Early & Quick Function Point Method

An empirical validation experiment

Roberto Meli
Data Processing Organization Srl
Rome, Italy
email: roberto.meli@dpo.it

Abstract— **The "Early & Quick Function Points Approach (E&QFPA)" is a mean to approximate the results of some standard Functional Size Measurement Methods like IFPUG, SiFP or COSMIC. E&QFPA is a set of concepts and procedures that, even when applied to non-detailed functional specifications of a software system, maintains the overall structure and the essential principles of standard functional size measurement methods. The E&QFPA combines different estimation approaches in order to provide better approximations of a software system functional size: it makes use of both analogical and analytical classification of function types (transactions and data). Moreover, it allows the use of different levels of detail for different branches of the system (multilevel approach). This paper illustrates the basic concepts of the method which is mature and well established in the Italian market, as well as the results of an empirical validation experiment conducted on a real business data set of IFPUG function point measures. The usage of such a method may contribute to the rapid quantification of user requirements very early in the production life cycle.**

Keywords: function; point; estimation; approximation.

I. INTRODUCTION

A Functional Software Measurement Method (FSMM) is a mean to measure Functional User Requirements (FUR) of a software application according to the rules of an international ISO/IEC standard [1]. The measurement process required by the most diffused FSMM is often perceived by the ICT personnel as excessively time consuming, expensive and difficult to apply in business contexts where the details needed for FSMM standard application are not always available or stable enough. Therefore, several simplified approximation processes have been proposed [3]-[7]. The Early & Quick Function Points Approach is one of these.

The paper is structured as follows: Section II summarizes the E&QFP method; Section III reports on the advantages in using the method in business practices; Section IV presents the criteria that were used to assess the accuracy of the method at various levels of application; Section V presents the conditions of the experiment and its results; Section VI contains a "qualitative" comparison of several types of approximation methods; Section VII states the conclusions.

II. BASIC CONCEPTS

This Section introduces the origin and the basic concepts of the E&QFPM needed to understand the framework used for the empirical experiment. The contents presented here are not sufficient to allow an in dept comprehension of the method in itself. The reader interested in mastering the method should refer to the standard documentation [19].

The Early & Quick FP method was created in 1997 by the author in order to facilitate the approximation (also called estimation) of the IFPUG Function Points values [8][1]. It was presented for the first time at the ESCOM 97 conference [10] and later at the IFPUG conference [11]. Since then, the original approach has evolved and its usage is increased [14]-[18]. The method has been reported in 2009 as the best choice in approximating methods by the CNIPA Italian Government Authority [12]. In 2000, the approach was extended, experimentally, to the COSMIC Functional Size Method [12]. In 2006, the Early & Quick Function Points Method - E&QFPM (IFPUG version) - has become a registered trademark but the method is available in the public domain since it is managed as a "Publicly Available Method", subject to the Creative Commons license, attribution-non derivative works. The E&QFP development team has opened the doors to external contributions and the technique evolves considering feedbacks from actual users in the market. DPO continues to support, improve and customize the method publishing a new version any time it is needed by the technical community. The method is not a commercial product. A certification program has been created to guarantee that the method is used consistently among different practitioners. After 15 years from the initial formulation, the latest evolution of the method, identified as version 3.1 [19] was released in April 2012 integrating the new Simple Function Point FSMM [20]-[22].

The Early & Quick Function Points Approach (E&QFPA) is a set of concepts and procedures that, even when applied to non-detailed functional specifications of a software system, maintains the overall structure and satisfies the essential principles of standard functional size measurement methods. It may be applied to approximate different types of Functional Size Measurement Methods (FSMM).

The E&QFPA combines different estimating techniques in order to provide better approximations of a software system functional size: it makes use of both analogical and analytical classification of function types (logical transactions and data). Moreover, it allows 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 of values, i.e., a set of minimum, most likely, and maximum values) is the weighted sum of the individual components' uncertainty levels. The "core driver" of the approach is an analytically and statistically originated table of FP (Function Points) values to be used in making functional size estimation.

The E&QFPA is based on the fundamental principles reported in TABLE I.

TABLE I - E&QFP FUNDAMENTAL PRINCIPLES

Principle	**Explanation**
Classification by analogy	Similarity in the overall functionality between new and existing known software objects.
Structured aggregation	Grouping of a certain number of lower level software logical objects in one higher level logical object.
Estimation flexibility	Data and transactional components are assessed autonomously. No predefined and fixed function/data ratio is assumed.
Multilevel approach	No discard of existent details, if available – no need of "invented" 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.

From now on we will restrict our interest in the IFPUG variant of the approach that we will simply call Early & Quick Function Points Method - E&QFPM.

The values in the derivation table for the IFPUG approximation model were originally stated by expert judgment and later on by the ISBSG data set analysis [23]. Once the values are determined for any specific version of the method they are not changed anymore in order to use the method in a consistent way across practitioners, environments, organizations etc. Local calibration of the E&QFPM is always possible in order to better the results for a specific context but it should be clearly reported by the practitioners as a variation of the standard version.

Figure 1 shows the estimation process starting with the Functional User Requirements interpretation and ending with the FP estimation.

