E&Q: An Early & Quick Approach to Functional Size Measurement Methods

M. Conte, T. Iorio, R. Meli & L. Santillo (DPO, Italy)

Abstract

The application of software functional size metrics, as IFPUG Function Points and COSMIC Full Function Points, frequently reveals serious difficulties arising from the lack of a detailed and complete description of the functional user requirements of the systems being measured. This kind of difficulties, combined with the obvious need to provide an estimation of the measures in a reduced and early time frame, compared to the standard measurement duration and time, led to the definition of the Early and Quick (Function Pont) technique. This work describes the generalized definition of the technique and how its structure and concepts are specialized for the above measurement methods, providing all the required information for applying it in practice. Moreover, we provide the goodness evaluation method and results of the technique with respect to the estimated and actual measures, through a given set of numerical indicators.

1. Introduction

The Early & Quick technique was originally proposed in 1997 for IFPUG Function Point Analysis [1], to make possible to size software projects or systems starting from non-detailed information, typically with reduced effort and time with respect to standard measurements. This non-detailed information is commonly found in early phases of software lifecycle, when not every item – required for standard measurement – is available (e.g. list of processed fields and referenced files, read and/or written by the system's transactions). In recent years, and given the standardization of the definition of software functional measurement by ISO/IEC, the Early & Quick technique has evolved and has been generalized, extending its applicability domain to the COSMIC-FFP measurement method [2] and taking advantage of enhancement opportunities derived from local or global statistical analysis, as the ISBSG benchmark [3, 4].

2. Referring Standards and Definitions

The basic concepts of software functional size measurement have been standardized by JTC 1 / SC 7 (Joint Technical Committee 1 / SubCommittee 7) of ISO/IEC in 1998 [5]. Among them we find:

- Functional User Requirements (FUR): a sub-set of the user requirements. The Functional User Requirements represent the user practices and procedures that the software must perform to fulfil the users' needs. They exclude Quality Requirements and any Technical Requirements.
- Base Functional Component (BFC): an elementary unit of Functional User Requirements defined by and used by a Functional Size Measurement (FSM) Method for measurement purposes.

Once established those fundamental concepts, every measurement method, as IFPUG Function Point Analysis and COSMIC Full Function Point, can identify its own BFCs' equivalents:

- Elementary processes (external inputs, outputs and inquiries) and logical files (internal and external) for IFPUG;
- Data movements (Entries, eXits, Reads and Writes) for COSMIC.

It's worth noting that the COSMIC method identifies BFCs at a more elementary level with respect to the IFPUG method: the COSMIC equivalent of the elementary process (the "functional process") is defined as "an elementary component of a set of Functional User Requirements comprising a unique cohesive and independently executable set of data movement types" [2]. Any measurement approach, and hence any early estimation technique – cannot ignore the essential concept of BFC, or of BFCs functional aggregations. Consequently, any estimation technique which is not based on those concepts – as for example backfiring from source lines of code – is not to be considered satisfactory.

3. The Early & Quick Estimation Technique

3.1. Concepts

The Early & Quick (E&Q) functional size estimation technique is a consistent set of concepts and procedures which, even though applied to non-detailed system or project information, maintain the overall structure and the essential concepts of standard functional size measurement methods.

The E&Q technique combines different estimating approaches in order to provide better estimates of a software system functional size: it makes use of both analogical and analytical classification of functions (transactions and data). Moreover, it permits the use of different levels of detail for different branches of the system (multilevel approach): the overall global uncertainty level in the estimate (which is a range, i.e. a set of minimum, more likely, and maximum values) is the weighted sum of the individual components' uncertainty levels. Finally, the technique provides its estimates through statistically and analytically validated tables of values. The technique focuses on the fundamental principles reported in Tab. 1.

Table 1. E&Q fundamental principles.

Principle	**Explanation**
Classification by analogy	Similarity by size and/or overall functionality between new and known software objects.
Structured aggregation	Grouping of a certain quantity of lower level software objects in one higher level software object.
No given function/data correlation	Data and transactional components assessed autonomously.
Multilevel approach	No discard of details, if available – no need of details, if unavailable.
Use of a derivation table	Each software object at each detail level is assigned a size value, based on an analytically / statistically derived table.

