IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 6, Ver. II (Nov. - Dec. 2016), PP 48-53
www.iosrjournals.org

Evaluation of Convertibility Issues Between IFPUG and Cosmic Function Points

Md. Forhad Rabbi[1], Shailendra Natraj[2], Olorisade Babatunde Kazeem[2]
[1]*Curtin University Australia*
[2] *Universidad Politechnica de Madrid SPAIN*

Abstract*: Software industry has matured with time, from small application of few lines of codes to software application of millions of lines of code. In the past few years, the concern of the industry regarding software size estimation has been the convertibility issue between the International Function Point User Group (IFPUG) and the COmmon Software Measurement International Consortium (COSMIC) in order to leverage their huge investment on the IFPUG. Since there is still no cost and effort estimation tool for COSMIC function points. IFPUG is one of the early estimation methods, however, with the introduction of a more scientific method like COSMIC which has a wider applicability than the IFPUG and both method using the same measuring unit and principle, the continued relevancy of the IFPUG is called to question. Due to similar underlining principle of the two methods and for organizations that have invested so much in the IFPUG not to lose all their investment because of migrating to using COSMIC, researchers have been trying to explore the possibility of converting the output of one method to the other. This paper reviews some of the popular conversion formulas that have been suggested so far to see a trend or how related, consistent and reliable the formulas could be. We estimate the function point of two case studies using the COSMIC and IFPUG. Then we insert our estimation result into the formulas to see how close or diverse the output will be in comparison with our calculation. The result varied widely and nothing conclusive can be said, though, two of the formulas give closer estimation range than others. We also highlight why COSMIC may be more desirable today than the IFPUG and presented the progress level on trying to establish a convertible relationship between the two methods.*
Keywords: *IFPUG, COSMIC, Project Estimation*

I. Introduction

Software industry has matured with time, from small application of few lines of codes to software application of millions of lines of code. Nowadays, developed software are grandly scalable and more robust with high usability features; this results in software of immense size and complexity, unfortunately, the early estimation used for these projects are often proved incorrect in the later part of the projects lifecycle. In recent years, the major concern of many large companies has been the project estimation as software cost estimates are critical to both developers and customers in terms of cost and effort and there is no panacea technique available in industry that guarantees 100Nevertheless, there are few project estimation techniques available in industry that helps in making the estimation quite optimal and worth the effort. Function points have been a means to determine the software size since the early 70s [6]. The IFPUG is the most traditional and widely used method for calculating the function size of a software project, however, different function point analysis (FPA) technique have evolved over the years; all aiming to correct the lapses of the IFPUG. Only four of them have matured to ISO measurement standard level - ISO 19761: COSMIC FFP [9], ISO 20926: Function point analysis, e.g., IFPUG 4.1 [12], ISO 20968: Mk II [11] and ISO 24570: NESMA [10] Most of these methods are designed to measure specific application type except for the COSMIC. Based on its wide applicability and the fact shown in studies [], that size estimation from COSMIC results in better effort and cost projection than that of the IFPUG. COSMIC FFP is rapidly replacing the IFPUG in the industry. Despite this fact, there are no tools yet to calculate effort and cost from the COSMIC function size while there are tools like the COCOMO II for effort and cost estimation from the results of the IFPUG. In view of this, researchers have been working on establishing a relationship model between the function points (FP) as calculated from IFPUG and COSMIC to satisfy the request of COSMIC users who have invested immensely in IFPUG data. Several diverse models [6][8] have been proposed, but there has been no work to harmonize the diversity among these models or establish the area of applicability of each. Therefore, the industry is still lost as to what method to use in converting COSMIC FFP to IFPUG FP. The aim of this paper is to investigate existing proposed conversion models (formulas) and see if any of them has the potential to emerge as unique and application independent. This paper is organized as follows: a cursory look at related work is presented in Section II and approach to the study in Section III. Section IV and V presents the detailed calculation and result of measuring the two case studies with IFPUG and COSMIC respectively. Section VI analyzes the result of the study while conclusion and future work is the focus of Section.

