An Empirical Evaluation of the Accuracy of NESMA Function Points Estimates

Luigi Lavazza

Dipartimento di Scienze teoriche e Applicate
Università degli Studi dell'Insubria
21100 Varese, Italy
Email: luigi.lavazza@uninsubria.it

Geng Liu

School of Computer Science and Technology
Hangzhou Dianzi University
Hangzhou, China
Email: liugeng@hdu.edu.cn

Abstract—**Functional size measures of software—especially Function Points—are widely used, because they provide an objective quantification of software size in the early stages of development, i.e., as soon as functional requirements have been analyzed and documented. Unfortunately, in some conditions, performing the standard Function Point Analysis process may be too long and expensive. Moreover, functional measures could be needed before functional requirements have been elicited completely and at the required detail level. To solve this problem, many methods have been invented and are being used to estimate functional size based on incomplete or not fully detailed requirements. Using these methods involves a trade-off between ease and timeliness on one side and accuracy on the other side. In fact, estimates are always affected by some error; knowing the magnitude of estimation errors that characterize the estimates provided by a given method is of great importance to people who use size estimates. This paper reports the results of an empirical study devoted to evaluate the accuracy of estimates provides by 'NESMA estimated' and 'NESMA indicative' methods, which are among the best known and most widely used Function point estimation methods. The results of the study show that the NESMA estimated method provides estimates that are accurate enough for practical usage.**

***Keywords*–Function Points; IFPUG; Function point Analysis; Functional Size Measurement; Functional Size Estimation; NESMA estimated; NESMA indicative; Early Size Estimation.**

I. INTRODUCTION

The availability of accurate functional size measures can help software companies plan, monitor and estimate development costs, and control software development processes. Among the functional size measurement methods that have been proposed, Function Point Analysis (FPA) [1] is by far the most popular. The International Function Points User Group (IFPUG) took charge of maintaining FPA and publishes the official Function Point counting manual [2].

In some conditions, performing the standard FPA process may be too long and expensive. Moreover, standard FPA can be applied only after the completion of the software requirements elicitation stage, while functional measures could be needed earlier, i.e, before functional requirements have been elicited completely and at the required detail level.

Therefore, many methods were invented and used to provide *estimates* of functional size measures based on less or coarser grained information than required by standard FPA. Among those methods, the NESMA method [3] is the most popular. Actually, the NESMA method was adopted by IFPUG as the election method for early estimation of unction Point size [4].

Inevitably, all the early functional size estimation methods involve some estimation error. Accordingly, project managers need to know—at least approximately—the magnitude of the potential error that affects size estimates. This is especially true for the NESMA method, since it has been proposed as the "official estimation method by IFPUG.

Not surprisingly, several researchers and practitioners evaluated the accuracy of the proposed functional size estimation methods (as described in Section III). However, most evaluations were based on academic software projects or used small datasets, hence most evaluations cannot be considered very reliable, and they are hardly generalizable. In order to assess the actual value for industry of the NESMA method, it is necessary to perform an experimental evaluation based on a large dataset collecting measures from industrial settings.

In this paper, we present the experimental evaluation of the accuracy of NEMSA estimates, based on a dataset that includes data from 479 software development projects in the finance and banking domain. The size of the dataset and the real-life nature of the data make the evaluation presented here the most exhaustive and reliable published evaluation of the NESMA method.

The rest of the paper is organized as follows. Section II briefly describes FPA and the NESMA functional size estimation measurement method. Section III illustrates the related work. Section IV describes the empirical study. Section V discusses the threats to the validity of the study.

Finally, Section VI draws some conclusions and outlines future work.

II. BACKGROUND

To make the paper easier to read and as self-contained as possible, in this section we outline the fundamentals of Function Point Analysis and the NESMA estimation methods.

A. Function Point Analysis

The basic idea of FPA is that the "amount of functionality" released to the user can be evaluated by taking into account the data used by the application to provide the required functions, and the transactions (or processes) through which the functionality is delivered to the user. Specifically, the data used by the application are classified into Internal Logic Files (ILF) and External Logic Files (EIF), the latter being essentially "read only" for the application being measured. Transactions are classified into External Input (EI), External

Output (EO) and External Queries (EQ), depending on their main purpose.

According to the counting manual [2], the measurement process is organized into the following activities: 1) Determining the type of function point count; 2) Identifying the Counting Scope and Application Boundary; 3) Identifying Data Functions; 4) Identifying Transaction Functions; 5) Weighting data and transaction functions. The latter activity requires that each data and transaction function is analyzed in detail, so that its "complexity" is determined and the corresponding weight can be assigned to the function. Activity 5) is relatively time and effort consuming.