The starting point of the technique is the product breakdown structure of the system being studied, whose basic elements are the following software objects:

- logical data groups, and
- elementary functional processes,

that is, the BFC elements or data movement aggregations of any standard measurement method (IFPUG, COSMIC, and so on).

Further aggregations, as depicted in Fig.1, are provided[1]:

- elementary logical data groups can be grouped in multiple data groups;
- elementary functional processes can be grouped in small, medium or large "typical" and "general" software processes;
- general processes can be grouped in small, medium or large "macro" software processes.

[1] Please note the changed terminology with respect to previous publications: "process" used at all levels of aggregation, instead of "function", and "typical process" instead of "micro-function".

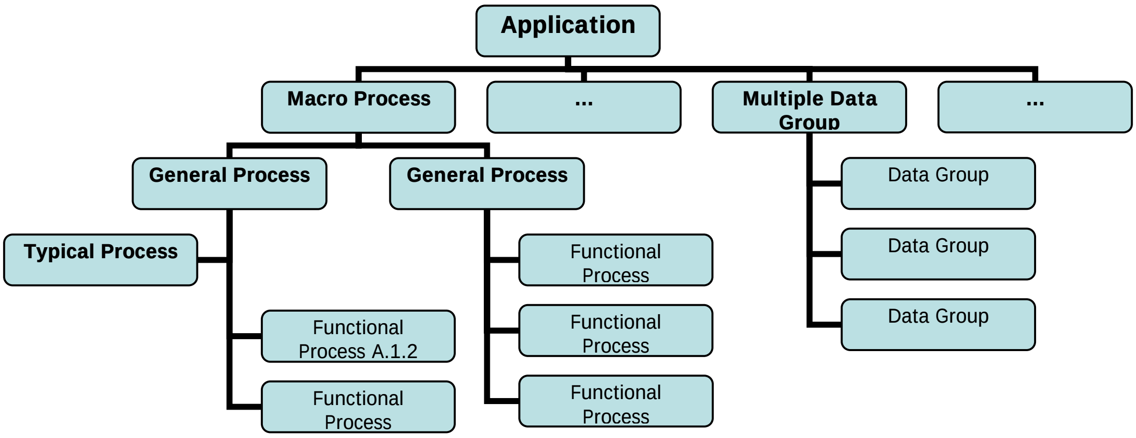

Figure 1. Functional hierarchy in the E&Q estimation technique (for sake of illustration, only one instance of Macro Process and one instance of Multiple Data Group are shown).

Each "software object" is assigned a set of size values (minimum, most likely, maximum) based on statistical/analytical tables, then the values are summed up to provide the overall estimation result (minimum, most likely, maximum). To obtain the estimated size of the software application or project being studied, a list of its processes and data groups is the only required item, even comprising non-homogeneous levels of detail. Knowledge of similar software objects will make it possible to assign the right level of classification to each element on the list, and therefore to derive its contribution to the overall size. The estimation uncertainty (given by the minimum-maximum value range) will be obviously greater for the higher levels of software objects aggregation, taking into account the higher lack of details. On the opposite, it can be sometimes more significant to make use of the higher levels to reduce the estimation time and to avoid to get together a large number of smaller elements, each of which comes with its own uncertainty.

Tab. 2 below shows the conceptual correspondences between the E&Q software objects and the BFCs and concepts of specific measurement methods. Specific application instances of the E&Q estimation technique for the cited methods are presented in further sections.

Table 2. Conceptual correspondences between E&Q, IFPUG and COSMIC methods.

