Validation of Application Results of COSMIC-FFP to Switching Systems

Shin-ichi Nagano, Ken-ichi Mase, Yasuo Watanabe, Takaichi Watahiki, Shigeru Nishiyama
Research and Development Center,
NIPPON TELEGRAPH AND TELEPHONE EAST CORPORATION
s.nagano, y.watanabe, watahiki.t, nishiyama.s@rdc.east.ntt.co.jp,
k.mase@east.ntt.co.jp

Abstract

We applied the COSMIC-FFP method, the latest functional size measurement method, to 13 enhancement projects of our switching systems software ranging from 4 to 121CFSUs, the units of COSMIC-FFP. In this paper, we describe the motivation we had for the decision to try to apply COSMIC-FFP, issues on its application to switching software, its measurement characteristics such as speed of measurement, relationship between functional size in CFSU and costs. We have good results from the linear regression analysis of relationship between CFSUs and costs. Factors that affected the results are also analyzed.

1. Introduction

Telecommunication services come in various types, such as discount services, network services, and Internet-related services. We need to provide these users demanded services at the right time in order to survive under tremendous competition.

Generally, it takes time and cost to develop communication software. Accurate estimation of development time and cost beginning from a planning phase to delivery of services, therefore, is an important factor. The services are planned and really started to develop after judging cost effectiveness and it is always the most difficult task to make that judgment. For this judgment, early cost estimation is critical.

Conventionally, the LOC[1] method has been used in estimating the cost of telecommunications software. This kind of software is a typical example of real-time software, and LOC has been used only because there have been no effective and practical methods for estimation for real-time systems.

LOC, as the name implies, is concerned with the number of lines of code, and is determined after internal processing design is completed and implemented. With LOC, it is difficult to make an early estimate for cost effectiveness judgment of development work.

The need has consequently arisen for a method that can estimate the development time and cost of communication software from its requirements.

A method that measures the size of software functions can be used to the estimation of development cost. The most renowned method in this regard is the IFPUG[2] Function Point Analysis method. This method measures size of software based on input/output information. Actually, this method uses user requirements as it is information that can be seen from the user's perspective. Although it has a broad range of application, it is not particularly oriented toward systems that involve real-time operation and algorithms. The SPR method and the Full Function Points (FFP) method, on the other hand, can be applied to real-time systems. The Full

---

[1] LOC : Lines of code

[2] International Function Point Users Group

Function Points method underwent a revision in 1999 and is now known as COSMIC-FFP (COmmon Software Measurement International Consortium - Full Function Points).

Among the various methods, we thought that it was better to apply COSMIC-FFP as the functional sizing method for the following reasons.

- It can estimate from requirements.
- A real-time system can be estimated.
- Acquisition of the manual is easy.
- the technique is simple.

But, COSMIC-FFP is a new method and does not have a large record of achievements and we were uncertain as to whether to use it in practice or not. We therefore decided to evaluate the method to determine whether it could be used in practice.

This evaluation was carried out from the viewpoint of development-cost estimation and availableness through really applying COSMIC-FFP to switching system software.

In this paper, we describe problems and their countermeasures when applying COSMIC-FFP to switching system software, and present the results of analyzing the measurement obtained.

2. Overview of COSMIC-FFP

COSMIC-FFP is a method for measuring the size of software. It provides, in particular, criteria for measuring functional size, and improves upon the IFPUG Function Point Analysis Method so that the software of real-time systems can also be targeted for measurement.

With COSMIC-FFP, the size of software can be determined from the functions required, by the user, of the software. Here, the functional requirements of software are expressed as a set of functional processes, each of which is expressed in turn by an ordered set of data movements. An individual data movement is referred to as a "sub-process" within the functional process in question. The result of measurement in COSMIC-FFP is the number of data movements, i.e., the number of sub-processes, and it is taken to represent "size."

In COSMIC-FFP, one data movement, the unit of measurement, is defined as one Cosmic Functional Size Unit (CFSU).

The software model adopted by COSMIC-FFP is shown in Fig.1.

