Utilizing Functional Size Measurement Methods for Real Time Software Systems

Cigdem Gencel
cgencel@ii.metu.edu.tr

Onur Demirors
demirors@ii.metu.edu.tr

Erhan Yuceer
Erhany@AYESAS.com

Abstract

There are various approaches developed for software size measurement. After Albrecht developed his original method which is based on quantifying functionality delivered to the users in 1979, variations of Functional Size Measurement (FSM) methods have been developed and become widely used. Among those, IFPUG FPA, Mk II FPA, COSMIC FFP and NESMA FSM have become international ISO standards being conformant to ISO/IEC 14143. This paper presents the results obtained by applying Mk II FPA and COSMIC FFP to two real-time software systems which have control as well as algorithmic components. The evaluation of these methods with respect to their measurement processes and the difficulties faced during the measurement process are also discussed.

1. Introduction

Reliable size measurement of software systems is still one of the significant challenges of software engineering. Until today, various approaches for sizing software have been developed. The metrics and methods based on "functionality" have become widely-used after originally introduced by Albrecht in 1979 [1], [2]. After that, variants of the method have been developed. During the 1980s and 1990s, several authors have suggested new FP counting techniques that intended to improve the original FPA or extend its field of application [3]. In 1996, the International Standards Organization (ISO) started to work on Functional Size Measurement (FSM) to establish common principles of those methods and published ISO/IEC 14143-1 standard on FSM [4]. Detailed descriptions of four methods; Mk II FPA [5], IFPUG FPA [6], COSMIC FFP [7] and NESMA FSM [8] are recently published as international standards that are certified by ISO.

There are a number of studies on the evaluation and comparison of the FSM methods in the literature. Rule discusses the similarities and differences between IFPUG FPA and Mk II FPA in his study [9]. In [10], Lother and Dumke evaluated FSM methods with respect to their suitability for certain functional domains and their maturity, and discussed the issues of FSM. In another study [11], three estimation methods applied early in the life cycle to a case project, are compared. In [12], Rollo discusses the issues associated with sizing web applications and evaluated IFPUG FPA, Mk II FPA and COSMIC FFP. Mk II FPA and COSMIC FFP methods are compared and the differences between them are discussed in another study [13].

In this paper we presented the results obtained by applying Mk II FPA and COSMIC FFP to two development projects of real-time applications. First application is one of the subsystems of an avionics management system for small to medium size commercial aircrafts. The other one is Collision Avoidance System (CAS) component of a Traffic Alert and Collision Avoidance System. Among other methods which are applicable to measure the size of real-time software such as Feature Points [14], 3-D Function Points [15], Analytical Software Size Estimation Technique-Real-Time [16] and Full Function Points (FFP) [17], we selected Mk II FPA and COSMIC FFP methods for being international ISO standards and having detailed measurement manuals, which are required in order to make reliable measurement.

The descriptions of the projects, the application of Mk II FPA and COSMIC FFP to these projects and the results obtained are presented in the second section. Last section discusses the results of this study, the difficulties faced during the measurement process, and the evaluations of the methods.

2. Application of the Methods to Projects

Both projects are developed by the same organization, which is a SW-CMM Level 3 company.

Project-1 is a development project of one of the subsystems of an avionics managements system for small to medium size commercial aircrafts on a Flight Display System. It is developed according to RTCA/DO-178B Software Considerations in Airborne Systems and Equipment Certification and will be certified by Federal Aviation Administration. The software complies with DO-257A, 'Minimum Operational Performance Standards for the Depiction of Navigation Information on Electronic Maps' as a basis and additional user requirements are integrated. The project was started in November 2003 and expected to be completed in September 2005. The coding phase is completed and the testing phase has been continuing. The project staff consisted of 1 project manager, 1 senior software engineer (development team leader), 6 software engineers (development team), 1 senior software test engineer (test team leader), 2 junior software test engineers (test team), 1 software quality engineer and 1 software configuration management specialist.