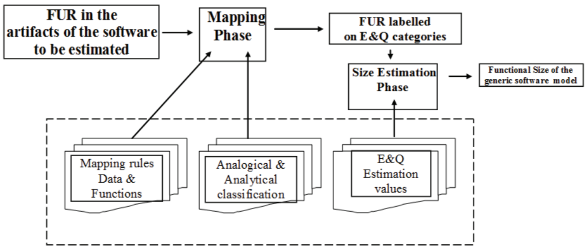


Figure 1. Early & Quick FP Estimation Process

The starting point of the process is the logical product breakdown structure of the system being estimated, and the mapping of FURs on the E&QFP elements. The basic E&QFP elementary components are the following software objects:

- logical data groups, and
- elementary functional processes,

that is, the same Base Functional Components (BFC) types of the IFPUG measurement method. Further aggregations, as depicted in Figure 2, are provided:

- data BFC can be grouped into general data groups;
- transactional BFC can be grouped into "general" logical processes;
- general processes can be grouped into "macro" logical processes.

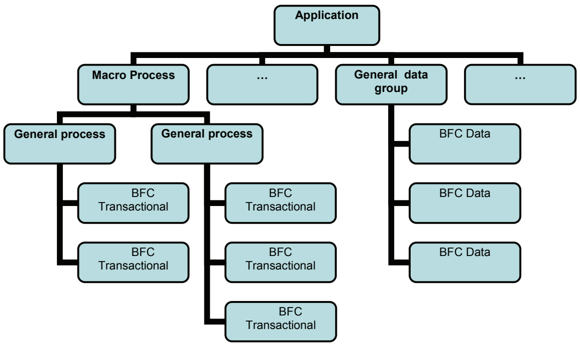


Figure 2. Functional hierarchy in the E&Q estimation method (for sake of simplicity, only one instance of macro process and one instance of general data group are shown)

Each "software logical object" is assigned a set of FP values (minimum, most likely, maximum) based on an analytical/statistical table, 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 being considered, a "structured" 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 easier to assign the right level of classification to each element on the list, and therefore to derive its contribution to the overall size.

Obviously, if any particular object in the functional tree is estimated assigning directly an FP value then no *contained object* (i.e., in a “father”-“son” relationship) must be considered to contribute directly to the grand total: as a matter of fact, all the “son’s” values are already included into the “father’s” values and must not be added to it. The estimation uncertainty (represented by the minimum-maximum range) is dependent on the level of object in the hierarchy and will be greater for higher levels of software objects aggregation, due to the higher lack of details.

Estimation using E&QFP technique may be done at three levels of detail depending on the granularity of the components used for estimation: summary, intermediate and detailed.

III. ADVANTAGES IN USING E&QFPM

E&QFPM is an approximation method which is:

- Fast
- Cheap
- Adaptable
- Reliable
- Documentable
- Easy to learn

Fast: the empirical study described in Section V showed a productivity from 1.9 to 6.3 times better than a standard measurement. This result is aligned with informal experience derived by daily use of the method by practitioners in several Italian organizations. The reason is that much less elements should be identified and evaluated in the User Requirements documentation than for the IFPUG measurement.

Cheap: because it is fast and does not involve more specialized people than a standard measurement.

Adaptable: because it is applicable either when the standard technique is not a possible choice - due to missing detailed information –either when the details are available.

Reliable: when we consider the “technical accuracy” (see later).

Documentable: since the approximation is based on elements extracted from the FUR that may be described, discussed, shared and tracked by the practitioners. Direct estimation techniques are only based on personal intuition.

Easy to learn: because the rules are much simpler than the ones in the standard reference measurement manual. This does not mean that the method is always easy to apply, actually, since when the FUR are not enough detailed to identify BFC (Base Functional Component – the smallest measurable part of FUR – see [2]), a strong experience is needed to practice the analogy, which is essential for a good estimation.

Of course, any estimation method (and E&QFPM is not different) has a unavoidable uncertainty in its usage that makes it incomparable with a measurement method in terms of accuracy but we must accept that if we use an estimation method it is because we are not able (due to missing information) or do not want (due to missing measurement resources) to use a measurement method and the cost of doing that is a higher potential error in sizing.

IV. EVALUATION RULES AND CRITERIA USED TO ASSESS THE RELIABILITY OF THE E&QFP METHOD

In this Section, we will clarify the criteria used to represent the outcomes of the E&QFP method when applied to a real life sample of software applications, which was not used to calibrate the method in itself.

First of all, we have to work out the meaning of the quality attribute named “accuracy” of the method. By “accuracy” we intend the absence of systematic and random errors. According to [1] it is the “closeness of agreement between a measured quantity value and a true quantity value of a measurand”. In our case the “*true quantity value” is the IFPUG FP value* for a software application and *the E&Q value for the same measurand is the “measured quantity value”* (measured with approximation, of course) that we want to analyse. In addition, we use the term “measurement error” or simply “error” to intend the “measured quantity value minus a reference quantity value” whereas the “relative error” is the “error divided by the reference quantity value”. Again, in our case, the *measured quantity value is the E&QFP value* determined for a software application and the *reference quantity value is the IFPUG FP value* for the same application.

