68
51	2	2	7	1	6	78
52	2	2	6	0	7	68
53	2	2	6	1	8	77
54	2	2	6	3	3	68
55	2	4	6	2	3	85
56	3	2	6	1	8	87
57	3	2	6	2	3	75
58	2	2	6	1	12	89
59	2	3	6	2	1	70
60	2	2	6	1	12	89
61	2	2	6	1	7	74
62	2	2	6	1	2	65
63	2	2	7	2	5	71
64	3	2	7	1	3	75
65	3	2	6	0	3	62
66	2	3	6	1	4	68
67	3	2	6	1	5	72
68	2	3	9	1	4	82
69	2	2	6	2	6	76
70	2	2	6	2	4	67
71	2	5	4	1	4	69
72	2	2	6	1	8	74
73	2	2	6	1	8	74
74	4	2	6	3	10	103
75	3	2	6	2	6	85
76	2	2	4	6	2	72

7 7	3	2	6	1	5	71

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