E&Q	**IFPUG**	**COSMIC**
Application	Bounded application	Bounded application / Layer (*)
Macro Process	-	-
General Process	-	-
Typical Process	-	-
Functional Process	Elementary Process	Functional Process
-	-	Data Movement
-	Data Element (I/O)	-

E&Q	**IFPUG**	**COSMIC**
Application	Bounded application	Bounded application / Layer (*)
Multiple Data Group	-	-
Logical Data Group	Logical File (Internal or External)	Object of Interest (**)

(*) The COSMIC method permits to size different layers of software within the same application boundary; each layer, with respect to the measurement and size estimation process, can be treated as a unique system.

(**) The COSMIC method defines and makes use of the conceptual equivalent of data group, but does not assign any size value to it (up to version 2.2, 2003).

From now, for sake of brevity, the acronyms reported in Tab. 3 will be used.

Table 3. Acronyms.

Acronym	**Description**	**Brief definition**
MP	Macro Process	A set of two or more average GP's. It can be likened to a relevant sub-system, or even a bounded application, of an overall Information System. Its size is evaluated based on the (estimated) quantity of included GP's.
GP	General Process	A set of two or more average FP's. It can be likened to an operational sub-system, which provides an organised whole response to a specific application goal. Its size is evaluated based on the (estimated) quantity of included FP's.
TP	Typical Process	A particular case of a GP: the set of the most frequent operational transactions on a data group or a small set of data groups. Usually associated with the term "Management of [object of interest]". Basically, it can be found in two "flavours": CRUD (Create, Retrieve, Update and Delete), or enlarged management (CRUD plus List – CRUDL – and a standard Report with total values).
FP	Functional Process	The smallest software process with autonomy and significance characteristics. It allows the user to achieve a unitary business or logical objective at the operational level.
MDG	Multiple Data Group	A set of two or more LDG's. Its size is evaluated based on the (estimated) quantity of included LDG's.
LDG	Logical Data Group	A group of logical data attributes, representing a conceptual entity which is functionally significant as a whole for the user.

3.2. On Implied Functional Processes

The main difficulty in applying an estimation method to the software functional size is that of identifying the so-called "implied" functions. The typical example is that of implied inquiries (e.g. list-boxes in a GUI environment): even when performing a standard measurement, the measurer not always finds the documented presence of such functionality, which in turn is to be fully counted as significant functional processes (given that they are user-significant and populated from one or more logical data groups). In practice, very often the user requirements fail to reach such a detailed level of description - or indicate once and for all, in a generic manner, that "where possible" such functionality will be present. The measurer is left with the task of establishing how many, and which, of these implied functional processes are to be actually measured, prior to implementation. This problem is even magnified when applying the E&Q estimation technique. Evolving from previous versions, the E&Q technique provides two alternatives to solve that:

- estimating the presence and quantity of implied functional processes (eventually assigning specific estimated values for such kind of functionality), or
- associating an estimated average size with one or more types of other software objects (as for example to Functional Processes, Typical Processes or Data Groups).

3.3. Reliability Indicator

The reliability indicator I provides a numerical evaluation of the goodness of the estimation with respect to the corresponding measurement method. This indicator does not express the variability range, but rather the deviation (a posteriori) between the actual measured size value M and the estimation range (S_{min}, S_{likely} e S_{max}).

The indicator is defined for non-zero ranges ($S_{min} \neq S_{max}$) – for the estimation of a single system/project i – by the following formula:

$$I_i = \frac{(S_{\max} - S_{\min}) - |M - S_{likely}|}{(S_{\max} - S_{\min})}$$

The indicator has the following features:

- I_i has a threshold value for M equal to one opf the range extremis;
- I_i gets worse for M going externally of the estimation range, and vice versa;
- I_i gets better for smaller ranges (S_{max} - S_{min});
- I_i gets better for smaller differences between M and S_{likely} and yields the best value for $M = S_{prob}$.

Fig. 2 shows a san example the form of I_i, with fixed values S_{min} = 10, S_{prob} = 13, S_{max} = 20 and actual measured value M varying on that range. In the best case ($M = S_{prob}$) we find I_i = 1 (maximum); in the extremis ($M = S_{min}$ o $M = S_{max}$), we find the threshold I = 0.5; for bad estimations (M external to the range), I_i < 0.5. Hence, the expected value of the reliability indicator, for a satisfactory estimation technique, is between 0.5 and 1.

