(Extended) Functional size measurement methods are also applicable in enhancement projects

Ton Dekkers
Sogeti Nederland B.V.
Goudsbloemvallei 50, 5237 MK Den Bosch, Netherlands
ton.dekkers@sogeti.nl

Looking at the distribution of the costs of IT, the largest part of the budget is allocated to maintenance and enhancement projects. New development comprises between 30 to 50% of IT costs. Functional size measurement methods are mostly used for new development only. With some extensions to common size measurement methods like Function Point Analysis and COSMIC Full Function Point one can tackle almost all IT activities. Furthermore the same productivity rates (performance) can be used in enhancement projects also. Over the last 10 years we used the extended measurement method based on Function Point Analysis very successfully in a great number of projects.

Sogeti Netherlands

Sogeti Nederland B.V. is a Dutch software services company with 1,700 employees. In august 2002 IQUIP Informatica, Gimbrere & Dohmen and Twinsoft merged to form Sogeti. Since 1988 IQUIP is known in the Netherlands as a promoter and initiator of functional size measurement. Sogeti continues the leading role of IQUIP by means of the Expertise Centre Metrics within the Engineering & Projecten division of Sogeti. Sogeti plays an active role in the promotion and further development of Function Point Analysis (FPA) [1] and COSMIC Full Function Points (CFFP) [2] by participating in working groups of the Netherlands Software Metrics Association (NESMA) and the Measurement Practices Committee of COSMIC.

Limitations

Before going into details it is better to address the limitations of functional size measurement methods first. The most important limitation is related to the purpose of the sizing. When using the outcome of sizing for estimating, one has to be aware that the number of function points (fp) or cosmic functional size units (cfsu) need to be above a minimal level. To make a reliable estimate for (new) development project the size should be over 200 fp or 100 cfsu. The same thing applies to enhancement / maintenance projects. Using the extended methods, a project should have a minimum of 100 maintenance function points (mfp) or 60 mcfsu.
In practise this means that the use of the extended methods is (only) applicable in organisations that work with releases. If every request for chance is put through in the system immediately, it is better to use "expert" estimates.
There is also another generic limitation. The requirements / request for changes should be defined clearly and completely otherwise functional size measurement will give a result accompanied with a lot of hypothesises.

Applicability

The above mentioned utilisation of (extended) functional size measurement is for budgeting purposes. Estimation, in a simplified form, is done by multiplying the result of the sizing with performance indicators like productivity rate (effort per unit) or cost per unit. Within release management functional size measurement can play a crucial role. Once the budget is agreed upon, the amount of functionality that can be produced within that budget is easy to calculate. In other words, the amount of functionality can be calculated by dividing the budget by the same performance indicators. When the size of required functionality (enhancements) for the release is known, this size can be matched with the

calculated size. If necessary the principal can select the functionality based on priority and fit within budget or can look for additional budget.
Last but not least one must not forget performance measurement. Not only to measure the performance of one's own organisation but also to measure the performance of the supplier's organisation. This is especially important in maintenance situations because of the trend towards outsourcing this activity.

One has to keep in mind that functional size measurement is an aid in managing a(n enhancement) project. The estimations based on FPA or CFFP should be looked at critically and have to judged in conjunction with the project specific circumstances. This is shown in the measurement model (figure 1).
The size of the software and the performance of system development are the key items in predicting the required effort and understanding the costs per unit of an application or project. Once the size and the development methods are known, the risks become visible too. These risks will be analysed and the impact and the probability of the risks and the costs of the measures are the criteria for transposing into measures. All components are input for project budget and planning.