For additional information on Function Point measurement, please see [2]. Unadjusted Function Points have been recognized as an international standard by ISO [5]. In this paper, we always refer to unadjusted function points, even when we do not qualify explicitly measures as unadjusted.

B. The NESMA estimated method

The NESMA method was proposed [3] to get an estimate of the functional size of a given application without analyzing data and transactions in detail.

Actually, there are two NESMA estimation methods: the Indicative NESMA method and the Estimated NESMA method.

The former estimates size (*EstSize*) based on the number of ILF (#ILF) and the number of EIF (#EIF), as follows:

$$EstSize = \#ILF \times W_{ILF} + \#EIF \times W_{EIF}$$

where W_{ILF} is 25 or 35 and W_{EIF} is 10 or 15, depending on ILF and EIF being identified based on a conceptual model in third normal form or not, respectively.

The process of applying the NESMA indicative method involves only identifying logic data and classifying them as ILF or EIF. Accordingly, it requires less time and effort than standard FPA. However, the Indicative NESMA method is quite rough in its computation: the official NESMA counting manual specifies that errors in functional size with this approach can be up to 50%.

The Estimated NESMA method requires the identification and classification of all data and transaction functions, but does not require the assessment of the complexity of each function: Data Functions (ILF and EIF) are assumed to be of low complexity, EI, EQ and EO are assumed to be of average complexity. Hence, estimated size is computed as follows:

$$EstSize = 7\ \#ILF + 5\ \#EIF + 4\ \#EI + 5\ \#EO + 4\ \#EQ$$

IFPUG adopted the NESMA estimated method as the official early function point analysis method [4].

C. Function Point counting and estimation example

In this section, we illustrate IFPUG FP counting and NESMA estimation using a slightly modified version of the Warehouse management software (WMS) by Fetcke [6].

The WMS is used by a company that operates several warehouses, where customers' goods are stored. Customers can deposit items into storage locations in the warehouse. After the items have been kept in the warehouse for some period of time, they can be retrieved by their owners. The customers get billed for the storage service.

The Entity/Relationship diagram representing the entities involved in the WMS is given in Figure 1. The entities and their attributes are described in Figure 2. Attributes `Owner` and `Storage_place` are references to entities `Customer` and `Place`, respectively.

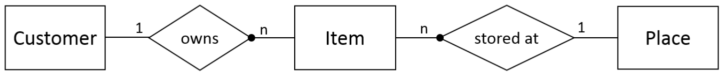

Figure 1. Entity/Relationship diagram of the WMS [6].

The WMS allows the user to perform several operations, such as adding a new customer, deposit an item, receive payment, print the customer item list, and many others. Here we report the specifications of the `Delete_customer` operation, which will be used to illustrate the functional measurement methods. The `Delete_customer` operation removes an instance of `Customer` from the system's repository, given the `Name` attribute of the customer. `Customer` data are removed if the `Amount_due` attribute is zero and the customer does not own any stored items. An error message is displayed, if the record cannot be removed or if there is no instance of `Customer` with the given name.

Based on the given specifications, we can measure the size of `Item` data file and the `Delete customer` transaction, according to IFPUG rules, as follows.
`Item` is an ILF, since it is managed by the WMS application. It has just one RET, since the only type of instance for `Items` is the one shown in Figure 2. According to the same figure, `Item` has 6 attributes, i.e., 6 DETs. Having 1 RET and 6 DETs, `Item` is a low complexity ILF, hence its size is 7 FP [2].
`Delete_customer` is an EI, since its main objective consists in updating a data file, namely the `Customer`. This transaction reads `Item` instances and reads and (possibly) deletes an istance of `Customer`: accordingly, it references 2 types of files (FTR = 2). In the execution of this transaction, only two DETs are moved through the boundaries: the `Name` given in input to identify the customer to be removed, and the error message issued if the deletion cannot be performed. As an EI with 2 FTR and 2 DETs it has low complexity and its size is 3 FP [2].

We can now compute the estimated size of `Item` data file and `Delete_customer` transaction using NESMA methods.

According to the NESMA estimated method, `Item` (an ILF) is assumed to be of low complexity, hence its size is estimated to be 7 FP, `Delete_customer` (an EI) is assumed to be of average complexity, hence its size is estimated to be 4 FP.

