Estimating maintenance projects using COSMIC-FFP

Tom Koppenberg, Ton Dekkers

Sogeti Nederland B.V.

tom.koppenberg@sogeti.nl, ton.dekkers@sogeti.nl

Abstract:

A large number of software projects are enhancement projects of existing software. For estimating new projects acceptance of COSMIC Full Function Points [1] is rapidly growing because it has already proven to be a good alternative for Function Point Analysis. Estimating enhancements using classic Function Point Analysis [2] has always been somewhat controversial, but we believe that COSMIC can be a very good alternative in the very near future.

Keywords

COSMIC-FFP, maintenance, enhancement

1 Introduction

Nowadays the majority of software projects are projects that enhance and extent existing software. For estimating new projects acceptance of COSMIC-FFP is rapidly growing because it has already proven to be a good alternative for Function Point Analysis [3]. Estimating enhancements using classic Function Point Analysis has always been somewhat controversial, but we believe that COSMIC-FFP can be a very good alternative in the very near future. This article gives an overview of the possibilities of estimating enhancement projects using COSMIC-FFP.

2 What is enhancement?

Before we can say anything about the measurement of enhancement, it needs to be clear what kinds of enhancements exist. Two categories of software enhancements can be distinguished: planned enhancements and ad-hoc enhancements. Ad-hoc enhancements can be explained as solving problems that can't be put off (e.g. bug fixing).

Planned enhancement can be separated into three types:

- Perfective enhancement; adapting the software to the new requirements of the user, like new functions or better performance.
- Adaptive enhancement; adapting the software to new circumstances, like new hardware, new middleware or new legal requirements.

- Corrective planned enhancement; more structured repair of problems or defects solved with ad-hoc enhancements.

3 Estimating ad-hoc enhancements

Ad-hoc enhancements can not be addressed in releases by definition. When a problem arises that must be dealt with immediately, planning and estimating are no issues. The problem must be dealt with as soon as possible. Estimating specific ad-hoc enhancements is hardly possible and because of its nature it has no added value. Resolving the problem always has the highest priority above estimating in these cases. However for staffing and budgeting purposes it is important to estimate the yearly costs of ad-hoc maintenance. The yearly costs for ad-hoc enhancements can be an important trigger to replace the software to be maintained.

The yearly costs for ad-hoc enhancement are rather easy to estimate with the existing methods. What is needed is the experience statistics of the number of hours per year per size unit that were needed for ad-hoc enhancements (reliability). Multiplying this figure with the size of a system leads to the yearly cost for ad-hoc enhancements. This method is independent of the method that is used to calculate the size of software.

4 The future of functional size measurement

Function Point Analysis will likely be replaced by next generation functional size measurement methods like COSMIC-FFP in the next decade [4]. The main reason for this is that Function Point Analysis does not always works well with a growing number of contemporary software developing environments. COSMIC-FFP can be seen as the most important candidate to replace Function Point Analysis.

The next paragraphs will show the difference between the current way to measure software enhancements with Function Point Analysis and the way COSMIC-FFP deals with measuring enhancements.

5 Estimating planned enhancement

From a functional point of view, planned enhancement projects can be split into:

- adding new functionality
- changing existing functionality
- deleting existing obsolete functionality
- replacing existing functionality (= delete + new)

The measurement of new functionality is identical with the regular measurement of functional size. The others are more difficult and more controversial because there are different ways of dealing with those enhancements within Function Point Analysis.

5.1 Enhancement and IFPUG Function Point Analysis

For understanding the advantages of COSMIC-FFP over Function Point Analysis it is useful to explain the principles of maintenance Function Point Analysis.

According to IFPUG, enhancements will be measured by measuring all concerned functions. Instead of using the productivity ratio for new development, a different productivity rate will be used for enhancements. In general this rate will be lower. A disadvantage of using this method is that empirical productivity facts found during developing new software can't be reused, because different productivity ratios will be used.

5.2 Enhancement and NESMA Function Point Analysis

In 1992 Sogeti (at that time named Interprogram) proposed a method for maintenance Function Point Analysis in which the change of the way data element types and record types are used in a function is measured. Later the NESMA Workgroup FPA and Enhancement published a document describing an approach based on the same principles [5].