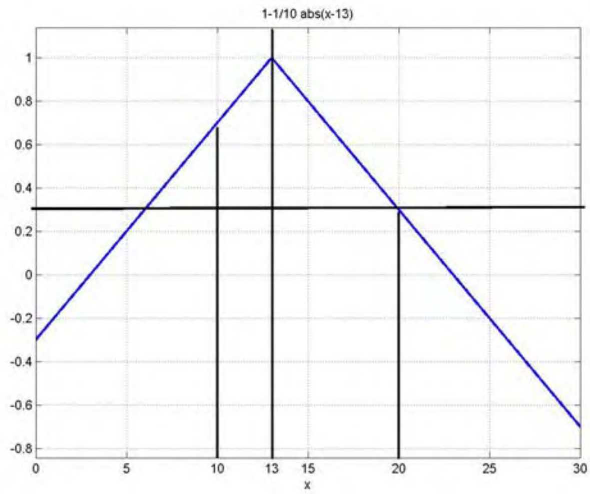

Fig. 2 Form of I_i with respect to the actual measured size M.

The overall reliability indicator of the estimation technique is given by the average I over N cases. Thus, the average reliability indicator provides, for future estimations, an evaluation of the associated "risk".

4. Early & Quick IFPUG Function Point (E&QFP 2.0)

Release 2.0 of the E&Q technique, applied to the IFPUG Function Point method 4.x, is an evolution of the technique, as it was published by the authors in previous works [7, 8]. Indications for such evolution came from:

- statistics derived from the ISBSG benchmark [3, 4], regarding projects measured with IFPUG 4.0 e 4.1 – precisely, on the average Unadjusted FP values per function type e the most frequent or most likely function breakdown (average profile);
- experiences and results derived when applying the technique in several (compared) estimation and measurement tasks (*N* >> 30) in the last 18 months;
- the introduction of a new software object (extended typical process);
- reviewed evaluation of the ranges of aggregated processes types ("small", "medium" and "large"), not excluding overlapping ranges;
- the extension of data object types, when it's possible to provide their exact type (internal or external).

4.1. IFPUG Data Functions and Their E&Q Equivalent

Logical Data Groups can be identified for ILF's (Internal Logical Files) and EIF's (External Interface Files), whenever a conceptual entity is found that is significant for the user of the examined system. These groups can be classified on a multiple scale of complexity:

- Simple, Average and Complex, or generic (unspecified type),
- Simple, Average and Complex, or generic (specified type – Internal or External),
- Low and High Multiplicity.

The first three levels (simple, average and complex – specified and unspecified) correspond exactly to those considered in the IFPUG method, while the other two are for particularly complex macro-files, grouping several distinct logical files. For the first three levels, the IFPUG complexity tables can be used, if detailed information is available about data and record element types. For the remaining levels, ranges of likely contained IFPUG logical files are provided. The difference between ILF's (maintained) and EIF's (referenced, but maintained by other systems) can be neglected ("unspecified type") or taken into account ("specified type"); the latter case, if identified, can provide the most accurate estimation values.

4.2. IFPUG Transactional Functions and Their E&Q Equivalent

Functional Processes correspond to the elementary processes of the standard IFPUG method – i.e. EI's, EO's and EQ's (respectively External Inputs, Outputs and Inquiries), while Typical Processes, General Processes and Macro Processes are higher aggregations of Functional Processes. Accordingly:

- Functional Processes can be classified as Input (FPI), Output (FPO) or Query (FPQ),
- Typical, General and Macro Processes are classified as small, average or large, depending on the estimated quantity, or list, of their subordinate components,

Due to the assigned minimum and maximum estimated values, there's no need of evaluating the functional complexity of such processes, leading to a significant gain of measurement effort and time.

Implicit processes – e.g. listboxes – can be treated in two alternative ways:

- Direct estimation (one simple query per estimated instance)
- Derived estimation via an average correlation from the quantity of data groups (since each query must be populated by logical data).

