

Early & Quick Function Point: Sizing More with Less

Luca Santillo, Massimiliano Conte, Roberto Meli
DPO (Data Processing Organization), Italy
luca.santillo@dpo.it, massimiliano.conte@dpo.it, roberto.meli@dpo.it

Abstract

The Early & Quick technique was originally proposed in 1997 for IFPUG Function Points, to size software in early stages of the development process, when functional requirements are still to be established in a detailed form and/or when a rapid measure is needed for existing software from a high-level viewpoint, within limited time. Typical lack of measurement details and requirements volatility in early project stages are overcome by the E&Q approach to provide a size estimate as a significant contribution to early project planning needs. Fundamental principles of the technique are classification by analogy, functionality structured aggregation, and multilevel approach, with statistical validation of numerical ranges. Recently, the technique has evolved, to fully comply with any Functional Size Measurement method (ISO/IEC 14143:1998), so to cover new generation methods (e.g. COSMIC Full FP 2.2) and updated releases of existing methods (e.g. IFPUG FP 4.1 and 4.2). This paper describes the current technique release 2.0, application cases, validation results, supporting tools, and further improvement directions.

1. Introduction

The Early & Quick technique was originally proposed in 1997 for IFPUG Function Point Analysis [7], 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 [9] and taking advantage of enhancement opportunities derived from local or global statistical analysis, as the ISBSG benchmark [4, 5].

2. Referring Standards and Definitions

The basic concepts of software functional size measurement have been standardized by ISO/IEC in 1998 [6]. Among them we find:

- Functional User Requirements (FURs): a sub-set of the user requirements. FURs represent the user practices and procedures that the software must perform to fulfil the users' needs. They exclude Quality Requirements and any Technical Requirement.
- 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 FSM method;
- data movements (Entries, eXits, Reads and Writes) for COSMIC FSM method.

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 FURs comprising a unique cohesive and independently executable set of data movement types" [1]. 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 Table 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 FSM method. Further aggregations are provided (Figure 1):

- several elementary logical data groups can be grouped into one or more multiple data groups;
- several elementary functional processes can be grouped into one or more small, medium or large “typical” and “general” software processes;
- several general processes can be grouped into one or more small, medium or large “macro” software processes.

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

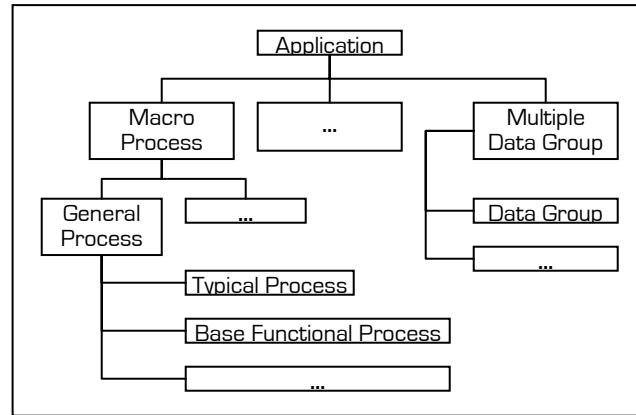


Figure 1. Functional hierarchy in E&Q technique.

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 makes 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) is obviously larger for higher level aggregations, carrying 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.

Conceptual correspondences are present between the E&Q software objects and the concepts of specific measurement methods (Table 2); note that COSMIC layers can be estimated as separate subsystems. Specific application instances of the E&Q estimation technique for the cited methods are presented in further sections.

Table 2. Conceptual links (E&Q/IFPUG/COSMIC).

E&Q	IFPUG	COSMIC
Application	Bounded application	Bounded application / Layer
Macro Process	-	-
General Process	-	-
Typical Process	-	-
Functional Process	Elementary Process	Functional Process
-	-	Data Movement (DM): E, X, R, W for Entry, eXit, Read, and Write
	Data Element Type (DET)	-
Multiple Data Group	-	-
Logical Data Group	Logical File (Internal or External)	Object of Interest

From now on, E&Q acronyms will be used (Table 3).

Table 3. E&Q acronyms explanation.