Using the NESMA indicative method, we can estimate the

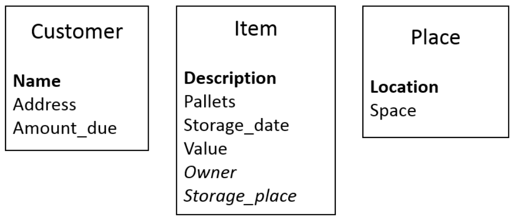

Figure 2. Entities of the WMS [6].

size of the whole WMS: Figure 1 shows that a third-normal fiorm model of data includes 3 ILF and no EIF, hence the expected size of WMS is $3 \times 25 = 75$ FP.

III. Related work

NESMA defined the application of FPA in the early phases of the application life cycle, and recognizes three function point analysis methods: Detailed function point analysis (currently corresponding to IFPUG measurement), Estimated function point analysis, Indicative function point analysis. Using a database of over 100 developed and implemented applications, NESMA empirically evaluated the accuracy of the estimated and indicative FPA approximation methods [10]. The results showed that: size measures of the high-level function point analysis and the detailed function point analysis are very close. Moreover, indicative function point analysis gives a surprisingly good estimate of the size of several applications.

H. S. van Heeringen described the size accuracyas well as the difference in measurement effortof the NESMA estimated and NESMA indicative methods, by measuring 42 projects [3]. The results show that the estimation error of NESMA estimated was in the [-6%, +15%] range, with average 1.5%; the estimation error of NESMA indicative was in the [-15%, +50%] range with average 16.3%. In both cases the estimation error was evaluated with respect to the detailed measurement.

Wilkie et al. [11] utilized five commercial projects used in the research to evaluate the cost-benefit trade-off of size measurement with respect to size estimation; they concluded that whilst the Indicative NESMA method was insufficiently accurate for the involved commercial organization, the NESMA Estimated approach was definitely viable.

IFPUG adopted NESMA methods for early "high-level" size estimation [4]. IFPUG suggested that 1) The High Level FPA method can be used to size an application early in the software development life cycle; 2) The High Level FPA method can also be applied as an alternative to standard FPA estimate (the outcome is not significantly different, while the estimation time is considerably shorter); 3) The indicative FPA method may be used to get a very fast, rough indication of the size of a project or an application, but it is not suited for contractual commitments.

Lavazza et Liu [9] used 7 real-time applications and 6 non real-time applications to evaluate the accuracy of the E&QFP [12] and NESMA with respect to full-fledged Function Point Analysis. The results showed that the NESMA indicative method yields the greatest errors. On the contrary, the NESMA estimated method yields size estimates that are close to the actual size. The NESMA indicative method is generally outperformed by the other methods. The NESMA estimated method proved fairly good in estimating both Real-Time and non Real-Time applications.

Morrow et al. used a dataset of 11 projects to evaluate the quality of sizing estimates provided by NESMA methods [13]. They also adapted NESMA methods' general principles to enhance their accuracy and extent of relevance, and empirically validated such an adapted approach using commercial software projects.

The main limitations of the mentioned research are that most of the research work used small datasets containing data concerning little projects of not industrial nature. In our paper, we evaluate measurement accuracy of the NESMA method with respect to FPA method over a dataset containing data from 479 industrial projects, of which several are above 10000 FP.

IV. The Empirical Study

The empirical study was made possible by the availability of a dataset that includes—for every application—both the measure in IFPUG UFP and the number of ILF, EIF, EI, EO and EQ. Using the latter numbers we were able to compute NESMA estimates, by applying the formulae given in Section II-B. So, having both the NESMA estimates and the IFPUG size for every application, we were able to evaluate the accuracy of estimates.

A. The Dataset

We use a dataset of 479 projects developed and used by a Chinese financial enterprise. The considered projects are all new development projects, that delivered applications to be employed in the financial and banking domain. The measures in the dataset were all computed by expert professionals.

TABLE I. Descriptive statistics of dataset.

			NESMA indicative	
	Standard FP	NESMA estimated	Not Normalized	Normalized
Max	82501	87100	87420	61895
Mean	3567	3435	3456	2438
St. Dev.	6725	6694	7258	5120
Median	1135	1122	985	700

Descriptive statistics of the dataset are given in Table I.

B. The Analysis

To assess the obtained estimates, in Figure 3 we plot the values of the estimates with respect to the actual size measured according to the standard IFPUG counting manual [2]. In Figure 3, we also draw the NESMA estimated = UFP line: if the estimates were perfect, all the points would lie on the line. As a matter of fact, most points are quite close to the line, thus indicating that in general the estimates are close to the actual measures.

To better appreciate the accuracy of estimates, in Figure 4 the situation for projects having size not greater than 20000 UFP is shown. It can be observed that most points are below the y=x line, thus indicating that the NESMA method tends to underestimate.