The first step in this approach is to make an inventory of all the (elementary) functions involved in a new release of the software. Some of the functionality will be new, but usually a lot of the functionality of a release will be a change to existing functionality. The effort to deliver these changes depends on the impact on the existing functionality.

The next step is to weight the impact and categorize the function in one of the maintenance classes. Each class has its own impact factor. The size of the functions in each class will be multiplied with the impact factor of that class to get the size of the enhancement in maintenance function points.

There are five different classes:

- low impact change;
- average impact change;
- high impact change;
- new functionality;
- deleted functionality.

Usually there is not much discussion about the last two classes. For the first three classes this is different: low, average and high impact tends to be a subjective classification.

Function Point Analysis for software enhancement is based on the assumption that the way in which a function changes is related to the change of the way data element types (DET) and file types (FTR) are used in a function. The number of changes can be defined as:

number of additions
+ number of deletions
+ number of changes

Δ DET / Δ FTR	0 – 1	2 – 5	> 5
0	L	L	A
1 – 2	L	A	H
> 2	A	H	H

Table 1: Impact factors for changed functions

Δ DET	0 – 1	2 – 5	> 5
-	L	L	A

Table 2: Impact factors for changed data

The valuation as mentioned in table 1 and 2 is based on absolute number of changes. This method is used by Sogeti. NESMA bases the valuation on relative changes.

The Sogeti approach is pretty much straightforward. Three impact factors must be matched to three maintenance classes:

- low impact change:	0.25;
- average impact change:	0.50;
- high impact change:	0.75.

The impact factors are empirical factors based on estimation of projects. The impact factor is de ratio between the effort for building functionality and the effort needed to change the functionality in a particular maintenance class.

For deleting functionality Sogeti works with an impact factor of 0.10. NESMA is advising a factor of 0.40.

The advantage of using the described approach is that the experiences from developing new software can be reused. A disadvantage is that the statistical evidence of the used factors is weak. Although in the years this approach is used the factors are not questioned and no research is know about calibrating these factors.

5.3 Enhancement and COSMIC-FFP

Applying this approach for measuring maintenance when using COSMIC-FFP is just a small step. The only difference is related to the sizing in itself. The size of the functional process (almost equals the elementary function) in COSMIC-FFP is measured based on the number of data movements, where in Function Point Anlysis the size is derived from function type and complexity classification.

In COSMIC-FFP for any functional process, the functional size of changes to the Functional User Requirements is aggregated from the sizes of the corresponding modified data movements according to the following formula:

S size(added data movements) +

S size(modified data movements) +

S size(deleted data movements)

Sizing enhancement is an integral part of the Measurement Manual. According to the Measurement Manual, all new, changed or deleted data movements should be counted [1]. The impact of enhancement is expressed in the number of data movements that are impacted by the enhancement in any way.

For new functionality this approach doesn't differ from Function Point Analysis, the functional process will be measured the same way as it will be measured for new development.

For changed functionality only the data movements will be measured that are changed, added or deleted. The guideline for business application software defines when a data movement is considered to be functionally changed [6]. This is when:

A) the data group and/or

B) it's data manipulation are changed in any way.

To be more precise:

A) a data group is changed if

- attributes are added to this data group and/or
- removed from the data group or one or
- more existing attributes are changed, e.g. in meaning or format

B) the data manipulation is changed if

- the specific formatting, presentation and/or validation of the data are changed or
- the data manipulation (i.e. processing or computation) associated with the data movement is changed in any way.

The functional size of deleted functions will be determined by the sum of all deleted data movements. For the replacement of a function only the data movements of the new function will be measured.

6 Estimating software enhancements using COSMIC-FFP

For estimating software enhancements the product delivery rate for new development can be used for new and changed functionality. For deleting complete functional processes this will lead to an overestimation of the necessary effort. COSMIC-FFP measures the size of the software enhancement in the number of impacted data movements. The deletion of a complete functional process results in the same size as a new functional process. COSMIC-FFP does not make use of enhancement classes and impact factors.

When using a product delivery rate for new software the effort for deleting functionality will be overestimated in comparison to new development. This can be solved by using another productivity rate for deleting functionality. Then there still is the problem that the ratio for effort between the separated functions will not be correct (by example new functions and deleted functions) so another rate will only work when the ratio between the number of new, changed and removed functions will be around the same. A second disadvantage when separated experience data is collected for new and enhanced projects, is that groups of statistics will be smaller, because new and enhanced projects anymore can’t be compared.