Project-2 is a Collision Avoidance Subsystem (CAS) provides the collision avoidance functionality for the TCAS system. CAS functionality is specified in detail in the CAS Requirements Specification (DO-185A, volume 2). In this study, we measured the size of CAS-Own Aircraft Algorithm. Own Aircraft function determines the TCAS operational mode, effective sensitivity level and other operation parameters used by the collision avoidance logic. The project was started in September 2004 and expected to be completed in July 2005. The project staff consisted of 1 project manager, 1 senior software engineer (development team leader), 3 software engineers (development team), 1 software test engineer (test team leader), 1 junior software test engineer (test team), 1 software quality engineer and 1 software configuration management specialist.

For both projects, the efforts were collected on a daily basis in 0.25 hour intervals for each work breakdown structure task. The efforts utilized for the life cycle processes of the projects are given in Table 1.

Table 1.Efforts utilized for Project-1 & Project-2

Software Development Life Cycle Phase	Project-1 Effort (person-hours)	Project-2 Effort (person-hours)
Development	17,453.50	2,199.75
Management	2,316.25	1,418.00
Training	1,437.50	179.50
Supporting	2,351.00	1,986.00
Total	23,558.25	5,783.25

2.1. The Results of the Application of the Methods

For size measurement, we used the software requirements specification document of Project-1 and Project-2, which involves 758 Functional User Requirements (FURs) and 158 FURs, respectively. Two estimators involved in the size measurement process. One of the estimators has the domain knowledge who is also one of the project managers of Project-1 in the development organization. Although both of the estimators are experienced in using the methods, they are not certified by UKSMA and COSMIC.

The functional size of Project-1 and Project-2 are measured as 4,513.20 Mk II FP and 1,120.54 Mk II FP, respectively (see Table 2). The effort utilized to make measurement is 61.38 person-hours for Project-1 and 54.50 person-hours for Project-2. By applying COSMIC FFP, the functional sizes of Project-1 and Project-2 are measured as 3,408.00 Cfsu and 879.00 Cfsu, respectively (see Table 3). The effort utilized to make COSMIC FFP measurement is 50.38 person-hours for Project-1 and 12.50 person-hours for Project-2.

In Table 4, the ratio of SLOC to functional sizes obtained by COSMIC FFP and MkII FPA are given.

Table 2. Size measurement details of the projects by Mk II FPA

Project Name	Number of Logical Transactions	Number of Input Data Element Types (DETs)	Number of Output DETs	Number of Data Entity Types Referenced	Functional Size (MkII FP)
Project-1	456	670	2,686	2,064	4,513.20
Project-2	99	273	119	561	1,120.54

Table 3. Size measurement details of the projects by COSMIC FFP

Project Name	Number of Functional Processes	Number of Entries	Number of Exits	Number of Reads	Number of Writes	Functional Size (Cfsu)
Project-1	456	337	812	1,991	268	3,408.00
Project-2	99	177	48	556	98	879.00

We compared SLOC to functional size ratios with the data in ISBSG dataset. In ISBSG dataset, there are 15 new development projects, which are measured by COSMIC FFP and SLOC of which also exist. The application types of all these projects are Management Information Systems. The average SLOC value is 722.8 with a minimum 100 and a maximum 1425. The average SLOC/size ratio is 5.534 with a minimum value 1.04 and a maximum 18.75. Unfortunately, we could not compare our results for Mk II FPA since the projects which are measured by Mk II FPA have no associated SLOC values in ISBSG dataset.

Table 4. Functional size and SLOC ratios

Project Name	Size (MkII FP)	Size (Cfsu)	SLOC (Total)	SLOC/MkII FP	SLOC/Cfsu
Project-1	4,513.20	3,408.00	20,076.00	4.4	5.9
Project-2	1,120.54	879.00	3,896.00	3.5	4.4

3.Discussion of the Results

In this study, our aim is not to compare and judge which method is more suitable to measure functional size of real-time software systems. Rather, we want to evaluate these results in order to shed light on their improvement opportunities.

Since Mk II FPA and COSMIC FFP use different metrics, the functional sizes obtained by Mk II FPA and COSMIC FFP are not directly comparable with each other. In order to compare, a conversion of Mk II FP size estimate to COSMIC size estimate need to be performed. The designers of COSMIC FFP stated that an 'average conversion' formula would result a project to be under-sized or over-sized. We compared the results of both methods according to the base counts in order to depict what kind of factors give rise to obtain different functional sizes.