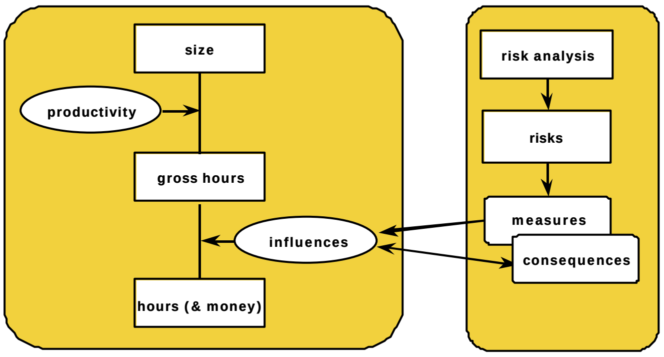

figure 1 - measurement model

By prefer estimates are made based on organisations own performance rates but when these rates are not available, rates provided by third parties can be used. Very useful are the project delivery rates (productivity rates) of the International Software Benchmarking Standards Group (ISBSG) [3]. This "not for profit" organisation provides rates based on a project database with over 2,000 projects from many countries, many branches and various development platforms.

Global Procedure

Functional size measurement is an objective method to determine the size of an information system. The size is measured in units (fp or cfsu). The same type of unit is applicable when measuring the size of enhancements (release).
Based on the requests for change (RfC) the functions involved are identified. When using FPA the functions correspond with the elementary functions and when using CFFP the functions are recognised as functional processes. When all functions are identified (and measured), the size of the release is known.
During the impact analysis the impact of the RfC's on the functions is assessed and based on that an enhancement type is allocated to the functions. The number of units allocated to an enhancement is multiplied with a factor related to that type.
When adding the results per type the weighted size is obtained. This size is the size required as input in the measurement model.

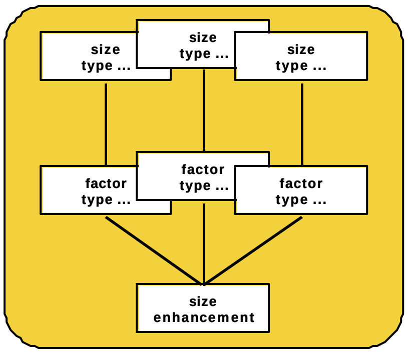

figure 2 - size measurement maintenance

Maintenance (definitions)
There are two viewpoints when looking at maintenance: the technological state of a system and the user value of the system. The system must comply with the technological requirements of the organisation. User value means that the software supports the practises and procedures to fulfil the needs of the user. So maintenance / enhancement can be defined as optimising the technological state and the user value in a controlled way.
In literature often the following distinction is made in maintenance categories:

- corrective maintenance;
- perfective maintenance;
- adaptive maintenance.

There are various definitions used for these categories therefore it is necessary to define first what is meant by these categories. The most common definitions are from Lientz and Swanson [4]:
Corrective maintenance is all activities needed to solve defects in the software. Not only defects in the software (programming bugs) but also errors in the design.
Perfective maintenance is enhancing the operational software. This means updating existing functionality based on RfC that are proposed by and agreed upon by the user and the principal.
Adaptive maintenance implies adding completely new functionality to the operational software. Additions are required because of new functional user requirements.

Estimating corrective maintenance
Defects, such as operational failure or the system is not functioning properly, need to be repaired immediately. Waiting for an estimate and a formal go / no go is undesirable and not acceptable for the organisation. The problem is that defects occur unexpectedly and interfere with other (important) activities and schedules. Of course, the size of the repaired functionality can be and has to be measured but one first needs to correct the defect and then do the measuring. Measuring corrective maintenance helps to get insight in two basic performance indicators: quality (defects per operational system per size unit) and mean time to repair (time per operational system / environment per defect). If these performance indicators are unavailable, time spent on corrective maintenance is a much easier indicator. A risk factor is determined based on hours spent on corrective maintenance: simply by dividing the size of the operational systems (portfolio) by the hours spent within a defined period of time. If the portfolio changes, new systems added or existing systems are replaced, the new required effort for corrective maintenance for the next period can be calculated. Taking this effort into account, scheduling activities like new development or enhancement time to delivery will be more realistic.

risk factor =	time spent (in period) / size operational systems
effort =	(new) size operational systems * risk factor

Identifying a risk factor for distinctive types of operational systems can refine capacity estimation. A distinction can be made between relatively new systems (less then one year operational), average aged systems and old systems (more than five years operational). Based on experience a system shows fewer defects after one year in production. When becoming of age (over five years) the enhancements influence the structure and stability of the systems and cause an increasing number of defects. One lesson learned: do not try making this distinction when first starting with measurement.
This “method” to estimate corrective measurement applies to both function point analysis and COSMIC full function points. To measure the size of the repaired functionality, the method for sizing perfective maintenance is used.