4.3. Ranges and Numerical Assignments

Each E&QFP element is assigned three estimated values (Unadjusted FP), i.e. minimum, most likely and maximum UFP (the Value Adjustment Factor is neglected here, since it doesn't affect the overall estimation technique). Aggregated elements, as multiple data groups and general and macro processes are classified according to the ranges of their (estimated) subordinate components. Tab. 4 below reports both the components ranges and the numerical assignments for the E&QFP 2.0 estimation technique[2].

Table 4. E&QFP 2.0 components ranges and numerical assignments.

Type	Level	Ranges / IFPUG Equivalent	Min. UFP	Most likely UFP	Max. UFP
LDG (unspecified)	Simple	Low complexity ILF/EIF	5.0	6.5	7.0
	Average	Medium complexity ILF/EIF	7.0	9.2	10.0
	Complex	High complexity ILF/EIF	10.0	13.7	15.0
	Generic	Generic complexity ILF/EIF	5.0	6.9	15.0
Internal LDG	Simple	Low complexity ILF	7.0	7.0	7.0
	Average	Medium complexity ILF	10.0	10.0	10.0
	Complex	High complexity ILF	15.0	15.0	15.0
	Generic	Generic complexity ILF	7.0	7.4	15.0
External LDG	Simple	Low complexity EIF	5.0	5.0	5.0
	Average	Medium complexity EIF	7.0	7.0	7.0
	Complex	High complexity EIF	10.0	10.0	10.0
	Generic	Generic complexity EIF	5.0	5.5	10.0
Multiple LDG	Low	2–4 generic LDG	10.0	21.0	30.0
	High	5–8 generic LDG	27.0	45.0	60.0
Funct. Process	Input	EI	3.0	4.3	6.0
	Output	EO	4.0	5.4	7.0
	Query	EQ	3.0	3.8	6.0
Implicit Process	Direct	Low/Medium EQ	3.0	3.3	4.0
	Derived	One avg. per each LDG (*)	2.7	3.0	3.3
Typical Process	Small	CRUD (Low/Medium complexities); CRUD + List (Low complexities)	12.0	14.0	16.7
	Medium	CRUD (Medium/High complexities); CRUD + List (Medium complexities); CRUD + List + Report (Low cplx's)	15.8	17.8	20.5
	Large	CRUD (High complexities); CRUD + List (Medium/High cplx's); CRUD + List + Report (Medium cplx's)	21.2	23.2	25.9
General Process	Small	6–10 generic FP's	22.0	37.0	57.0
	Medium	10–15 generic FP's	37.0	57.0	81.0
	Large	15-20 generic FP's	57.0	81.0	110.0
Macro Process	Small	2–4 generic GP's	75.0	170.0	325.0
	Medium	4–6 generic GP's	150.0	285.0	485.0
	Large	6–10 generic GP's	220.0	455.0	810.0

(*) Multiply by factor 3 or 6,5, respectively, in case of Low or High Multiplicity LDG's.

[2] Please note with respect to previous publications: the innovative distinction of Typical Processes (formerly: "Micro-functions") into Small, Medium, and Large, the updated ranges, particularly for General Processes (formerly: "Functions"), and the updated numerical assignments over all the table, and specifically for Macro Processes (formerly: "Macro-functions").

4.4. Example of Application

At conference time, an example will be available for download from DPO's website (www.dpo.it) in order to illustrate in an exhaustive manner the application of the E&QFP 2.0 estimation technique on a complete system.

4.5. Reliability Indicator for E&QFP 2.0

Tab. 5 contains an excerpt of comparisons of estimated and actual FP values, including the reliability indicator values. The average reliability indicator for the observed sample is $I = 0.72$ (> 0.5), thus denoting the good reliability of the new release of the technique applied to the IFPUG method.

Table 5. Reliability indicator values for a mixed sample (DPO).