II. Background and Related Work

The IFPUG FPA is a very solid method for measuring the software functional requirements. No method in this realm lasted as much as it does almost three decades and gain as much wide acceptance [3]. However, the software engi-neering has recorded several progressive changes, software is being developed in a way not envisaged by Albrecht when describing the IFPUG in 1979 [3] [7]. For example, today we see systems developed from composition of several components instead of being developed as whole, event-driven embedded systems, web-based systems and a hybrid of all. The IFPUG uses a nominal scale unit, therefore, little or nothing can be said while comparing the function points of different projects. A project with 200FP cannot be said to be twice as big or complex as another one with 100FP. Thus, in the absence of an estimation tool, historical data of function points has little to offer in any new projection.

These and a few other problems partly stimulated our desires to focus this study on COSMIC as it has solution to some of the problems encountered in the use of IFPUG. The COSMIC FPA is more widely applicable to modern paradigm of system development than the IFPUG. Neverthe-less, a lot of organizations have invested hugely on IFPUG and it is hard and unwise to scrap such investments over-night. Therefore, this study is focused to further explore the relationship that may exist between the outputs of the two methods for easy transferability. The motive behind function size measurement is to have an idea of the projects size to make approximate project cost and effort estimation, early in the projects lifecycle. It is still currently difficult to make such estimation from the result of the COSMIC FP, as there are no direct conversion guidelines or tools. Organizations still need to rely on historical data from IFPUG FP (which has tools like COCOMO) to be able to translate COSMIC FPs to effort and cost; More so that several organizations have worked with IFPUG estimation for years. There have been many attempts to translate COSMIC to IFPUG, yet no definite consensus. Though, it should be noted that some researchers opine that a universal conversion formula to convert COSMIC FFP size to an IFPUG FPA size would be grossly misleading [2]. This is because COSMIC uses some measurements that are hard to equalize to IFPUG e.g. layer measurement or viewpoint measurement, e.g. developer or machine viewpoint. Thus, there is no direct correlation between the two techniques.

The only means of conversion therefore will be to measure some sizes of software developed or enhanced in similar circumstances on both the IFPUG and COSMIC methods and to establish an empirical conversion formula. This approach may take time to mature and may have to be peculiar to organizations at the beginning. This is exactly what researchers are doing but no trend is established yet in all the formulas available. The studies reported here all measure function points from end user viewpoint and did not use value adjustment factor (VAF) in conformity with ISO 14143-1 and 20926[6]. Each of the studies suggested a conversion formula between IFPUG and COSMIC. For the formulas, Cfsu denotes COSMIC FFP while UFP denotes Unadjusted function points. Except otherwise mentioned, all these formulas are derived using linear regression model. In 1999, Fetcke measured four applications of a data storage

system [6]. At the end of his study, he proposed a convert-ibility formula:

Y (Cfsu) = 1:1 (U F P) 7:6 (1)

The same year, Abran [6] proposed:

Y (Cfsu) = 1:0 (U F P) 3 (2)

Vogelezang on the other hand in 2003 used the same ap-proach on eleven projects of Rabobank initially measured with NESMA and proposed the convertibility formula [5]:

Y (Cfsu) = 1:2 (U F P) 87 (3)

The constant was attributed to the contribution of logical files. This formula was also found to be applicable to big projects with FP 200. This led to the proposition of another formula suitable to smaller projects with FP ¡ 200:

Y (Cfsu) = 0:75 (U F P) 2:6 (4)

Also, in 2005, a new convertibility formula was suggested by Desharnais using dataset from the documentation of finished projects of a governmental organization [6]:

Y (Cfsu) = 0:84 (U F P) + 18 (5)