A solution can be found in using different productivity rates for the different kinds of enhanced functions.

For new functions we can use the same productivity ratio as we use for new development. For changed functions we can use this productivity ratio as well. Depending on the weight of the enhancements the rate *changed Cfsu*/*original Cfsu* will raise. When all data movements of a function will be changed, the expected effort for the enhancements will be the same as the effort for new development of the original function. When half of the data movements will be chanced, the expected effort will be half of the effort for new development.

For deleted functions a lower productivity per Cfsu can be expected. The easiest way to determine the productivity for deleting functions is to multiply the productivity rate for new development by an impact factor. Sogeti uses an impact factor of 0.10 in line with estimating enhancements with Function Point Analysis. Experiences within the organization can be used to calibrate the impact factor.

Retesting is not mentioned in the Measurement Manual. The number of functions that had to be retest can be much larger than the number of enhanced functions. Retesting can be estimated by multiplying the unchanged functions that have to be retest by a productivity rate. Analogously to the removal of functions Sogeti has determined this productivity rate as the productivity rate for new development multiplied by an impact factor 0.10. Based on own experiences, this factor can be calibrated. NESMA doesn't explicitly estimate retesting of unchanged functions.

The effects of applying the approach with the default values is shown in table 3:

Funct. process	**Before**	**New**	**Change**	**Remove**	**Test**	**Total**	**After**
FP-1	5						5
FP-2	10				10		10
FP-3	5		2				5
FP-4	8		4				8
FP-5	6		4				7
FP-6	5			5			0
FP-7	0	9					9
Cfsu	**39**	**9**	**10**	**5**	**10**		**44**
Prod. Rate (h/Cfsu)		8	8	8	8		
Impact		1.00	1.00	0.10	0.10		
Hours		**72**	**80**	**4**	**8**	**164**	

Table 3: Example estimating enhancement project with COSMIC-FFP

The advance of using Cosmic for measuring enhancements is that the productivity rate for new development can be reused and so it's possible to compare the productivity rate of new development and enhancements. A disadvantage is that the right impact factor for deleting functions and retest has to be found.

The advantages of having a calibrated impact factor are obvious: the estimates will be more accurate and the productivity is in balance with the effort of removal and testing.

There are also disadvantages in regard to calibrated impact factors: calibration takes time for research and analysis and a above that creates different “sizes” for each organization. The last complicates benchmarking.

The disadvantages make that managers takes the usually minor effects of a “methodical” fixed impact factor for granted and no calibration is done.

7 Conclusion

COSMIC-FFP is already successful for new development. Since the major part of all software development consist of enhancement COSMIC-FFP can only be successful if there’s reliable method to estimate maintenance projects. In this paper we’ve described a method that provides the opportunities for consistent measurement of maintenance with COSMIC-FFP.

References

1. Abran, A., Desharnais, J.M., Oligny, S., St-Pierre, D., Symons, C., COSMIC-FFP Measurement Manual – the COSMIC implementation guide for ISO/IEC 19761: 2003, January 2003, http://www.lrgl.uqam.ca/cosmic-fpp
2. IFPUG, “Function Point Counting Practices Manual, version 4.2, International Function Point Users Group, 2004, http://www.ifpug.org
3. Dekkers, A.J.E., Vogelezang, F.W., 'COSMIC Full Function Points: Additional to or replacing FPA', Proceedings of the 8th European Systems and Software Engineering Process Group Conference (ESEPG 2003), London, June 2003
4. Vogelezang, F.W., 'COSMIC Full Function Points – The next generation of functional sizing, NESMA Jubilee Book, 2004
5. Engelhart, J.T., Langbroek, P.L., Dekkers, A.J.E., Peters, H.J.G., Reijnders, P.H.J., Function point analysis for software enhancement, A professional guide of the Netherlands Software Metric Users Association, NESMA 2001 (in 1998 the Dutch version was already published)
6. Lesterhuis, A., Symons, C., Guideline for the application of COSMIC-FFP for sizing business/MIS application software, version 001, 2004