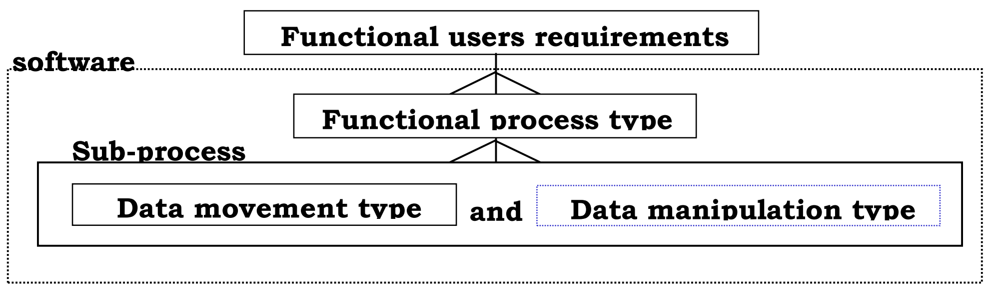


Fig.1 Software model of COSMIC-FFP

3. Problems and Countermeasures in Performing a Measurement

This section describes problems of the measurement of COSMIC-FFP and their countermeasures, and technical problems when COSMIC-FFP is used in estimating switching system software development.

3.1. Problems and Countermeasures When Performing Measurements

Performing COSMIC-FFP measurements requires that the measurer be skilled in applying the method and that measurement conditions be established. COSMIC-FFP, however, is a new method to us. At the beginning, the measurers were not yet accustomed to COSMIC-FFP and the amount of reference literature and case studies was insufficient. Regarding these, we took three countermeasures to eliminate measurement errors by measurers caused by individual skill levels and achieve more accurate measurements.

These countermeasures are the appropriate selection of measurement documents, the early acquisition of measurement techniques by measurers, and establishment of detailed measurement methods, as described below.

3.1.1. Selection of Measurement Documents

The selection of documents that describe inputs to measurements effects measurement results. In the measurement process, a measurer will extract functions to be implemented on the basis of measurement documents. If the descriptions in such documents are vague or inaccurate, functions to be measured will be missed in the extraction process and final measurement results will likewise be inaccurate. Sufficient care must be given when selecting documents to be used in measuring.

In the measurement discussed here, we use two documents referred to as "service specification" and "requirement specification".

The service specification describes what a client needs. It provides an overview of the services and functions that the client desires without regard to system implementation. This document is prepared by the client without worrying about implementation methods, and as such is the most appropriate for determining requirements from the client.

The requirement specification, on the other hand, is a document that describes functions that the system itself must provide. It lists system requirements that have been formulated by investigating the service specification created by the client and deciding on processing schemes. The requirement specification provides an all-inclusive description of requirements that the system must meet to realize the services in question.

We decided to use the requirement specification document for our measurements because it describes the function of the target system more accurately than the service specification.

3.1.2. Acquisition of Measurement Skills by Measurers

No measurers with thorough knowledge of COSMIC-FFP were available to work on this measurement. Since time seemed to be needed to learn about COSMIC-FFP, there was the fear that this would hinder making an efficient measurement. The following three countermeasures were consequently taken.

(1)Employ programmers well-acquainted with the target software as measurers
Although the number of staff members learning about the COSMIC-FFP method were few, there were many who were familiar with the specifications and development of the software targeted for the measurement. Having such staff members perform the measurements enabled us to shorten the time required for acquiring the knowledge needed for making a measurement and made an efficient measurement possible.

(2)Hold briefing sessions by specialists in the method
While it was not possible to collect many staff members with extensive skills in the COSMIC-FFP method as described above, it was possible to gain the support of specialists well-versed in the method. Inviting these specialists to give briefing sessions made it easier for the measurers to acquire COSMIC-FFP skills.

(3)Prepare case studies of switching system software measurement and hold

briefing sessions
Case studies of measuring switching system software were prepared and explained to make it easier for measurers to gain an understanding of the method. As mentioned above, the measurers for this measurement activity were all programmers in the switching system software in question. Preparing these case studies in switching system software helped to make the measurers familiar with the COSMIC-FFP method.