Estimation is a human intensive process which involves personal capabilities in addition to technical tools and rules. In evaluating the accuracy of an approximation method (its capability to predict exact values) we should be able to separate the judgement on the human capabilities from the judgement of the method itself. This is usually not easy to do since methods are used by human beings: their knowledge of the software requirements, of the application domain and of the method itself are directly related to the final accuracy of results, in real specific situations. Nevertheless, for FP approximation, we believe it is possible to assess separately the **“technical accuracy”** and the **“operational accuracy”**.

As we have seen, the E&QFP method is based, mainly, on the classification of logical requirements with respect to a standard or customized assignment table provided by the method itself. If we apply the estimation method on an already built application we have the capability to construct the exact hierarchy of logical objects to be used in the estimation at various levels of detail. Furthermore, since we have all the detailed information, we are able to assign each aggregated object exactly in the right class of the table. For example, if we aggregate some detailed requirements into a general requirement and we identify it as a General Process we are able to count exactly how many BFCs are grouped into the GP and we may exactly classify it as a small, medium or large GP, as required by the E&QFPM.

In this way, we may not be wrong with classifications and the final accuracy will be purified by the human classification errors. The residual error is given by the difference between the actual measurement of the BFCs included in the GP and the weight that the method assigns to it: this is what we call “technical accuracy”. In real life, estimators will not experience this simplified situation since the requirements will not be generally known at the most detailed level and in addition to the eventual “technical” error it is possible to have a classification error which leads to an “operational” error which we expect to be greater than the “technical” one, unless underestimations compensate overestimation in the overall exercise. In addition to this aspect, it is also essential to understand that the method may be used at three different levels of detail each of one characterized by a different technical accuracy, so that it is impossible to assess the accuracy of the method, generally speaking, but it is possible to assess the accuracy of the method *at a specific level of application*.

Ignoring these essential aspects may lead to inconsistent results and false conclusions on the reliability of the method in itself as reported in [24], where the capability of freshly and quickly trained practitioners was measured together with the “technical accuracy” and estimation levels of detail were mixed up comparing incomparable results and deriving an overall accuracy for the method, which simply doesn’t exist. If we want to draw conclusions on the accuracy of the method and not on the capability of practitioners then we have to depurate the experiment from the participant’s bias using the approach outlined here. In real life contexts, the quality of the operational estimation will depend on the expertise of the estimator, on the available FUR details, on the level of application of the method. If we do not fix all these parameters in an experimental situation it is impossible to conduct adequate empirical observations.

Before introducing the indicators that we used to assess the “technical accuracy” we want to state that in this paper we assume that given a specific software application the exact value of FP is the value measured using the standard IFPUG rules by a certified FP Specialist (CFPS) and the estimated values to be compared with the measured value are those expressed by a certified E&QFP specialist for the same application. The error is the difference between the approximated value and the standard measured value, the relative error is the difference between the approximated value and the measured value all divided by the measured value.

A. Portfolio error

This is the error derived for the entire sample conceived as if it was a portfolio of applications, in other words a set of software applications managed as a whole for business reasons. In real cases, it is important to know what is the general behaviour of the portfolio in addition to the behaviour of single components. Economical resources are distributed over individual software applications but it is important to know if underestimations of portfolio components might be compensated by overestimations of other components or if the errors are of the same nature and sum up to unacceptable levels. “Portfolio error” is calculated adding up all the measurements and estimations and calculating the difference between them as if the set of applications was actually only one bigger application (its measure is the sum of the individual measurements and its approximation is the sum of the individual approximations). This indicator is only a “business” indicator, useful when associated to the other indicators like the following (more traditional) ones.

B. Prediction at level X

This indicator – pred (X) - is simply the relative number of estimations (%) that fall inside the range of “actual value” +/- X%. Usually X is set at 25% for model’s evaluation. Since the E&QFP method is a “good performer” with respect to the technical accuracy, we decided to lower this threshold to 10%.

C. Mean Relative Error / Mean Absolute Error

The mean relative error is the average of the relative errors (with their sign). The mean absolute error is the average of the absolute errors.

D. Median Relative Error / Median Absolute Error

The median is the value that is roughly in the middle of the data set. If n is odd, the median is the single value in the middle, namely the value with rank (n + 1)/2. If n is even, there is not a single value in the middle, so the median is defined to be the average of the two middle values, namely the values with ranks n/2 and n/2 + 1. The median value is less sensible to the influence of extreme values with respect to the arithmetic mean. The median could be calculated over the relative values or over the absolute values.

E. Reliability Indicator

The reliability indicator *RI* provides a numerical evaluation of the accuracy of the estimation with respect to the corresponding measurement method. This indicator does not express the variability range, but rather the (a posteriori) deviation between the actual measured size value *M* and the estimation range (S_{min}, S_{ml} and S_{max}) – where *ml* means *most likely*. 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:

$$RI_i = \frac{(S_{\max} - S_{\min}) - |M - S_{ml}|}{(S_{\max} - S_{\min})} \qquad (1)$$

The indicator has the following features:

RI_i has a threshold value for *M* equal to one of the range extremis (the smallest value of the two);

RI_i gets worse for M going externally of the estimation range, and vice versa;

RI_i gets better for smaller ranges (S_{max} - S_{min});

RI_i gets better for smaller differences between M and S_{ml} and yields the best value for $M = S_{ml}$.