Estimation of perfective and adaptive maintenance using FPA

Step 1: Determining size release

The functionality effected by maintenance has to be identified. This is done in the diagnostic phase. Based on the BfC the analyst identifies all elementary functions (Transaction Types: EI - External Input, EO - External Output, EQ - External Enquiry) and Logical Files (ILF - Internal Logical Files, EIF - External Interface Files) that are effected by the enhancement in one way or another. Also the "domino"-effect" has to be taken into account: a change in a ILF effects all elementary functions that use (read, update) this ILF. Furthermore when a elementary function changes and there is a dependence between that elementary function and other elementary functions (e.g. function processes data from another function or provides data to another function), these functions are effected as well.
The size of the release is the sum of the sizes of all identified functions and files. In this case the size is expressed in function points (fp).

Step 2: Allocate to enhancement types

The basic procedure to make an estimate in case of new development is to multiply the size of the application (in fp) with the productivity rate (hours/fp). The productivity rate is correlated to a repeatable process, the activities are carried out in the same way, using agreed upon standards and tools.
Unfortunately maintenance is not the same as new development. In principal the activities are more or less the same, but the way the activities are carried out depends on the impact the RfC has upon the identified functionality. In the estimate the impact of the various "types" of enhancements must be visible and taken into account. Based on experience and common sense the following enhancement types are recognised:

- functionality to be changed (low, average and high complexity);
- functionality to be added (new);
- functionality to be removed;
- functionality to be tested.

The allocation to the effected functionality to the enhancement types is done during the impact analysis phase.
As a rule the allocation of the types requires no further discussion, as they are obvious. However the type 'change' makes a distinction between changes with 'low', 'average' and 'high' complexity. This distinction is fairly subjective.
The method described in this paper allows the user of the method to decide how this classification will be done. In the Netherlands three ways of classification are used:

- purely based on one's own understanding of the impact;
- based on a matrix where the number of effected logical files and data-element-types determines the classification;
- based on a matrix where the relative percentage of effected logical files and data-element-types determines the classification.

The latter two provide a certain level of objectivity in classification and to avoid discussion it is best to choose one of these. Sogeti has defined the matrix based on numbers in 1992 [5] and stills uses this approach for its own purposes and also for its customers. In 1997 the NESMA published a paper [6][7] describing the relative classification matrix however field trials by Sogeti showed better results using its own approach, therefore there was no reason to change.

Fundamental in both approaches are the principles of FPA. The complexity of the elementary functions identified by FPA is classified as 'low', 'average' and 'high'. The same classification is used for changes.

The next step is quite obvious, use the same variables to classify the changes that are used for classifying the functions.

What are chances? In this approach changes include all items that are different then before:

Number of changes =	number of items added + number of items changed + number of items removed.

For valuing the chances to the Logical Files (ILF, EIF), the number of changes related to the Record Types (Δ RT) and the Data Element Types (Δ DET) has to be identified. For the Transaction Types (EI, EO, EQ) the number of changes of the File Types Referenced (Δ FTR) and the Data Element Types (Δ DET) are relevant.

<u>Valuation based on number of changes</u>

After determining the number of changes the impact factor is selected based on the matrices. There is only one matrix defined for Transaction Types (EI, EO, EQ).

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

A similar matrix is used for Logical Files (ILF, EIF). In this case Δ RT is found not to have impact on effort, so only the first row is used:

Δ DET	0 – 1	2 – 5	> 5
-	L	L	A

When a Logical File changes of type following is agreed upon: EIF -> ILF use impact factor “Low” on the new size of the ILF, from ILF -> EIF no impact counted for.