Acronym	Name	Description
BFP	Base 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.
IFP	Implied Functional Process	Listbox-like and other typical functions, which may be not explicitly documented.
TFP	Typical Functional Process	A particular case of GFP: a set of the most frequent operations on a data group. Usually denoted as "Management of [object of interest]". It can be found in two "flavors": CRUD (Create, Retrieve, Update and Delete), or CRUDL (CRUD + List, or standard Report).
GFP	General Functional Process	A set of two or more average BFPs. It can be likened to an operational sub-system, which provides an organized whole response to a specific application goal. Its size is evaluated based on the (estimated) quantity of included BFPs.
MFP	Macro Functional Process	A set of two or more average GFPs. 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 GFPs.
LDG	Logical Data Group	A group of logical data attributes, representing a conceptual entity which is functionally significant as a whole for the user.
MDG	Multiple (Logical) Data Group	A set of two or more LDGs. Its size is evaluated based on the (estimated) quantity of included LDGs.

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 Base Functional Processes, Typical Functional Processes or Logical Data Groups).

3.3. Reliability Indicator

The reliability indicator I_i provides a numerical evaluation of the goodness of a given estimate i with respect to the corresponding measurement method [2]:

$$I_i = 1 - \frac{|M - E_{likely}|}{(E_{max} - E_{min})}$$

where M represents the actual measure value, and E_{min} , E_{likely} , and E_{max} represent the estimation range values (minimum, most likely, maximum, respectively). I_i does not express the estimate variability, but rather the (a posteriori) deviation between the actual measure and the given estimate. Such indicator has the following features:

- I_i has a threshold value for M equal to E_{min} or E_{max} ;
- I_i increases for M belonging to the estimate range;
- I_i decreases for smaller ranges ($E_{max} - E_{min}$);
- I_i increases for E_{likely} closer to M (maximum $I_i = 1$).

The overall reliability indicator of the estimation technique is given by the average I over several cases, providing an evaluation of associated size estimation risk.

4. E&Q Application to IFPUG Case

Release 2.0 of the E&Q technique – applied to the IFPUG Function Point method 4.2 – is an evolution of the technique, since it was published by the authors in previous works [7, 10]. Improvements came from:

- statistics derived from the ISBSG benchmark [4, 5], regarding projects measured with IFPUG 4.0 e 4.1, on average Unadjusted FP values per function type and average functional breakdown profiles;
- field trial results, on full estimation and measurement cases ($N \gg 50$) in the last 2 years;
- the introduction of a new software object (extended typical process, corresponding to the CRUDL);

- reviewed evaluation of the ranges of aggregated processes types (“small”, “medium” and “large”), not excluding overlapping ranges;
- data object types extension, when the exact type is known (internal or external logical files);
- general taxonomy improvement, for ease of reference.

4.1. E&Q Items for Data Functions

Logical Data Groups can be identified for ILFs (Internal Logical Files) and EIFs (External Interface Files), whenever a conceptual entity is found that is significant for the user of the examined system [3]. These groups can be classified on a multiple scale of complexity:

- Low, Average, High, or generic complexity (for unspecified file type),
- Low, Average, High, or generic complexity (for specified file type: ILF or EIF),
- Low and High Multiplicity.

The first three levels 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 (maintained) ILFs and (referenced) EIFs 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. E&Q Objects for Transactional Functions

Base 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) [3], while Typical Processes, General Processes and Macro Processes are higher aggregations of Functional Processes. Accordingly:

- a Base Functional Processes can be classified as an Input, an Output or a Query,
- Typical, General and Macro Functional Processes can be classified as low, average, or high level, depending on the estimated amount of their lower components,
 - Implicit processes – e.g. listboxes – can be treated in two alternative ways:
 - *direct estimation* (one query per estimated instance);
 - *derived estimation* via an average correlation from the quantity of data groups (since any significant query must be populated by identifiable data attributes).

4.3. Ranges and Numerical Assignments

Each E&Q element is assigned three estimated values (expressed as Unadjusted FP), i.e. minimum, most likely and maximum UFP values. The Value Adjustment Factor from the IFPUG method is neglected here, since it doesn’t affect the overall estimation results. Aggregated elements, as multiple logical data groups and general and macro functional processes are classified according to the ranges of their (estimated or listed) subordinate components. Table 4 reports the components numerical assignments for the E&QFP estimation technique (“Early & Quick IFPUG Function Point”), release 2.0.

Table 4. E&QFP 2.0 components.