Figure 3 shows an example of the shape of RI_i *(y-values)*, with fixed values $S_{min} = 10$ FP, $S_{ml} = 13$ FP, $S_{max} = 20$ FP and actual measured value M varying on that range (x-values in Function Points). In the best case ($M = S_{ml}$), we find $RI_i = 1$ (maximum); in the extremis ($M = S_{min}$ o $M = S_{max}$), we find the threshold $RI_i = 0.3$ (=min(0,3;0,7)); for bad estimations (M external to the range), $RI_i < 0.3$. Hence, in this case *the expected value of the reliability indicator, for a satisfactory estimation technique, is between 0.3 and 1*. The closest to 1 is the RI and the better is the estimation.

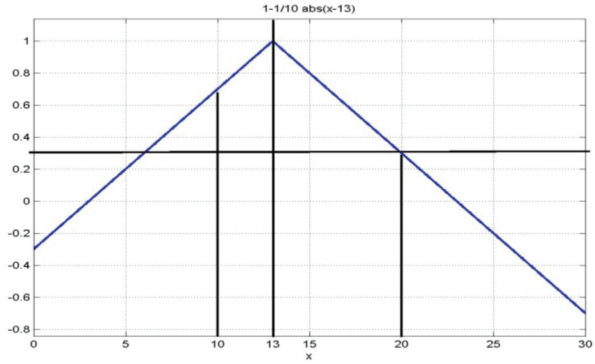


Figure 3. Shape of RI_i with respect to the actual measured size M

The overall reliability indicator of the estimation method is given by the average *RI* over *N* cases. Thus, the average reliability indicator provides, for future estimations, an evaluation of the associated “risk”.

V. A VALIDATION EMPIRICAL EXPERIMENT

A validation empirical experiment has been conducted in order to assess the accuracy of the estimation results, using a real life data set of 65 IFPUG FP measurements ranging from 113 FP to 1601 FP (51 baselines; 7 new developments; 8 enhancements). The measurements were taken by Certified Function Point Specialists of the organization originating the data set.

This data set was not used to calibrate the version of the model, it was used to test the “technical accuracy” of the standard model on an independent data set. Details on the data set are reported in the Supplemental Material Section.

A. Measurement and Approximation productivity

TABLE II shows the IFPUG average measurement productivity registered for the 65 cases compared to the approximation productivity of the three level of application of the E&QFP method.

TABLE II - SIZING PRODUCTIVITY

	IFPUG	Detailed	Intermediate	Summary
avg(hours/FP)	45.2	85.7	177.0	284.2
ratio	1.0	1.9	3.9	6.3

B. Universe description

The following TABLE III shows the universe description.

TABLE III – UNIVERSE DESCRIPTION

Universe description	
Average	513.6
Median	403.0
Moda	170.0
Kurtosis	0.88179
Asymmetry	1.18710
Interval	1488
Minimum	113
Maximum	1601
Sum	33384
Count	65

C. Portfolio error

The portfolio error is extremely low as it is shown by the following TABLE IV. This means that using the estimation technique at any level, the total portfolio is estimated in an extremely precise way.

TABLE IV – PORTFOLIO ERRORS

	IFPUG value	Estimation	Difference	%	Abs(%)
Detailed	33384	32466,5	-917,5	-3%	3%
Intermediate	33384	33821,7	437,7	1%	1%
Summary	33384	33732,2	348,2	1%	1%

D. Prediction at level X

The most part of the estimations are beneath the 10% of absolute error as shown in TABLE V. We used an improved version of the typical Pred(25%) due to the high quality of estimations available.

TABLE V – PREDICTION AT LEVEL 10%

	total	<=10%	Pred(10%)
Detailed	65	51	78%
Intermediate	65	55	85%
Summary	65	46	71%

E. *Error magnitudes*

In TABLE VI, we show the minimum, median, average, maximum errors with their sign and as absolute values for the three level of details.

TABLE VI – ERROR MAGNITUDES

	Detailed		Intermediate		Summary	
	error%	abs(error%)	error%	abs(error%)	error%	abs(error%)
min	-18%	0%	-17%	0%	-17%	0%
median	-2%	4%	-2%	5%	-1%	6%
avg	-3%	6%	-1%	6%	-1%	7%
max	20%	20%	27%	27%	24%	24%

F. *Average Reliability Indicator*

The average RI indicator (shown in TABLE VII) is very high in the case of intermediate and summary estimation and low in the case of detailed estimation, due to the very tight range of confidence interval for the detailed estimation. In 15 cases out of 65, the RI was negative since the measured FP value was outside the min-max estimation range, but in none of those cases the absolute error was higher than 20%.

TABLE VII - AVERAGE RELIABILITY INDICATOR

	avg(RI)
Detailed	0,33
Intermediate	0,81
Summary	0,86

G. *Scatterplot diagrams*

Figure 4, 5 and 6 just report, for a visual check, the IFPUG values plotted against the approximated values for the same cases using the three different levels of application of the method.