System	Min. UFP	Most likely UFP	Max. UFP	Actual size (UFP)	Reliability Indicator
S1	336.0	412.0	550.0	348	0.70
S2	251.0	308.0	399.0	207	0.32
S3	482.0	591.0	782.0	705	0.62
S4	378.0	463.0	601.0	415	0.78
S5	135.0	166.0	217.0	154	0.85
S6	387.0	474.0	631.0	400	0.70
S7	401.0	496.0	600.0	479	0.91
S8	183.0	203.0	238.0	189	0.75
S9	96.0	104.5	115.0	107	0.87
S10	736.0	856.0	938.0	825	0.85
S11	113.0	148.0	202.0	160	0.86
S12	113.0	135.0	173.0	168	0.45
S13	72.0	91.1	120.0	77	0.71

5. Early & Quick COSMIC Full Function Point (E&QCFFP 2.0 proposal)

Release 2.0 of the E&Q technique, applied to the COSMIC Full Function Point method 2.2, is a new proposal, following the general approach illustrated in [2] and originally started in [9]. Indications for such evolution came mainly from:

- experiences and results reported by practitioners
- analogy with the IFPUG instance of the E&Q technique

Due to the ongoing research and trial, the following is to be considered only a proposal of the E&Q technique applied to the COSMIC measurement method. Further tests will provide adjustments or eventually definitive confirmation of the proposal and the figures within.

5.1. COSMIC Data Functions and Their E&Q Equivalent

In the actual COSMIC-FFP release, Object of Interests are identified, but not assigned any numerical values. Thus, the E&Q equivalents can be easily identified, but do not receive any numerical valued ranges. As for the IFPUG application, data items identification could help in estimating collateral processes (as the implied functional processes) – this issue is still under study.

5.2. COSMIC Transactional Functions and Their E&Q Equivalent

Functional Processes correspond exactly to the functional processes of the standard COSMIC method, with no distinction in type. However, since the COSMIC method is not bounded in the numerical size that can be assigned to any functional process, it's been found adequate to distinguish scale levels of the estimated processes, based on a quartile statistical distribution of the size of functional processes, measured in CFSU (COSMIC Functional Size Unit); every read, write, entry or exit data movement within each process is assigned the conventional value of 1 CFSU.

Moreover, in exact analogy with the IFPUG application, Typical Processes and higher level aggregations (General and Macro Processes) are well-defined for the COSMIC domain.

5.3. Ranges and Numerical Assignments

Each E&QCFFP element is assigned three estimated values (CFSU), i.e. minimum, most likely and maximum estimated size. Aggregated elements, as typical, general and macro processes are classified according to the ranges of their (estimated) subordinate functional processes. Tab. 6 below reports both the components ranges and the numerical assignments for the E&QCFFP 2.0 Release Candidate estimation technique.

Table 6. E&QCFFP 2.0 RC components ranges and numerical assignments.

Type	Level	Ranges / COSMIC Equivalent	Min. CFSU	Most likely CFSU	Max. CFSU
Funct. Process	Small	1-5 Data movements	2.0	3.9	5.0
	Medium	5-8 Data movements	5.0	6.9	8.0
	Large	8-14 Data movements	8.0	10.5	14.0
	Very Large	14+ Data movements	14.0	23.7	30.0
Typical Process	Small	CRUD (Small/Medium processes); CRUD + List (Small processes)	15.6	20.4	27.6
	Medium	CRUD (Medium/Large processes); CRUD + List (Medium processes); CRUD + List + Report (Small processes)	27.6	32.3	42.0
	Large	CRUD (Large processes); CRUD + List (Medium/Large processes); CRUD + List + Report (Medium proc's)	42.0	48.5	63.0
General Process	Small	6–10 generic FP's	20.0	60.0	110.0
	Medium	10–15 generic FP's	40.0	95.0	160.0
	Large	15-20 generic FP's	60.0	130.0	220.0
Macro Process	Small	2–4 generic GP's	120.0	285.0	520.0
	Medium	4–6 generic GP's	240.0	475.0	780.0
	Large	6–10 generic GP's	360.0	760.0	1,300.0

5.4. Example of Application

At conference time, an example will be available for download from DPO's website (www.dpo.it) in order to illustrate in an exhaustive manner the application of the E&QCFFP 2.0 Release Candidate estimation technique on a complete system.