In Mk II FPA, the size of the processing component of a LT is defined to be proportional to the number of referenced entity-types. An entity-reference in Mk II FPA is generally equivalent to a Read or Write in COSMIC FFP [7]. Therefore, the sizes of the processing component are roughly equivalent on both scales. However, one of the rules of Mk II FPA is that each LT must have at least one input DET, must make one reference to a Data Entity Type and must have one output DET as a minimum. On the other hand, COSMIC FFP principles say that "a functional process comprises at least two data movements, an entry plus either an exit or a write". Therefore, for a specific LT, we should count at least 1 entity reference for the processing component in Mk II in case a file in persistent storage is neither read nor written. In COSMIC FFP, we do not to count anything related to the processing components if there is no Read or Write to persistent storage.

In Project-1, the number of references to Data Entity Types is 2,064 by MkII FPA and it is 561 in Project-2. By COSMIC FFP, the total number of data groups that are read or written is 2,259 in Project-1 and 654 in Project-2. The higher number in COSMIC FFP means there exist Data Entity Types which are both read and written since these entities are counted only once in Mk II FPA. Although we expect higher size by COSMIC FFP, the weight factor used in Mk II FPA changes the result. That is, when calculating the functional size of the processing component, we multiply the number of references by 1.66 in Mk II FPA whereas there is no weight factor in COSMIC FFP. The functional size of the processing component is 3,426.24 Mk II FP and 2,259 Cfsu for Project-1, 931.26 Mk II FP and 654 Cfsu for Project-2.

One of the main factors that cause the difference between the functional sizes of input and output components, obtained by these two methods is the level of granularity in each method's size measurement process. COSMIC FFP method estimates functional size at a higher level of granularity than Mk II FPA. The COSMIC-FFP unit of measurement, 1 Cfsu, has been fixed at the level of one data movement. On the other hand, in Mk II FPA method, the size of the input and output components of a LT is defined to be proportional to the number of DETs in the input and output components. The users of COSMIC FFP are warned to be careful when comparing the sizes of two different pieces of software where the average number of DETs per data movement differs sharply across the two pieces of software [7].

In Project-1, the number of input DETs is 670, the number of output DETs is 2,686 in Mk II FPA and the functional size of input and output components are 388.60 Mk II FP and 698.36 Mk II FP, respectively. By COSMIC FFP, the number of Entries is 337 and the number of Exits is 812 and the functional size of Entries and Exits are 337 Cfsu and 812 Cfsu, respectively. In Project-2, the number of input DETs is 273, the number of output DETs is 119 in Mk II FPA and the functional size of input and output components are 158.34 Mk II FP and 30.94 Mk II FP, respectively. By COSMIC FFP, the number of Entries is 177 and the number of Exits is 48 and the functional size of Entries and Exits are 177 Cfsu and 48 Cfsu, respectively. Thus, the factors discussed above result in a difference of 25 % between MkII and COSMIC functional sizes in Project-1 and 22 % in Project-2. Therefore, the practitioners should be careful when making effort estimation from functional size figures. The size unit of measure is very important when using average ratios of SLOC to functional size.

During the measurement process of Project-1, one of the significant difficulties we faced was identifying the elementary components of FURs, which are LTs in Mk II FPA and Functional Processes in COSMIC FFP. Generally, a FUR consists of one or more LTs or Functional Processes. In the measurement manual of COSMIC FFP, it is stated that "A functional process is derived from at least one identifiable FUR" [7]. In order to identify LTs or Functional processes, FURs are decomposed into their elementary components. However, in our case, we needed to gather and group two or more FURs in order to form one LT or one Functional Process. The total number of FURs is 758 whereas the total number of LTs (or Functional processes) is 456. This is due to the fact that the FURs in the software requirements specification document are very detailed, i.e. functional transactions are specified and the related requirements which would form one LT (or Functional process) were organized such that each is specified in different parts of the document. As a result one quarter of the total effort on Mk II FPA measurement was utilized for this purpose.