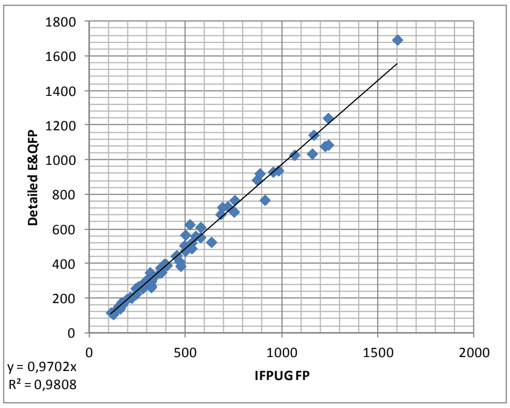


Figure 4. IFPUG FP vs Detailed E&QFP

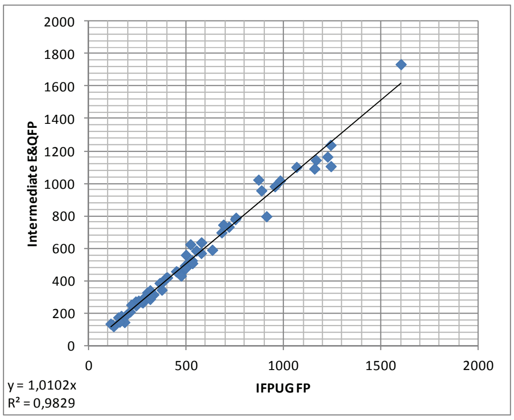


Figure 5. IFPUG FP vs Intermediate E&QFP

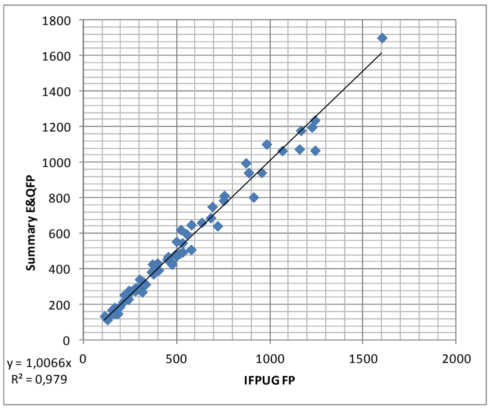


Figure 6. IFPUG FP vs Summary E&QFP

VI. COMPARISON WITH OTHER APPROXIMATION METHODS

In [5], there is a classification of the approximation methods useful to compare E&QFPM to other available methods.

These methods, also known as Algorithmic Model Methods, provide one or more transformation algorithms, which produce a software size estimate as a function of a number of variables, which in turn relate to software attributes. Generally, these methods are correlated to a decomposition process. By decomposing an application into its major functions, estimation can be performed in a step-by-step fashion. This category is further specialized into three main subclasses: Technology Driven Methods, Logic Driven Methods (also called Architecture Driven Methods) and Hybrid Methods.

Technology Driven Methods - This term denotes the derivation of the FP value, for a given software system, from its technical elements, for example Lines of Code or from the number of classes or objects (in an OO environment), physical tables, screen forms, widgets and the like. Using this kind of method, it is not necessary to develop a logical model of the application in terms of functionalities or data entities since the derivation algorithm is a based on statistically derived ratios, i.e., it is based on a statistically proven (hopefully) correlation between a technical measure and a logical measure (the FP count). This kind of method is not reliable due to the significant differences between a physical model and a logical model of software and to the modern programming technologies. An additional problem is that the ratio between technical and logical measures might be easily manipulated in order to "drive" the final FP estimation in a pre-defined direction.

Logic Driven Methods - There are many Logic Driven Methods for estimating FP size, mostly because of statistical research on benchmarking data sets. Not every method is quoted in technical literature, but many of them are widely known in practice. The main characteristic of these methods is that they are based on a "logical model" of the application to be estimated. This means that the model is totally compliant with the Functional Size Model of the IFPUG method. One subset of Logic Driven Methods may be called **Extrapolative Counts**: this kind of methods assumes that we count only one or two FP components (typically the number of Logical Files) of the application, and derive the rest of the count on a statistical or theoretical basis. All of these models should be carefully analyzed, in order to understand their applicability in a particular domain. Very often this method may be customized to reflect the FP distribution of a particular environment instead of a global public available database. This method is simple to use but quite error prone, since a "missing" object may involve a lot of derived FP to be disregarded and the vice versa. As a final consideration, this method needs a very accurate identification of the estimation driver (i.e., ILFs), which is not usually possible at a moment in time when the requirements are still uncertain, vague, approximate and instable. A second type of Logic Driven Methods may be called **Sampled Counts:** using this method, the IFPUG standard count of a part of the entire system is carried out and from this partial count, the rest of the system can be estimated. While in the previous situation the whole system is investigated with respect to some FP components (EI, EO, EQ, ILF or EIF), using the Sampled Count method only a portion of the system is investigated with respect to all the FP components. This method is simple as the previous one but it could be even more error prone since the assumptions about the proportion of the known part over the rest of the system are not really reliable. In addition, there is still the problem of obtaining very detailed information on the "known" part to be counted using the standard procedures and rules in situations where these data might be unavailable. A third type of Logic Driven Methods may be called **Average Complexity Estimation:** this type of method consists in identifying all the IFPUG Base Functional Components – BFC (EI, EO, EQ, ILF, EIF) and assuming an average or most likely complexity for each one of them. This is often a quite precise method but it needs a detailed insight in the software logical requirements as if it was a standard count.