6. Conclusions

"Estimating" means using less time and effort in obtaining an approximate value of size. The advantages of an estimation technique are obviously counterbalanced by an unavoidable minor accuracy. Therefore, we should always strongly distinguish between different terms and concepts: "counting (Function Points)" means measuring software size through the use of the standard practices (IFPUG, COSMIC, and so on), while "estimating (Function Points)" denotes an approximate evaluation of the same size through other different means. Since the E&Q results must not be considered as accurate measures, in all cases where an inspection or measure comparison is requested (i.e. when litigation is a possibility), a subsequent exact measurement is necessary, and the standard measurement practices (IFPUG, COSMIC, and so on), must be used.

With respect to a typological classification of estimation methods [6], the E&Q technique is a mixed-type method: it is based on (direct) analogy and (derived) analysis. The former lets the estimator discover similarities between a new "piece" of a software application and similar "pieces" encountered and already classified in other software applications. The latter guarantees a certain grounding and stability for the estimate, since the weights of the various software objects are both based on global statistical analysis and conceptually established, as they are connected with the way in which the various software objects in the classification structure are constructed. Moreover, the E&Q technique fully complies with the concepts definitions and the structure of any functional size measurement method, as defined by ISO/IEC [1]. From this point of view, the E&Q technique can be regarded as a general estimation technique, which can be extended and applied to any FSM method that is found to be compliant with the ISO/IEC standard.

The reliability of the E&Q technique, as any other human-based estimation method, is obviously proportional to the estimator's ability to "recognize" the components of the system as part of one of the proposed classes. This ability may sharpen through practice by comparing the different counts obtained using standard versus E&Q practices. However, in the authors' experience as trainers and consultants, even the initial application of the technique by new users provides encouraging results, which can be taken as valid starting estimated values. In any case, the E&Q size estimation technique has proved to be quite effective, providing a response within ± 10% of the real size in most real cases, while the savings in time (and costs) can be between 50% and 90% (depending on the comprised aggregation level – from Functional Processes to Macro Processes) with respect to corresponding standard measurement procedures.

7. References

[1] IFPUG, "Function Point: Manuale delle Regole di Conteggio, Versione 4.1.1", IFPUG, Aprile 2000 (ed. originale: "Counting Practices Manual, Release 4.1.1", April 2000).

[2] COSMIC, "Manuale di Misurazione COSMIC-FFP, Versione 2.2", COSMIC, Ottobre 2003 (ed. originale: "COSMIC-FFP Measurement Manual, Version 2.2", January 2003).

[3] ISBSG, "The Benchmark Release 5", ISBSG, 1998.

[4] ISBSG, "The ISBSG Estimating, Benchmarking & Research Suite Release 8" (Data CD), ISBSG, 2003.

[5] ISO/IEC, "14143-1:1998 'Information technology – Software measurement – Functional size measurement - Part 1: Definition of Concepts'", JTC 1 / SC 7, ISO/IEC, 1998.

[6] Meli, R. and Santillo, L., "Function Point Estimation Methods: a Comparative Overview", in FESMA '99 Conference proceedings, Amsterdam, 4-8 October, 1999.

[7] Santillo, L. and Meli, R. "Early Function Points: some practical experiences of use", in ESCOM-ENCRESS '98 Conference proceedings, Rome, 27-29 May, 1999.

[8] Meli, R., "Early and Quick Function Point Analysis from Summary User Requirements to Project Management", in: IFPUG, "IT Measurement – Practical Advice from the Experts", Chapter 26, pp. 417-440, Addison-Wesley, 2002.

[9] Meli, R., Abran, A., Ho, V.T. and Oligny, S., "On the applicability of COSMIC-FFP for measuring software throughout its life cycle", in ESCOM 2000 Conference proceedings, Munich, 18-20 April, 2000.