Example:
The HRM department requests a birthday list of all employees in EMPLOYEE (first name, middle name, surname, department name, location and date of birth) sorted by birthday. The department name and the location have to be retrieved from DEPARTMENT.
Measurement: Type of transaction - External Output, number of FTR - 2, number of DET - 6, size - 5 function points
RfC: Add phone and email to the list. Both are available in EMPLOYEE.
Measurement: Type of transaction - External Output, number of FTR - 2, number of DET - 8, size - 5 function points; D FTR – 0, D DET – 2, enhancement type – L.

<u>Valuation based on relative changes</u>

This valuation is described in "Function Point Analysis for Software Enhancement", the principles of the previously described approach apply to this method. The only difference is the determination of the variables that are used to value the impact.

% FTR = (Δ FTR / number of FTR before change) * 100%
% DET = (Δ DET / number of DET's before change) * 100%

For this method there is one matrix defined for transaction types (EI, EO, EQ) …

% DET / % FTR	< 66.8 %	66.8% – 100%	> 100%
< 33.3%	I	II	III
33.3% – 66.7%	II	III	IV
66.8% - 100%	III	IV	V
> 100%	IV	V	VI

… and one matrix for the Logical Files.

% DET	< 33.3%	33.3% - 66.7%	66.8% - 100%	> 100%
-	I	II	III	IV

Changing file type (EIF <-> ILF) gets the impact factor 0,40. The number of fp that will be used to determine the mfp is the number of fp of the ILF (old / new).

Example: Birthday list
RfC: Add phone and email to the list. Both are available in EMPLOYEE.
Measurement: Type of transaction - External Output, number of FTR - 2, number of DET - 8, size - 5 fp, % FTR – 0, % DET – 66.7%, enhancement type – I.

Step 3: Determine the size of the enhancement project
When enhancement types are allocated to the effected functionality (Logical Files and Transaction Types) the size is derived from the actual size in function points of each item. The enhancement type corresponds with the impact factor.
The aim of the impact factor is to define the correlation between the effort for a change and the effort of new development. On empirical basis the values of the various impact factors are defined. In the 10 years that the methods have been used there was no reason for changing the chosen values. Sogeti's experiences refer only to its own approach based on numbers.

In the following table the impact factors used in both approaches are compared.

	Enhancement type		Impact factor	
	Sogeti	NESMA	Sogeti	NESMA
New	New	New	1.00	1.00
	Low	I	0.25	0.25
	Average	II	0.50	0.50
Change	High	III	0.75	0.75
		IV		1.00
	Replace *	V	1.10	1.25
		VI		1.50
Remove	Remove	Remove	0.10	0.40
Test (only)	Test (only)	-	0.10	-

* replace = remove + new

The impact factor is used to calculate the size in mfp for each item affected. The sum or the derived size of each item is the total size of the enhancement project.

Example birthday list:
Sizing: actual size - 5 function points, enhancement type – L, impact factor – 0.25, enhancement size- 1.25 mfp.

Example determining size enhancement project with FPA
Using an example helps to understand how it works. Some remarks on this example:

- allocating enhancement type is not part of the example;
- The RfC proposed only changes that have effect on transaction types FPA-1 thru FPA-6 and require a new transaction type FPA-7.

Trans. Type	Before	New	Change L	Change A	Change H	Rem	Test	Total	After
FPA-1	4						4		4
FPA-2	7						7		7
FPA-3	4		4						4
FPA-4	6			6					6
FPA-5	5				7				7
FPA-6	7			7					7
FPA-7	0	4							4
fp	33	4	4	13	7	0	11		39
Impact		1.00	0.25	0.50	0.75	0.10	0.10		
mfp		4.00	2.00	6.50	5.25	0.00	1.10	18.85	

To be able to calculate the effort, a productivity rate is chosen: 8 hours/fp. The rate includes all activities to be carried out during new development.
Because the impact factor is weighting of effort related to new development, the productivity rate is also applicable for changes. In this case the base effort is 151 hours (18.85 mfp * 8 h/mfp).