Hybrid Methods – These methods merge technical driven aspects with logical modelling and might accelerate the estimation process but further research is needed to demonstrate their value.

The **E&QFPM** is the most flexible method, since it is based on a mix of approaches including Sampled Counts, Average Complexity Estimation but introduces analogy and multilevel approach. Any of the quoted methods may be considered as specific cases of use of the E&QFPM.

VII. CONCLUSIONS

The analysis of the empirical experiment has confirmed and improved results shown by other studies and current practice. The Early & Quick FP estimation technique is a competitive mean to approximate IFPUG FP values in such a precise way that in many organizations it is used as a primary way to evaluate assets and projects. The effect of expert analogy and domain experience (a potential source of errors in the field) becomes less important as much as the technique is used at the intermediate and detailed level. In these cases, the technical accuracy becomes very close to the operational accuracy. It is important to highlight that the eventuality of committing a large error, in using the E&Q method at a low level of detail of requirements, is largely compensated by the fact that no other approximation technique may be used on summary requirements which are missing the needed details.

ACKNOWLEDGMENT

The experimental data set (IFPUG measures) has been supplied by an Italian public agency and the estimations were made by Luciano Luciani (DPO) IFPUG Certified Function Point Specialist (CFPS) and Certified Early & Quick Function Point Specialist (CEQ-FPS).

REFERENCES

[1] JCGM, JCGM 200:2008 - International vocabulary of metrology — Basic and general concepts and associated terms (VIM), http://www.iso.org/sites/JCGM/VIM/JCGM_200e.html, [retrieved: March, 2015]

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

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

[4] H. van Heeringen, E. van Gorp, and T. Prins, "Functional size measurement - Accuracy versus costs - Is it really worth it?", SMEF 2009, Rome, Italy, May 2009.

[5] R. Meli, "Functional size approximation: why bother with details ?", IWSM Metrikon MENSURA, Amsterdam, The Netherlands, 2009.

[6] C. Jones, "A new business model for Function Point metrics", August 2009.

[7] L. Lavazza and G. Liu, "An Empirical Evaluation of Simplified Function Point Measurement Processes", Int.

Journal On Advances in Software, vol. 6, n.1/2, pp. 1–13, 2013.

[9] ISO/IEC 20926: 2003, “Software engineering – IFPUG 4.1 Unadjusted functional size measurement method – Counting Practices Manual”, Geneva, 2003.

[10] R. Meli, "Early Function Points : a new estimation method for software projects" - ESCOM 97 - May 1997 - Berlin (Germany).

[11] R. Meli, "Early and Extended Function Point: a new method for Function Points estimation" - IFPUG - Fall Conference - 15-19 September 1997 - Scottsdale, Arizona USA.

[12] CNIPA, Linee guida sulla qualità dei beni e dei servizi ICT per la definizione ed il governo dei contratti della Pubblica Amministrazione - Strategie di Acquisizione delle Forniture ICT v3.4, pp. 205/249, 2005.

[13] R. Meli, A. Abran, V. T. Ho, and S. Oligny, “On the applicability of COSMIC-FFP for measuring software throughout its life cycle”, ESCOM-SCOPE 2000, Munich, April 18-20, 2000.

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

[15] M. Conte, T. Iorio, R. Meli, and L. Santillo, "E&Q: An Early & Quick Approach to Functional Size Measurement Methods", SMEF2004, Roma, Italia, January 2004.

[16] L. Santillo, M. Conte, and R. Meli, "Early & Quick Function Point: Sizing More with Less" , 11th IEEE International Software Metrics Symposium, 19-22 September, 2005 Como, Italy.

[8] International Function Point Users Group, “Function Point Counting Practices Manual - Release 4.3.1”, January 2010.

[17] R. Ellafi and R. Meli, "A Source Code Analysis Function Point Estimation Method Integrated with a Logic Driven Estimation Method", SMEF2006, Roma, Italy, May 2006.

[18] T. Iorio, R. Meli, and F. Perna, "Early & Quick Function Points® v3.0: enhancements for a Publicly Available Method", SMEF2007, Italy, May 2007.

[19] DPO, Early & Quick FP Reference Manual, http://www.dpo.it/en/eqfp/risorse.htm, [retrieved: March, 2015].

[20] R. Meli, "Simple Function Point: a new Functional Size Measurement Method fully compliant with IFPUG 4.x", SMEF2011, Roma.