To verify if and to what extent the NESMA method underestimates the IFPUG size, in Figure 5 a boxplot of NESMA estimation errors is given (errors are defined as the standard IFPUG size measure minus the estimate). For the sake of readability, in Figure 5, errors having magnitude greater than 1000 UFP are not shows. It can be seen that both the median and the mean (shown as a blue diamond) are above the zero error line. Actually, about 75% of the projects are underestimated. We can thus conclude that—in the considered dataset—the NESMA method underestimates functional size.

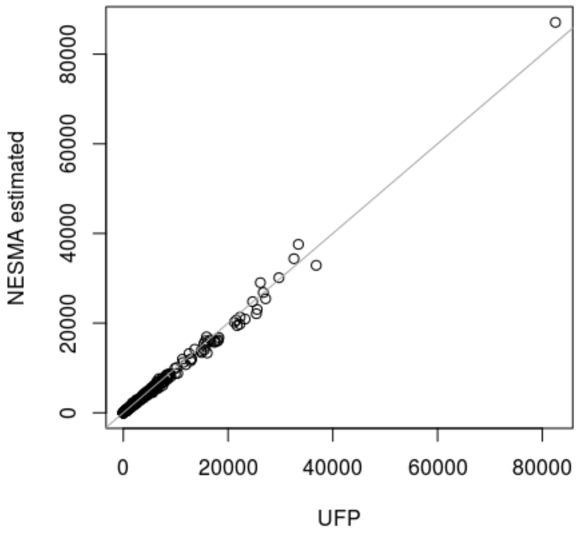

Figure 3. Standard IFPUG UFP measures vs. NEMSA estimates.

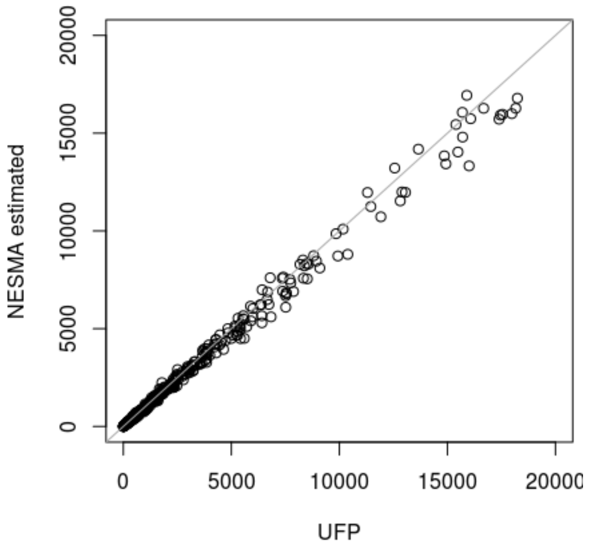

Figure 4. Standard IFPUG UFP measures vs. NEMSA estimates: zoom on projects not greater than 20000 UFP.

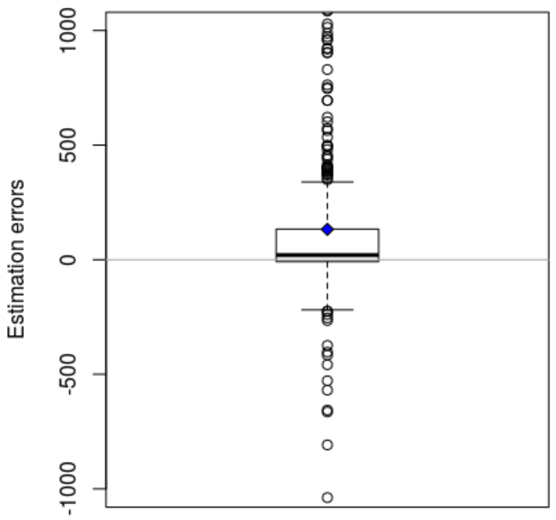

Figure 5. Boxplot of NEMSA estimation errors (no outliers).

C. Accuracy Evaluation

It is now necessary to evaluate quantitatively the accuracy of NESMA estimates. First of all—as suggested by Shepperd and McDonell [7] and by Lavazza and Morasca [8]—we checked whether NESMA estimates perform better than "baseline" models. Shepperd and MacDonell [7] proposed that the accuracy of a given estimation method be measured via the Mean Absolute Residual (MAR): $MAR = \frac{1}{n}\sum_{i=1..n}|y_i - \hat{y}_i|$. Shepperd and MacDonell suggest to use random estimation, based solely on the known (actual) values of previously measured applications, as a baseline model. Shepperd and MacDonell observed also that the value of the 5% quantile of the random estimate MARs can be interpreted like α for conventional statistical inference, that is, any accuracy value that is better than this threshold has a less than one in twenty chance of being a random occurrence. Accordingly, the MAR of a proposed model should be compared with the 5% quantile of the random estimate MARs, to be reasonably sure that the model is actually more accurate than the random estimation.