In 2007, Cuadrado et al proposed a theoretical formula which he also verified empirically on 33 software applications. In the
formula, he tried to eliminate the effect of logical files

which were identified as the source of the offsets of earlier formulas from linear regression [8]:

EI EO EQ
X **X** **X**i
max***(2***:F T Ri ***+ 1) +*** max***(2***:F T Ri ***+ 1) +*** max***(2***:F T Ri ***+ 1)***
i=***0*** i=***0*** ***=0***

CF SU (6)

EI EO EQ
X **X** **X**i
max***(2***:***2***F T Ri ***+ 1) +*** max***(2***:***2***F T Ri ***+ 1) +*** max***(2***:***2***F T Ri ***+ 1)***
i=***0*** i=***0*** ***=0***

III. Research Methodology

Our research work is based on two steps. In first step we have done qualitative part of our research and in the second step following first step, we have done an experiment as a quantitative approach. This quantitative method is based on the outcomes of qualitative approach.

A. Qualitative Method

For our research work, we have performed literature review including IFPUG and COSMIC manual review. Depth knowledge about those two tools has been achieved through this technique. Beside, to analyze our results after using IFPUG and COSMIC, we have gone through some research papers related with IFPUG and COSMIC. We have used Google Scholars search engine, ACM, IEEE database for source of our information.

B. Quantitative Method

In the quantitative part of our research work, we have calculated functional point using both IFPUG and COSMIC tools on PC GEEK and Locator. In this experiment, we have defined requirements for the desired application at first then use those functional point calculation tools to calculate functional points.

IV. Application of Ifpug

A. IFPUG Tool

The PC GEEK project used as case study is a small business application while the Locator is an event-driven application. We used the IFPUG tool to calculate the func-tional point from the requirements. The IFPUG has two main basic functional components (BFC) types transactional and data functions. So, according to the given specification, the elementary processes of the case project are identified and classified into BFC types in subsections below.

1. External Input
2. External Output
3. External Query B. Data Function

Data functions are classified into two types, Internal log-ical file (ILF) and external interface file (EIF).

1) Internal Logical File
2) External Interface File

C. Case Study 01:Pc Geek

1.System Requirements: The requirements contained in this case study are based on the companys process descrip-tion, supplied by the customer. These requirements will be used in the whole development process, which contains the cost estimation that is covered in this document as well as the development of the system and the delivery and acceptance of the developed system. In order to estimate the exact requirement, we first drew the borderline between the various systems so that we could estimate each of these systems separately. The recognized separated systems here are as below:

1) Warehouse Management
2) Customer Management
3) Supplier Purchasing Management
4) Goods Reception Management
5) Assemblies and Manufacturing Management
6) Shipment Management D. Points Calculation

Function size measurement is an important way to esti-mate the software size. And it is a usable tool for early estimation in software projects. The earliest technique for this measurement is the function point calculation using the IFPUG approach. In the following section, we will use this technique to calculate the function points in the PCGEEK system and Simple Locator.

E. Result

According to IFPUG Method, the functional point for PCGEEK project is 154 (Calculation is in Appendix A1)

1) CASE STUDY 02: SIMPLE LOCATOR: The trans-actional function is divided into three in order to properly identify each transaction type.

F. External Input (EI)

There are five elementary processes classifiable under this BFC type:
1) Create company contact
2) Update company contact
3) Delete company contact
4) Packet sent
5) Phone contact completed G. External Output (EO)

None of the processes is classifiable under this type as it must contain outputs of algorithm, calculation or derived data.

H. External Inquiry (EQ)

Three elementary processes fall under this type:
1) Retrieve company contact
2) Display data
3) Return error I. Data Functions

The data function is also classifiable into two.