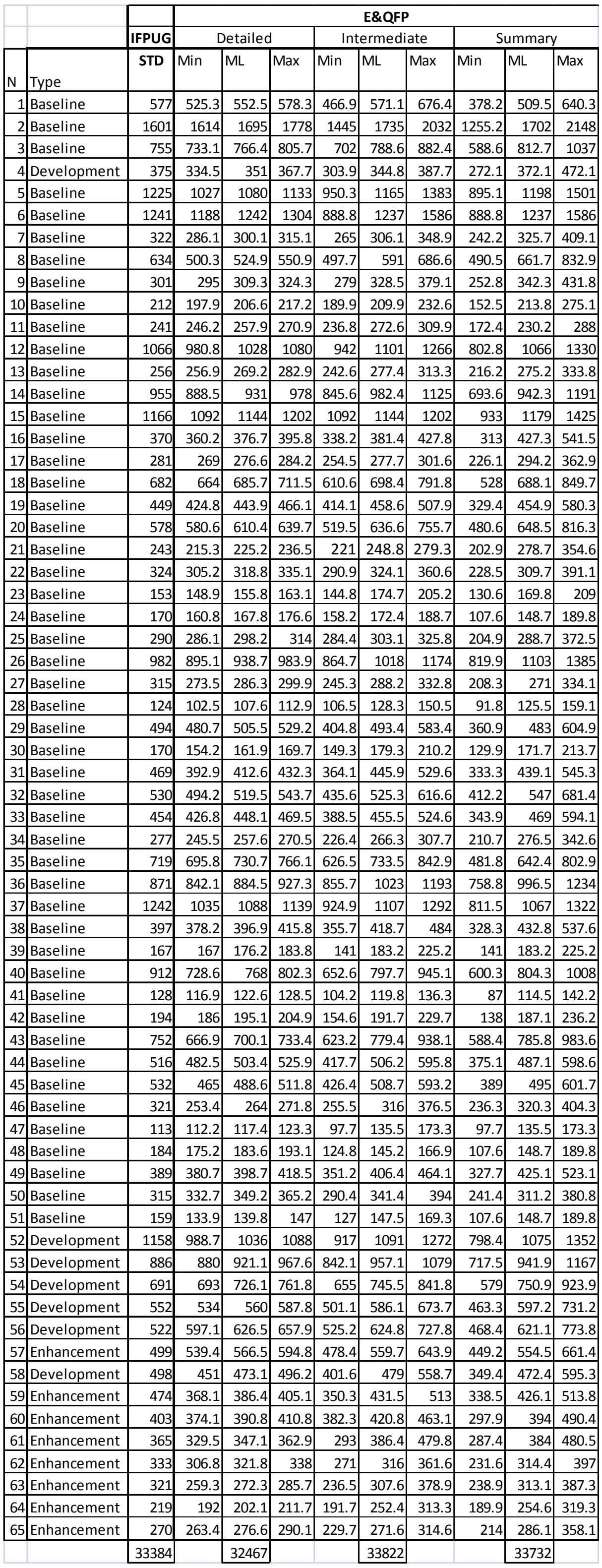
[21] R. Meli, "Simple Function Point! A new method for functional size measurement fully compatible with the IFPUG method 4.x", UK Software Metrics Association & COSMIC International Conference on Software Metrics and Estimating, London , October 2011, London.

[22] SiFPA, "Simple Function Point Functional Size Measurement Method - Reference Manual SiFP-01.00-RM-EN-01.01", http://www.sifpa.org/en/index.htm, [retrieved: March, 2015].

[23] International Software Benchmarking Standards Group, Worldwide Software Development: The Benchmark, (from rel 5, 1999 to rel 12, 2013).

[24] K. Almakadmeh and A. Abran, “Experimental Evaluation of an Industrial Technique for the Approximation of Software Functional Size”, International Journal Of Computers & Technology Vol 10, No 3, 2013.