The effort utilized to make the measurement by Mk II FPA is 61.38 person-hours whereas it took 50.38 person-hours to make COSMIC FFP measurement in Project-1. In Project-2; the effort utilized for Mk II FPA measurement is 54.50 person-hours whereas it is 12.50 person-hours for COSMIC FFP measurement. Although it seems that we utilized greater effort for Mk II FPA measurement in both of the projects, we can not make such judgment since we measured both projects by Mk II FPA and COSMIC FFP consecutively. Therefore, we did not utilize extra effort to identify Functional processes in COSMIC FFP since we had already identified LTs in Mk II FPA.

Another difficulty was related to measuring the size of algorithmic manipulations and conditional statements. Neither Mk II FPA nor COSMIC FFP is designed to measure the size of these components. Project-2 involves intense algorithmic manipulations whereas both projects involve intense conditional statements. Thus, the size of these algorithms could not be measured by both methods.

As a result we can say that both methods can be used for measuring the size of real-time systems, but with some restrictions when algorithmic components exist. COSMIC FFP can be applied earlier in the development life cycle than Mk II FPA, since COSMIC FFP does not need the number of DETs in the input and output components. However, this requires that the average number of DETs does not vary across the BFC Types.

It is for sure that there still exist some improvement opportunities for FSM such as formulating conversion functions and measuring the algorithmic manipulations.

4. References

[1] A.J. Albrecht, "Measuring Application Development Productivity", in Proceedings IBM Applications Development Symp., Monterey, California, October 1979.

[2] A.J. Albrecht, and J.E. Gaffney, "Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation", IEEE Transactions on Software Engineering, vol. SE-9, no. 6, November 1983.

[3] C. Symons, "Come Back Function Point Analysis (Modernized) – All is Forgiven!)", Proc. of the 4th European Conf.on Software Measurement and ICT Control, FESMA-DASMA 2001, Germany, 2001, pp. 413-426.

[4] ISO/IEC 14143-1:1998 Information Technology - Software Measurement - Functional Size Measurement - Part 1: Definition of Concepts, 1998.

[5] ISO/IEC 20968, Software engineering - Mk II Function Point Analysis - Counting Practices Manual, 2002.

[6] ISO/IEC 20926, Software engineering - IFPUG 4.1 Unadjusted FSM Method-Counting Practices Manual, 2003.

[7] ISO/IEC 19761:2003: COSMIC Full Function Points Measurement Manual v. 2.2, 2003.

[8] ISO/IEC 24570:2005: Software engineering - NESMA Functional Size Measurement Method v.2.2 - Definitions and counting guidelines for the application of Function Point Analysis, 2005.

[9] G. Rule, "A Comparison of the Mark II and IFPUG Variants of Function Point Analysis" [Online], http://www.gifpa.co.uk/library/Papers/Rule/MK2IFPUG.html, 1999.

[10] M. Lother, and R.R. Dumke, "Points Metrics-Comparison and Analysis", Intern. Workshop on Software Measurement, IWSM'01, pp.155-172, Canada 2001.

[11] O. Demirörs, and Ç. Gencel, "A Comparison of Size Estimation Techniques Applied Early in the Life Cycle", Proc. European Software Process Improvement Conference (EurSPI), Springer Verlag Springer Lecture Notes in Computer Science (LNCS), 2004.

[12] T. Rollo, "Sizing e-Commerce", Proc. of the ACOSM 2000 Australian Conf. on Software Measurement, Sydney, 2000.

[13] C. Gencel, O. Demirors, E. Yuceer, "A Case Study on Using Functional Size Measurement Methods for Real Time Systems", 15th Intern. Workshop on Software Measurement-IWSM'05, Montreal, Canada, Sept. 2005, to be published.

[14] T.C. Jones, A Short History of Function Points and Feature Points, Software Productivity Research Inc., USA, 1987.

[15] S.A. Whitmire, "3D Function Points: Scientific and Real-time Extensions to Function Points", Pacific Northwest Software Quality Conference, 1992.

[16] D.J. Reifer, "Asset-R: A Function Point Sizing Tool for Scientific and Real-time Systems", Journal of Systems and Software, vol.11, no.3, pp.159-171, March 1990.

[17] A. Abran, D. St-Pierre, M. Maya, and J.M. Desharnais, "Full Function Points for Embedded and Real-Time Software", UKSMA Fall Conf., London (UK), October 30-31 1998.