Internal Logical File (ILF)
1) Company contact file
2) Packet detail file

External Interface File (EIF) This has only one file:
1) Error file

J. Result

According to IFPUG, the FP for simple locator case is 46. (Calculation is in Appendix 1.2)

V. Application of Cosmic

A. COSMIC Tool

According to COSMIC Method V3.0 measurement manual [1] COSMIC method is very much suitable for business ap-plication software to measure functional size. Large amount of data about event can also be managed by this method. COSMIC has an advantage to measure component of any size. Moreover, this COSMIC is ISO certified and it is very easy to learn and apply to measure the software size.279

B. Result: PCGEEK Project

According to COSMIC method v3.0, for any functional process, the functional sizes of individual data movements shall be aggregated into a single functional size value in units of COSMIC Functional Point (CFP) by arithmetically adding them together [2].

Size (functionpoint) = (Entries) + size (Exits) + size(Reads) + size(W rites) [2],

Size(F unctionP oint) = size(1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1) + size(1 + 1 + 1 + 1 + 1 + 1
+ 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1
+ 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1) + size(1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1) + size(1 + 1
+ 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1) = 97

C. Result: Simple Locator

COSMIC process is based on Functional User Require-ment (FUR). This FUR can be decomposed in more func-tional process. There are four types of data movement related with one data group. Later, COSMIC Functional Point (cfp) is calculated summarizing each data movements.

Size (functionalprocess) = (1+1+1+1+1+1)+(1+1+ 2+1+1+1)+(0+0+1+0+0+0)+(1+1+0+1+1+1) = 6 + 7 + 1 + 5 = 19 cfp

So, the functional size of the product, simple locator application, is 19 COSMIC Functional Points (cfp).

VI. Analysis of Case Studies

For the PCGEEK case study, using COSMIC, the FP is measured to be 97 and 25 for the simple locator, as shown in Section IV. The measured IFPUG values is inserted into the formulas, the result is shown in table 1; the Letherthius and the Cuadrados formula seem to give a better approximation as our calculation result. Both formulas gave a single digit percentage difference (5) in case of PCGEEK and also gave the smallest approximation in the locator though in this case 28% and 24% respectively. The problem with the locator

We hope to explore this further probably it will be possible to say some things on the correctness or applicability of each or any of the formulas. Also, it is impossible for us to use this result directly in a tool like COCOMO, since estimation with COSMIC is still at specific organizations level and does not have factors like EIF detail required by COCOMO. Despite this, if an organization is able to establish its own estimation model, experience has shown that output from COSMIC gives better cost and effort estimation than output from IFPUG [5]. Though, equation 6 holds some promises as a useful application independent formula desired. This may be because it was developed theoretically independent of any application before being empirically tested. This claim needs further verification to establish. Generally, more work still need to be done in establishing an application independent convertibility relationship between the IFPUG and the COSMIC for the use of organizations that have expended so much on IFPUG and are willing to migrate to COSMIC. Also, there is need for the creation of a good effort and cost estimation tool like COCOMO for COSMIC users.

VII. Conclusion

This paper examined the two most widely used function size measurement methods, COSMIC and IFPUG. The main was to present the progress level on effort to establish a convertible relationship between the two methods. Also, existing formulas were evaluated to see if any of them has the ability of being application independent. We presented six conversion formulas and review their consistency by applying them on the FP results of two projects PCGEEK and Locator we calculated. The result varied widely and nothing conclusive can be said, though, two of the formulas give closer estimation range to our calculation results than others. More research is still required to establish a unique convertibility relationship between IFPUG and COSMIC independent of any application and also in the creation of a cost and effort estimation tool for COSMIC users.

Table I: Application of Various Formulas to the Case Study

Authors	PCGEEK(97)	%Difference	Locator(25)		%Difference
Fetcke	166	71.0	43		72.0
Letherthuis	102	5.0	-38		NA
200F P					
Letehrthuis	116	19.5	32		28.0
200F P					
Desharnais 2005	150	54.6	57		128.0
Abran [1999]	155	59.8	43		72.0
Cuadrado et al.	77 97 99	2.0	19	19 25	24.0
[2007]					

system may be due to size or application type. We did not have enough project calculations to be able to make any reasonable claim on any of the formulas. Few factors like project size, project type, consistency in calculation pattern (IFPUG and COSMIC) need also be taken into consideration.