3.1.3. Establishment of Measurement Ways

Due to the fact that our measurers were originally unfamiliar with the COSMIC-FFP method, there was naturally some fear that measurement errors would occur due to differences in the ways different individuals measured. One way of preventing such measurement errors due to differences in measurers' skills and method interpretation would be to provide manuals describing application of the method in detail. At the time of making these measurements, however, measurement manuals for switching system software did not exist. That is to say, there were no manuals that could describe how to identify functional processes and sub-processes in relation to conditions particular to switching system software. The problem thus arose that measurement results might differ due to differences in measurement ways among the measurers.

To solve this problem, we focussed on eliminating individual differences among the measurers in the ways that they measured. The main purpose of the measurement described here was to evaluate the measurement method at a stage in time that preceded the preparation of manuals describing measurements in detail. We therefore attempted to solve this problem of no manual-based measurement criteria by eliminating individual differences in measurements.

We adopted the following three countermeasures to remove differences in measurements among measurers.

(1)Perform a case study
Before beginning the targeted measurement, all measurers were asked to perform the same case study and then compare their measurement processes and results to provide each other with feedback on measurement methods. Performing the same case study in this way facilitated technical discussions on the COSMIC-FFP method on the basis of a common example.

(2)Assign staff to monitor measurements
As one countermeasure to eliminating individual differences in measurement methods, staff were assigned to monitor the measurement ways of each measurer and to detect and resolve individual differences. These staff members reviewed measurement results and checked whether measurements made by one measurer differed from those of other measurers.

(3)Employ a small number of measurers
For the measurement described here, we used a small number of top-notch personnel as measurers, namely four. This approach facilitated communication among the measurers and made countermeasures (1) and (2) above all the more effective. A small number of measurers, moreover, made it possible to assemble the group in one place and to carry out the measurement in a centralized manner.

3.2. Technical problem - Static Functional Enhancement

Next, we describe the technical problem in applying COSMIC-FFP to the switching system.

The software model adopted by COSMIC-FFP, moreover, represents software functional requirements as a set of functional processes. Each functional process, in turn, is a set of ordered sub-processes that carry out data movements. Measurements in COSMIC-FFP target sub-processes, that is, data movements.

Some of functional enhancements, however, may not be directly involved with data movements. A typical example is a so-called "capacity expansion". In this case, process contents after making the enhancement are exactly the same as before—only the amount of data that can be handled has been increased. In other words, a capacity expansion involves a change to constant data and there is no change in the data movements before and after the enhancement.

When making a change to constant data, no changes are made at all to functional processes. Measurement results made by COSMIC-FFP before and after an enhancement do not differ.

In actual software development, however, even capacity expansion generates a non-zero development cost. At a design phase, capacity expansion incurs the costs of studying the range of its effect and performing degradation tests. Such studies and tests associated with capacity expansion must be performed carefully in the case of a real-time system in which processes must be completed within a fixed time period. At the least, a change in constant data will require a development cost that cannot be ignored.

For the above reason, as COSMIC-FFP cannot measure a capacity-expansion type of enhancement that does not involve data movements, some countermeasures need when COSMIC-FFP is intended to use for cost estimation of software developments.

In the measurement discussed here, development items that do not involve data movements are excluded from measurement.