SUPPLEMENTAL MATERIAL : THE EXPERIMENTAL DATA SET



			E&QFP								
		IFPUG	Detailed			Intermediate			Summary		
N	Type	**STD**	Min	ML	Max	Min	ML	Max	Min	ML	Max
1	Baseline	577	525.3	552.5	578.3	466.9	571.1	676.4	378.2	509.5	640.3
2	Baseline	1601	1614	1695	1778	1445	1735	2032	1255.2	1702	2148
3	Baseline	755	733.1	766.4	805.7	702	788.6	882.4	588.6	812.7	1037
4	Development	375	334.5	351	367.7	303.9	344.8	387.7	272.1	372.1	472.1
5	Baseline	1225	1027	1080	1133	950.3	1165	1383	895.1	1198	1501
6	Baseline	1241	1188	1242	1304	888.8	1237	1586	888.8	1237	1586
7	Baseline	322	286.1	300.1	315.1	265	306.1	348.9	242.2	325.7	409.1
8	Baseline	634	500.3	524.9	550.9	497.7	591	686.6	490.5	661.7	832.9
9	Baseline	301	295	309.3	324.3	279	328.5	379.1	252.8	342.3	431.8
10	Baseline	212	197.9	206.6	217.2	189.9	209.9	232.6	152.5	213.8	275.1
11	Baseline	241	246.2	257.9	270.9	236.8	272.6	309.9	172.4	230.2	288
12	Baseline	1066	980.8	1028	1080	942	1101	1266	802.8	1066	1330
13	Baseline	256	256.9	269.2	282.9	242.6	277.4	313.3	216.2	275.2	333.8
14	Baseline	955	888.5	931	978	845.6	982.4	1125	693.6	942.3	1191
15	Baseline	1166	1092	1144	1202	1092	1144	1202	933	1179	1425
16	Baseline	370	360.2	376.7	395.8	338.2	381.4	427.8	313	427.3	541.5
17	Baseline	281	269	276.6	284.2	254.5	277.7	301.6	226.1	294.2	362.9
18	Baseline	682	664	685.7	711.5	610.6	698.4	791.8	528	688.1	849.7
19	Baseline	449	424.8	443.9	466.1	414.1	458.6	507.9	329.4	454.9	580.3
20	Baseline	578	580.6	610.4	639.7	519.5	636.6	755.7	480.6	648.5	816.3
21	Baseline	243	215.3	225.2	236.5	221	248.8	279.3	202.9	278.7	354.6
22	Baseline	324	305.2	318.8	335.1	290.9	324.1	360.6	228.5	309.7	391.1
23	Baseline	153	148.9	155.8	163.1	144.8	174.7	205.2	130.6	169.8	209
24	Baseline	170	160.8	167.8	176.6	158.2	172.4	188.7	107.6	148.7	189.8
25	Baseline	290	286.1	298.2	314	284.4	303.1	325.8	204.9	288.7	372.5
26	Baseline	982	895.1	938.7	983.9	864.7	1018	1174	819.9	1103	1385
27	Baseline	315	273.5	286.3	299.9	245.3	288.2	332.8	208.3	271	334.1
28	Baseline	124	102.5	107.6	112.9	106.5	128.3	150.5	91.8	125.5	159.1
29	Baseline	494	480.7	505.5	529.2	404.8	493.4	583.4	360.9	483	604.9
30	Baseline	170	154.2	161.9	169.7	149.3	179.3	210.2	129.9	171.7	213.7
31	Baseline	469	392.9	412.6	432.3	364.1	445.9	529.6	333.3	439.1	545.3
32	Baseline	530	494.2	519.5	543.7	435.6	525.3	616.6	412.2	547	681.4
33	Baseline	454	426.8	448.1	469.5	388.5	455.5	524.6	343.9	469	594.1
34	Baseline	277	245.5	257.6	270.5	226.4	266.3	307.7	210.7	276.5	342.6
35	Baseline	719	695.8	730.7	766.1	626.5	733.5	842.9	481.8	642.4	802.9
36	Baseline	871	842.1	884.5	927.3	855.7	1023	1193	758.8	996.5	1234
37	Baseline	1242	1035	1088	1139	924.9	1107	1292	811.5	1067	1322
38	Baseline	397	378.2	396.9	415.8	355.7	418.7	484	328.3	432.8	537.6
39	Baseline	167	167	176.2	183.8	141	183.2	225.2	141	183.2	225.2
40	Baseline	912	728.6	768	802.3	652.6	797.7	945.1	600.3	804.3	1008
41	Baseline	128	116.9	122.6	128.5	104.2	119.8	136.3	87	114.5	142.2
42	Baseline	194	186	195.1	204.9	154.6	191.7	229.7	138	187.1	236.2
43	Baseline	752	666.9	700.1	733.4	623.2	779.4	938.1	588.4	785.8	983.6
44	Baseline	516	482.5	503.4	525.9	417.7	506.2	595.8	375.1	487.1	598.6
45	Baseline	532	465	488.6	511.8	426.4	508.7	593.2	389	495	601.7
46	Baseline	321	253.4	264	271.8	255.5	316	376.5	236.3	320.3	404.3
47	Baseline	113	112.2	117.4	123.3	97.7	135.5	173.3	97.7	135.5	173.3
48	Baseline	184	175.2	183.6	193.1	124.8	145.2	166.9	107.6	148.7	189.8
49	Baseline	389	380.7	398.7	418.5	351.2	406.4	464.1	327.7	425.1	523.1
50	Baseline	315	332.7	349.2	365.2	290.4	341.4	394	241.4	311.2	380.8
51	Baseline	159	133.9	139.8	147	127	147.5	169.3	107.6	148.7	189.8
52	Development	1158	988.7	1036	1088	917	1091	1272	798.4	1075	1352
53	Development	886	880	921.1	967.6	842.1	957.1	1079	717.5	941.9	1167
54	Development	691	693	726.1	761.8	655	745.5	841.8	579	750.9	923.9
55	Development	552	534	560	587.8	501.1	586.1	673.7	463.3	597.2	731.2
56	Development	522	597.1	626.5	657.9	525.2	624.8	727.8	468.4	621.1	773.8
57	Enhancement	499	539.4	566.5	594.8	478.4	559.7	643.9	449.2	554.5	661.4
58	Development	498	451	473.1	496.2	401.6	479	558.7	349.4	472.4	595.3
59	Enhancement	474	368.1	386.4	405.1	350.3	431.5	513	338.5	426.1	513.8
60	Enhancement	403	374.1	390.8	410.8	382.3	420.8	463.1	297.9	394	490.4
61	Enhancement	365	329.5	347.1	362.9	293	386.4	479.8	287.4	384	480.5
62	Enhancement	333	306.8	321.8	338	271	316	361.6	231.6	314.4	397
63	Enhancement	321	259.3	272.3	285.7	236.5	307.6	378.9	238.9	313.1	387.3
64	Enhancement	219	192	202.1	211.7	191.7	252.4	313.3	189.9	254.6	319.3
65	Enhancement	270	263.4	276.6	290.1	229.7	271.6	314.6	214	286.1	358.1
		33384		32467			33822			33732	