Estimation of perfective and adaptive maintenance using CFFP
Step 1: Determining size release
The functionality effected by maintenance has to be identified, this is done in the diagnostic phase. Based on the BfC the analyst identifies all functional processes that are effected by the enhancement in one way or another. The size of the release is the sum of the sizes of all identified functional processes. In this case the size is expressed in Cosmic functional size units (cfsu).

Step 2: Allocate to enhancement types
The same enhancement types are recognised when using CFFP. The biggest difference is the way of looking at changes.
A functional process is size by counting all implied data movements (entries, read, writes and exits). Changes are measured by identifying the data movements that are effected by the RfC.

Example Birthday List:
Measurement: data movements – 5 (read EMPLOYEE, read DEPARTMENT, exit EMPLOYEE data, exit DEPARTMENT data, exit messages), size – 5 cfsu.
RfC: *Measurement: data movements effected 2 (read EMPLOYEE, exit EMPLOYEE data), enhancement size – 2 cfsu.*

Step 3: Determine the size of the enhancement project
When enhancement types are allocated to the effected functional processes the size is derived from the actual size in cfsu of each item. The enhancement type corresponds with the impact factor, except changes (see step 2).

Example determining size enhancement project with CFFP
To show how it works, the same example from FPA is used.

Funct. Process	Before	New	Change	Rem	Test	Total	After
FP-1	5				5		5
FP-2	8				8		8
FP-3	5		2				5
FP-4	8		3				8
FP-5	6		4				7
FP-6	9		5				9
FP-7	0	5					5
cfsu	41	5	14	0	13		47
Impact		1.00	1.00	0.10	0.10		
(m)cfsu		5.00	14.00	0.00	1.30	20.30	

The productivity rate chosen to calculate effort: 7.2 hours / cfsu.
The base effort is 146 hours (20.30 cfsu * 7.2 h/cfsu).

Application of Functional Sizing in Maintenance Projects

Outsourcing
In this case a utility company has a system operational and a software services company has carried out the maintenance for more 10 years. Activities include software repair (bug fixing), software enhancement, help desk and knowledge maintenance.
The management of the utility company wants to get an insight into the performance of the software services company and to get a grip on costs. Sogeti was asked to assess the current contract between both parties and to draw up a blue print for a new contract between the two parties in which pricing would be based upon delivered performance.
The first step was to determine the size of the application. Because the system had been operational for almost 15 years and was not well documented, the sizing was done based on the user manual and the operational application itself. The size agreed upon was 6,900 fp. Time spent for sizing was 104 hours including preparation and reporting.
The performance analysis was based upon a comparison between some previous projects and the corresponding invoices, this analysis took about 40 hours. For drawing up the blue print of the new contract another 16 hours were needed. In total 168 hours was spent to get to achieve a contract that was acceptable to both parties.
The basis of the delivered performance was a productivity rate of 8.0 hours / fp. Because of the architecture of the system (modelling and reusable routines) the productivity rate was fixed at 6.5 hours / fp in the contract. The size of releases is the size according to the Sogeti method to measure enhancements projects. The invoice of the software services company should state the delivered size in mfp. If required Sogeti can audit the size of the enhancement project.
For the other activities performance indicators on a yearly basis were agreed upon. For knowledge maintenance, corrective maintenance and helpdesk the following performance indicators will be used: respectively 0.15 h/fp, 0.10 u/fp and 0.10 u/fp. If the utility company was to outsource these activities to the same supplier, one can expect synergetic advantages and work with an all-in indicator of 0.3 h/fp.
After one year the performance indicators will be reviewed and if necessary updated. After the first year the all-in indicator was updated to 0.35 h/fp.

Average maintenance costs decreased by almost 10% and customer satisfaction increased. The latter was caused by the fact that estimated delivery time per enhancement project was more accurate and realistic.