4. Measurement Results and their Analysis

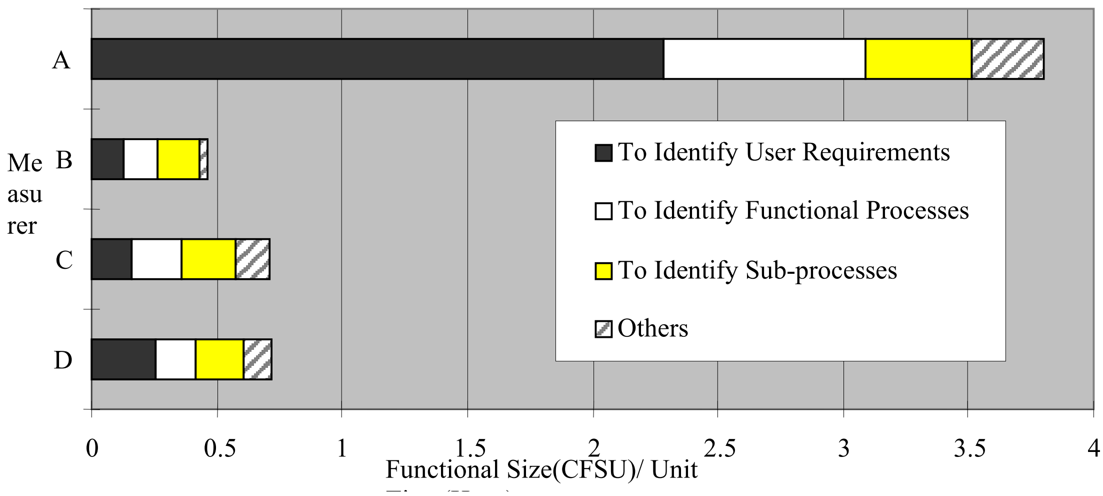
These measurement results are discussed below in terms of “ease of use” and “size and cost.”

4.1. Ease of Use

Ease of use will be discussed here on the basis of measurement speed. We define measurement speed as the number of functions that can be measured per unit time. In COSMIC-FFP, this would be the number of CFSUs per unit time. Trends in measurement speed over time may be supposed to indicate the learning curve of the method, i.e., ease of use.

Measurement results are shown in Fig. 2. These results show measurement speed for each measurer and measurement speed for each measurement step in COSMIC-FFP. Here, “COSMIC-FFP measurement steps” refer to the work procedure performed according to the COSMIC-FFP measurement method. Main steps are “to get functional user requirements,” “to identify functional processes,” and “to identify sub-processes.” There are other steps in the COSMIC-FFP measurement procedure like “to identify boundaries,” “to apply measurement function,” and “to calculate total,” but as these take only a short time, they have been included in the three main steps described above.

Measurement Speed by Phase



Comparing the results for measurement speed, measurer A is by far the fastest with the other measurers exhibiting about the same speed. The average measurement speed of those three measurers excluding measurer A is 0.6CFSU/h. In contrast, the measurement speed of measurer A is 3.8CFSU/h, which is about six times greater.

Specifically, measurer A is in charge of the design and implementation of those functional enhancements targeted for measurement and has sufficient knowledge of those measurement items from their requirement specifications to their internal processing schemes. The other measurers, on the other hand, while working in the design and implementation of the targeted system, were not involved in the design and implementation of the functional enhancements to be measured. In short, they had knowledge and experience for understanding the specifications, they must begin, however, with an understanding of the functional enhancements to be measured.

At the same time, as measurer A and the other three measurers were all beginners as far as the COSMIC-FFP method was concerned, there were no differences in terms of measurement method skills and knowledge. The above results therefore reveal that familiarity with the specifications of measured items was a major factor in the high measurement speed demonstrated by measurer A.

For measurers B, C, and D, measurement speed was about the same at about 30% for measurement steps “to get functional user requirements,” “to identify functional processes,” and “to identify sub-processes.” In contrast, measurer A achieved measurement speeds of about 10%, 30%, and 60% respectively. It was found that measurement speed in identifying functional process and sub-processes and overall measurement speed tends to increase if one is familiar with specifications.

4.2. Relationship between Size and Cost

One of the most interesting applications of COSMIC-FFP is software development cost estimation. We describes in this section the results of analysis of the relationship between sizes measured in COSMIC-FFP and costs.

4.2.1. Size and Cost Relationship

4.2.1.1. Overall Relationship

The results of performing regression analysis on development cost and the results of COSMIC-FFP measurements are shown in Fig. 3 using heavy line.