Type	Level/Cplx	E _{min}	E _{likely}	E _{max}
BFP	Input	3.0	4.3	6.0
	Output	4.0	5.4	7.0
	Query	3.0	3.8	6.0
IFP	Direct	3.0	3.3	4.0
	Derived (per each file)	2.7	3.0	3.3
TFP	Low CRUD/L	12.0	14.0	16.7
	Average CRUD/L	15.8	17.8	20.5
	High CRUD/L	21.2	23.2	25.9
GFP	Low (6-10 BFPs)	22.0	37.0	57.0
	Average (10-15 BFPs)	37.0	57.0	81.0
	High (15-20 BFPs)	57.0	81.0	110.0
MFP	Low (2-4 GFPs)	75.0	170.0	325.0
	Average (4-6 GFPs)	150.0	285.0	485.0
	High (6-10 GFPs)	220.0	455.0	810.0
LDG	Low	5.0	6.5	7.0
	Average	7.0	9.2	10.0
	High	10.0	13.7	15.0
	Generic	5.0	6.9	15.0
ILF	Low	7.0	7.0	7.0
	Average	10.0	10.0	10.0
	High	15.0	15.0	15.0
	Generic	7.0	7.4	15.0
EIF	Low	5.0	5.0	5.0
	Average	7.0	7.0	7.0
	High	10.0	10.0	10.0
MLDG	Generic	5.0	5.5	10.0
	Low (2-4 LDGs)	10.0	21.0	30.0
	High (5-8 LDGs)	27.0	45.0	60.0

4.4. Reliability Indicator for E&QFP 2.0

Updated evaluations on over 40 cases lead to an average value of reliability indicator between 0.7 and 0.8, depending on the aggregation level that mostly used by the estimators. A value of $I > 0.5$ denotes a very good reliability of E&Q estimate for IFPUG Function Point size.

5. E&Q Application to COSMIC Case

The E&Q approach, applied to the COSMIC Full Function Point measurement method, release 2.2, follows from the general approach illustrated in [1, 2] and originally proposed in [9]. Indications for such evolution came mainly from:

- experiences and results reported by practitioners;
- analogy with the IFPUG case;
- statistical adjustments by software domain [11].

The following is considered with “Release Candidate (RC)” status; validations will provide refinements to the proposed figures and ranges and explicit reliability indicator evaluations.

5.1. E&Q Objects for Data Functions

In the actual COSMIC-FFP release [1], Object of Interests (“OoI”) 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 case, data items identification could help in estimating collateral processes (as the implied functional processes).

5.2. E&Q Objects for Transactional Functions

Base Functional Processes exactly correspond 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 given amount of functional processes by size scale, measured in COSMIC Functional Size Unit; every Entry, eXit, Read, and Write data movement within each process is assigned the conventional value of 1 CFSU [1].

In exact analogy with the IFPUG case, Typical Functional Processes and higher level aggregations (General and Macro Functional Processes) are well-defined for the COSMIC case.

5.3. Ranges and Numerical Assignments

Each E&Q 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. Tables 5a and 5b report the components numerical assignments for the E&QFFP estimation (“Early & Quick Full Function Point”), candidate release 2.0, for Business Application Software (BAS) and Realtime (RT) domains.

Other domains could further apply, with different ranges and figures (to collect and analyse). As an example of the heuristics process used to obtain figures for aggregated values, note that – in Tab. 5a – for a Low GFP (6-10 BFPs) E_{min} is approx. 6 times the E_{likely} value for a Low BFP, E_{likely} is approx. 8 times the E_{likely} value for an Average BFP, and E_{max} is approx. 10 times the E_{likely} value for a High BFP (the Very High BFP case is considered statistically “rare”, that is on the tail of frequency distribution, with respect to the first three levels, when estimated in aggregated GFPs).

Table 5a. E&QFFP_BAS 2.0RC components.

Type	Level/Cplx	E_{min}	E_{likely}	E_{max}
BFP	Low (2-5 DM)	2.0	3.6	5.0
	Average (5-8 DM)	5.0	6.3	8.0
	High (8-14 DM)	8.0	10.5	14.0
	Very High (14+ DM)	14.0	18.0	25.0
IFP	Direct (= Low BFP)	2.0	3.0	4.0
	Derived (per each Ool)	1.8	2.7	3.6
TFP	Low CRUD/L	14.4	18.0	25.2
	Average CRUD/L	25.2	30.0	42.0
	High CRUD/L	42.0	50.0	65.0
GFP	Low (6-10 BFPs)	20.0	50.0	100.0
	Average (10-15 BFPs)	40.0	80.0	160.0
	High (15-20 BFPs)	55.0	110.0	210.0
MFP	Low (2-4 GFPs)	60.0	150.0	300.0
	Average (4-6 GFPs)	200.0	400.0	800.0
	High (6-10 GFPs)	440.0	880.0	1,600.0

Table 5b. E&QFFP_RT 2.0RC components.