Release management
The IT department of the public organisation has to provide three releases a year. Due to budget limitations these releases have to be delivered with the available staff. The business departments and IT management were not happy with the release process, there were always problems getting the release ready in time. Most of the releases did not contain all of the agreed functionality. This had an adverse effect upon the next release. Introducing functional size measurement could help to make the release process more manageable.
Three previous releases were sized with the Sogeti maintenance approach. With the size delivered and the hours spent the productivity rate was derived: 12.5 h/fp. Based on that, the number of (m)fp that can be delivered in one release was calculated. The available capacity was 128 man months per release, one man month is 120 hours (21.75 days * 8 h/day * 0.7 effective). To take summer holidays into account the IT department calculated with a man month of 110 hours for the summer release and 125 hours for the other three releases (fits nice with productivity rate as well). Experience in the last three releases showed that about 10% of the time was spent on maintenance of the previous release and about 15% was on “emergency changes”. The support of acceptance test and production test takes another 5%. This means only 70% of the time was effectively available for a release. In a regular release one man month equals 7 (m)fp, with 128 man months this is approximately 900 fp. The pilot release was limited to 800 fp.
At first the business departments showed little confidence and were not pleased. The users had to agree upon a smaller than desired release and were aware of previous experiences. When the pilot release was delivered without the usual stress and contained all the agreed functionality, the departments became very positive. The four subsequent releases showed the same results. Due to downsizing of the IT department the releases are now smaller but because of improved productivity (11.2 h/fp) the releases contain sufficient functionality and match the users expectations. All in all the users are more satisfied than before and the release management process is under control.

Conclusions

With the extensions to the size measurement methods, Function Point Analysis and COSMIC Full Function Point, described in this paper measurement of an enhancement project is possible. The productivity rates from new development can be used in enhancement projects because the valuation of changes is relative to new development.
Over 10 years of experience in applying the methods in enhancement projects has proved to benefit principal, user and supplier.

References

[1] NESMA, “Definitions and counting guidelines for the application of function point analysis A practical manual, version 2.0”, Netherlands Software Measurement Association, 1996, http://www.nesma.org

[2] A. Abran, J.M. Deshairnes, S. Oligny, S. St-Pierre, C. Symons, “COSMIC-FFP Measurement Manual – Version 2.2, The COSMIC Implementation Guide for ISO/IEC 19761: 2003”, École de Technologie supérieure – ETS, Montreal, 2003, http://www.lrgl.uqam.ca/cosmic-ffp/casestudies/

[3] ISBSG, “The ISBSG Estimation, Benchmarking & Research Suite (release 8)”, International Software Benchmark Standards Group, 2003, http://www.isbsg.org.au

[4] B.P. Lientz, E.B. Swanson, "Software Maintenance Management: A Study of the Maintenance of Computer Application Software in 487 Data Processing Organizations", Addison-Wesley, 1980, ASIN: 0201042053
[5] Ton Dekkers, "Van Functiepuntanalyse naar Integrale Functiepuntanalyse", IP/Informatica Projectgroep, 1994, ton.dekkers@sogeti.nl
[6] NESMA, "Rapport Functiepuntanalyse in onderhoud en beheer, functionele en technische richtlijnen, versie 1.0", Netherlands Software Measurement Association, 1996, http://www.nesma.org
[7] NESMA, "Function Point Analysis for Software Enhancement", Netherlands Software Measurement Association, 2001, http://www.nesma.org

Author

Ton Dekkers has been working as a practitioner, manager and consultant within the area of software metrics and software quality for a great number of years. Within this area he specialises in estimation, performance measurement, risk analysis, priority management and QMap (Quality Management approach). He is a regular speaker both at national and international conferences and a trainer in software estimation, risk management and Quality Tailor-Made (QTM – QMap in practise).
Ton Dekkers is senior project consultant of the division Engineering & Projecten within Sogeti Nederland B.V. He is responsible for the Expertise Centre Metrics and R & D in the area of Estimating & Performance measurement.
Outside Sogeti Ton is chair of the NESMA workgroups New Technology and COSMIC.