The correlation coefficient was found to be R=0.986 for the heavy line in Fig.3, and the figure itself shows that there is high correlation between development cost and COSMIC-FFP measurement results. The regression equation is as follows;

Cost= 0.173 x CFSU+ 0.335

(4-1)

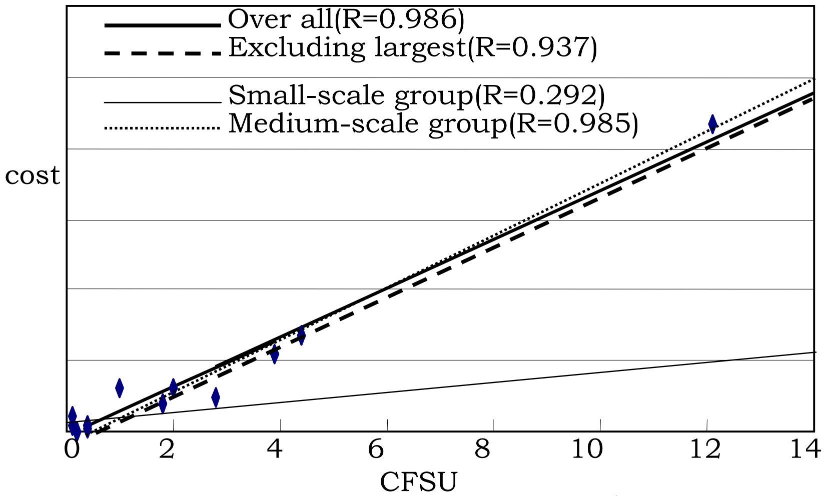


Fig.3 Measurement result

4.2.1.2. Trend When Excluding Largest Size

When obtaining the relationship between development cost and measurement results, it is important to clarify how each result affects that relationship. This is because the sizes of measurement results can provide information on the accuracy of a prediction equation and how to select the best one from a set of prediction equations.

Examining the distribution of data in Fig. 3, we see that the data value corresponding to the largest size on the right side away from the other data values. This data value may therefore have a great effect on the overall relationship. Here, we divide results into three groups; one for data of the largest size (one data value), another for a medium-size data group (six data values in the scale range from 5 – 120), and the other for a small-size data group (six data values less than a scale of 5). The following discusses respective correlation between development cost and measurement results according to size.

A heavy broken line in Figure 3 shows the correlation between development cost and measurement results for 12 data values only excluding the largest size value.

Cost = 0.13 x CFSU + 0.18

The correlation coefficient is R=0.937, similar to the results for overall relationship, the correlation with the regression coefficient is high. The similarity and high correlation in both of the above cases tells us that largest-scale data is not a determining factor in the relationship between development cost and measurement results.

4.2.1.3. Small-Size Group and Medium-Size Group

We next divide results into a small-size group and medium-size group and look for relationships. The results of regression analysis for each of these groups are shown using a broken line and a line in Fig. 3, respectively. The regression equations are as follows;

$$\textbf{Cost} = \textbf{0.04} \ \textbf{x CFSU} + \textbf{0.23} \quad (4\text{-}3)$$

$$\textbf{Cost} = \textbf{0.18} \ \textbf{x CFSU} - \textbf{0.97} \quad (4\text{-}4)$$

For the medium-size group, correlation is high. The results show a relationship similar to that of overall results. On comparing with the results that exclude the largest scale (Fig. 3), moreover, it can be seen that the relationship shown by the medium-size group is closer to the overall one.

In contrast, results for the small-size group show a relationship completely different from that of overall results. And the correlation coefficient here is low.

In the case of a small-size group in which the number of CFSUs is less than ‘5,’ even a difference in measurement results of ‘1’ means a minimum difference of 1/5 or 20%. A case in which the smallest measurement unit is relatively large is one factor in low correlation.

4.2.2. Analysis of Relation between Development Cost and Measurement Results by COSMIC-FFP Viewing from Implementation

Analyzing the relationship between LOC and measurement results by COSMIC-FFP is interesting because COSMIC-FFP's sub-processes somehow do something to do with implementation, which LOC represent. In doing so, we chose the medium-size data group since a high correlation between development cost and measurement results by COSMIC-FFP was obtained for all data groups other than the small-size one, and since the medium-size data group in particular behaved very similar to overall one. In short, it was thought that analyzing the medium-size data group could enable us to extract overall features effectively.