References

[1] Everald E. M., Software Metrics: SEI Curriculum Module SEI-CM-12-1.1, Software Engineering Institute, Carnegie Mellon University. December 1988.

[2] The COSMIC Functional Size Measurement Method Version 3.0, Measurement Manual, Common Software Measurement International Consortium, September 2007.

[3] Vogelezang F. W., COSMIC Full Function Points The next generation of functional sizing Software, Measurement European Forum SMEF 2005, January 28-30, Rome (Italy), 2005.

[4] Abran, A., Symons, C., Oligny, S., An Overview of COSMIC-FFP Field Trial Results, 12th European Software Control and Metrics Conference ESCOM 2001, April 2-4, London (England), 2001.

[5] Vogelezang F. W., Implementing COSMIC-FFP as a replace-ment for FPA, International workshop on software measurement, IWSM/MetriKon 2004.

[6] Abran A., Cuadrado J., and Desharnais J. M., Convertibility of Func-tion Points to COSMICFFP: Identification and Analysis of Functional Outliers, MENSURA 2006, Cadiz (Spain), Nov. 6-8, 2006.

[7] Gencel C. and Demoror O., Functional Size Measurement Revisited, ACM Transactions on Software Engineering and Methodology, Vol. 17, No. 3, Article 15, June 2008.

[8] Cuadrando-gallego J., machado-Piriz F. and Aroba-Paez J., On the conversion between IFPUG and COSMIC software function size units: a theoretical and empirical study, The journal of systems and software 81, 661-672, 2008.

[9] ISO/IEC19761:2003, Software Engineering – COSMIC-FFP A Func-tional Size Measurement Method, international Organization for Stan-dardization - ISO, Geneva, 2003.

[10] ISO/IEC 24750: 2005, Software engineering – NESMA functional size measurement method version 2.1 – Definitions and counting guidelines for the application of Function Points Analysis, International Organi-zation for Standardization – ISO, Geneva, 2005.

[11] ISO/IEC 20968: 2002, Software engineering – Mk II Function Point Analysis – Counting Practices Manual, International Organization for Standardization – ISO, Geneva, 2002.

[12] ISO/IEC 20926: 2003, Software engineering – IFPUG 4.1 Unadjusted functional size measurement method – Counting Practices Manual, International Organization for Standardization – ISO, Geneva, 2003.

APPENDIX: **A1.1**

FP Calculation for PCGEEK Project:

FP Calculation for PCGEEK Project				
Function Type	Complexity		Total Complexity	Total
EI	7	Low * 3	21	
	8	Avg * 4	32	
	0	High* 6	00	53
EO	4	Low * 4	16	
	0	Avg * 5	00	
	0	High* 7	00	16
EQ	9	Low * 3	27	
	2	Avg * 4	08	
	0	High* 6	00	35
ILF	5	Low * 7	35	
	0	Avg * 10	00	
	0	High* 15	00	35
EIF	3	Low * 5	15	
	0	Avg * 7	00	
	0	High* 10	00	15
			Total	154

APPENDIX: A1.2

FP Calculation for Simple Locator Project:

Function Type	Complexity		Total Complexity	Total
EI	4	Low * 3	12	
	1	Avg * 4	4	
	0	High* 6	0	16
EO	0	Low * 4	0	
	0	Avg * 5	0	
	0	High* 7	0	0
EQ	3	Low * 3	9	
	0	Avg * 4	0	
	0	High* 6	0	9
EIF	0	Low * 5	0	
	1	Avg * 7	7	
	0	High* 10	0	7
ILF	2	Low * 7	14	
	0	Avg * 10	0	
	0	High* 15	0	14
			Total	46