Type	Level/Cplx	E_{min}	E_{likely}	E_{max}
BFP	Low (2-3 DM)	2.0	2.5	3.0
	Average (3-5 DM)	3.0	4.0	5.0
	High (5-10 DM)	5.0	7.5	10.0
	Very High (10+ DM)	10.0	15.0	20.0
GFP	Low (6-10 BFPs)	15.0	32.0	75.0
	Average (10-15 BFPs)	25.0	38.0	110.0
	High (15-20 BFPs)	38.0	70.0	150.0
MFP	Low (2-4 GFPs)	64.0	115.0	280.0
	Average (4-6 GFPs)	130.0	190.0	420.0
	High (6-10 GFPs)	190.0	310.0	700.0

6. Supporting tools

Two separate software tools have been developed, supporting the E&Q technique: the *Sfera* application and the *requestimate* addin for *Requisite Pro*®. *Sfera* is a standalone environment for software measurement, from early estimates to exact measures; estimated sizes can be gradually translated into exact measures along with the requirements analysis; estimates and exact measures can also coexist in a single measurement exercise, for separate portions of the same system or software project. The *Sfera* tool offers advanced analysis and estimation functions, as effort estimation based on regression and

customizable models, productivity benchmarking and application portfolio management. *requestimate* is a plugin for the Rational / IBM *Requisite Pro* requirement management tool, that adds predefined and customizable capabilities to apply the E&Q classification over the requirements documentation, with no need of moving between two environments, in order to ease the measurement while managing the requirements, in a runtime process.

7. 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 [8], the Early & Quick estimation technique is a mixed-type method: it is based on both analogy and analysis. The former lets the estimator discover similarities between a new “piece” of a software application and similar “pieces” that were already classified for other applications. The latter guarantees stability for the estimate, since the software objects weights are based on statistical analysis and conceptually grounded. 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 [6]. 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. 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 been proved in practice to be quite effective, providing a response within $\pm 10\%$ 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 Base Functional Processes to Macro Functional Processes) with respect to corresponding standard measurement procedures.

Further improvements are guaranteed by the continuous validation process (for IFPUG case) and further refinements (for both IFPUG and COSMIC cases), while the overall E&Q technique can be easily extended to any further FSM method other than the examined cases.

13. References

- [1] Abran, A., Desharnais, JM., Oigny, S., St-Pierre, D., Symons, C., *COSMIC-FFP Measurement Manual (COSMIC Implementation Guide for ISO/IEC 19761: 2003) - Version 2.2*, Common Software Measurement International Consortium, January 2003. www.lrgl.uqam.ca/cosmic-ffp
- [2] Conte, M., Iorio, T., Meli, R., and Santillo, L., “E&Q: An Early & Quick Approach to Functional Size Measurement Methods”, *SMEF '04 Conference proceedings*, Rome, 28-20 January, 2004.
- [3] IFPUG, *Function Point Counting Practices Manual, Release 4.2*, International Function Point Users Group, www.ifpug.org, January 2004.
- [4] ISBSG, “The Benchmark Release 5”, International Software Benchmarking Standards Group, www.isbsg.org, 1998.
- [5] ISBSG, “The ISBSG Estimating, Benchmarking & Research Suite Release 9”, International Software Benchmarking Standards Group, www.isbsg.org, 2004.
- [6] ISO/IEC, “14143-1:1998 ‘Information technology – Software measurement – Functional size measurement – Part 1: Definition of Concepts’”, JTC 1 / SC 7, ISO/IEC, 1998.
- [7] R. Meli, “Early and Extended Function Point: a New Method for Function Point Estimation”, IFPUG '97 Conference, Scottsdale, 15-19 September, 1997.
- [8] R. Meli, and L. Santillo, “Function Point Estimation Methods: a Comparative Overview”, *FESMA '99 Conference proceedings*, Amsterdam, 4-8 October, 1999.
- [9] R. Meli, A. Abran, V.T. Ho, and S. Oigny, “On the applicability of COSMIC-FFP for measuring software throughout its life cycle”, *ESCOM '00 Conference proceedings*, Munich, 18-20 April, 2000.
- [10] R. Meli, “Early and Quick Function Point Analysis from Summary User Requirements to Project Management”, in: IFPUG, *IT Measurement – Practical Advice from the Experts*, pp. 417-440, Addison-Wesley, 2002.
- [11] F. Voegelzang, “Early estimating using COSMIC-FFP”, *SMEF '05 Conference proceedings*, Rome, 16-18 March, 2005.