4.2.2.1.Analysis of Correspondence between Sub-Processes and Source Code

On performing a detailed analysis of the relation between LOC and measurement results by COSMIC-FFP, we examined the correspondence between sub-processes measured by COSMIC-FFP and actual software.

We supposed that there is some relationship between sub-processes in COSMIC-FFP and actual software. On analyzing the relation between development cost and measurement results by COSMIC-FFP, we believe that we can realize more detail considering an analysis of the relationship between LOC and measurement results by COSMIC-FFP.

There are three types of correspondences between sub-processes and software. The first, called pattern-1, occurs when a program corresponding to a sub-process exists. The second, called pattern-2, occurs when a sub-process is identified but no corresponding program exists. The third, called pattern-3, occurs when a program exists but no corresponding sub-process does.

Examining the features of the above three patterns sheds light on the correspondence between the COSMIC-FFP measurement method and actual structured software and is therefore useful in understanding the high correlation between cost and measurement results.

4.2.2.2.Pattern-1

Most of the data in the medium-size data group falls into pattern-1. In the COSMIC-FFP measurements, pattern-1 covers 60%.

In addition, LOC per 1CFSU vary narrowly and over 90% of them are included into the range of the standard deviation(0.33) centering on a mode.

The data distribution in this regard is shown in Fig. 4. Lines of code per 1CFSU are shown on the horizontal axis while the associated frequency of appearance is shown on the vertical axis.

4.2.2.3.Pattern-2

Pattern-2 refers to the situation in which a new function is required in the switching system but a new program for that function does not exist. While a corresponding program to achieve that function is available, an existing program can be used to do a functional enhancement instead of having to add a new program.

Existing programs can often be used in this way in implementing command interfaces. In a switching system, various commands are provided to enable system operators and connected systems to issue processing instructions and to check system status. In some cases, new functions are added to these commands when making functional enhancements to the switching system, and new parameters that specify detailed command operations are added. The program, in turn, has been created to deal with these additional parameters other than those for functions originally allowed, and processing was implemented accordingly. As a result, existing processing can be used as-is and the request can be processed.

4.2.2.4.Pattern-3

Pattern-3 can occur due to coding rules and the manner of module division and to policies governing class design, and then it covers 72% of the cases. The other reasons cover only 28%, which have something to do with internal processing to implement.

The reason that it covers the majority of cases is described below. In this regard, the measurement described here concerns functional enhancements only. Only sub-processes that were added, modified, or deleted to satisfy requirement specifications in existing functional processes were counted. Existing sub-processes that could be used as-is were not counted as an addition. In actual program implementation, however, a new program may be added, even if identical to an existing one, as a consequence of coding rules, module-division customs, and class-design policies. A disparity can therefore arise between sub-processes and the actual program.

4.2.2.5.The Reason Why the Correlation with Development Cost Is High

There may be several ways to rationalize the high correlation between development costs and the measurement results by COSMIC-FFP. As mentioned above, from the viewpoint of COSMIC-FFP sub-processes and implementation, it can be explained by the two following reasons.

(1) The pattern-1 covers 60% of the whole and the variations of LOC per 1CFSU are included into a small range.

(2). The average value acquired when LOC of pattern-3 are divided by the number of the sub-processes of pattern-2 is contained in the range of the standard deviation centering on the mode of pattern-1. This must be incidental but is worth further analyzing.

5. Summary

This paper has analyzed the results of measuring sizes of real time software systems using COSMIC-FFP and the relationship of the development costs of switching system software and the results of COSMIC-FFP measurements. The analysis, we believe, has clarified the structure of those results. It is also worth mentioning that it was found that COSMIC-FFP is effective in estimating development costs.

References

[1]A.Abran,"An Implementation of COSMIC Functional Size Measurement Concepts",

Keynote presentation at the FESMA 99 Conference Amsterdam, Oct. 7, 1999
[2]A.Abran, J.-M. Desharnais, S.Oligny, D. StPierre, C.Symons, ”COSMIC-FFP Measurement Manual, Version2.0”, 1999
[3]A.Abran, C.Symons, S.Oligny, “An overview of COSMIC-FFP field trail results”, ESCOM 2001 London, April 2-4,2001