Lavazza and Morasca [8] proposed to use a "constant model," where the estimate of the size of the i^{th} application is given by the mean size of the other applications.

With our dataset, the MAR of the constant model is 3864 UFP, while the 5% quantile of absolute residuals for random estimates is 4566 UFP. The MAR of NESMA estimates is 246 UFP, much smaller than both baselines. Consequently, we can state that the NESMA method satisfies the necessary conditions for being considered an acceptable estimation method.

Concerning the accuracy of NESMA estimates, in Figure 6 the distribution of absolute errors is given. The blue diamond is the mean, i.e, the MAR of the estimates. The median of absolute residuals is 57 UFP, however the MAR is definitely greater (246 UFP), because of several large errors.

In general, relatively large estimation errors are deemed acceptable in very large projects. To help practitioners appreciate the "importance" of errors with respect to the size of the project, in Figure 7 we give the boxplot representing the distribution of absolute relative errors (the relative error of an estimate is the estimation error divided by the actual size).

Figure 7 shows that the great majority of estimate errors are less than 10%, and in only a few (out of 479) cases, errors are greater than 25%. Considering that NESMA estimates are produced without considering the details of functional requirements, this level of accuracy is likely acceptable by most practitioners.

D. Analysis of the Indicative Method

The analysis reported above concentrated on the NESMA Estimated method, since the NESMA Indicative method is reported to be much less accurate [3],[9].

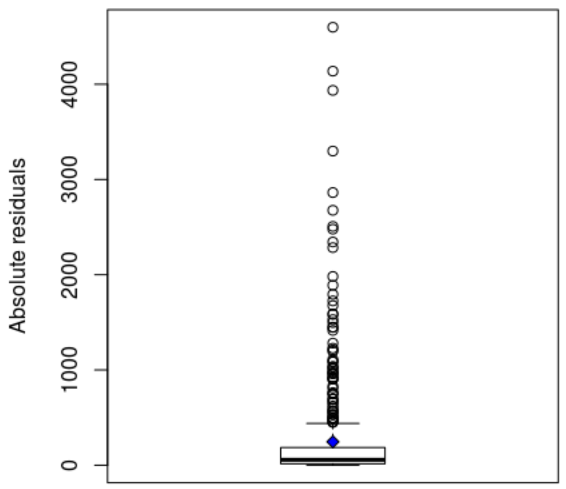

Figure 6. Boxplot of absolute errors.

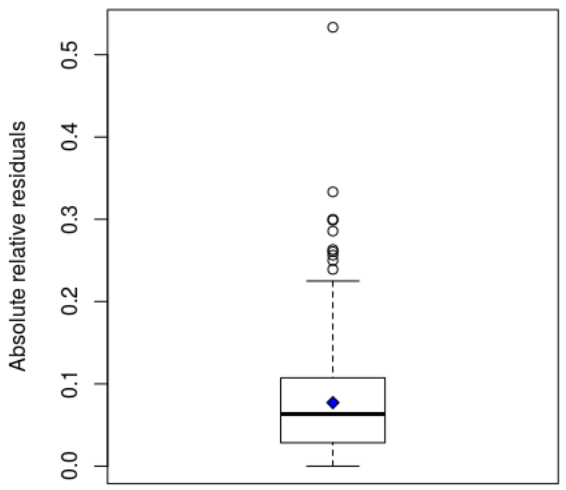

Figure 7. Boxplot of absolute relative errors.

However, we consider necessary to check if the NESMA Indicative method is really less accurate than the NESMA Estimated method. To this end, for each of the 479 project in the dataset, we computed the estimated size according to the NESMA Indicative method. Since we had no reliable information concerning the normalization of the data models used to identify ILF and EIF, we applied the NESMA Indicative method with both the normalized and not normalized weights.

The results we achieved agree with the previously published evaluations. Figure 8 shows the distributions of relative errors of estimates obtained via NESMA Indicative methods in comparison with the NESMA Estimated method. It is quite clear that the NESMA Indicative methods is definitely less accurate than the NESMA Estimated method. This difference in accuracy is even more evident in Figure 9, where the absolute residuals of NEMSA methods are shown (outliers are

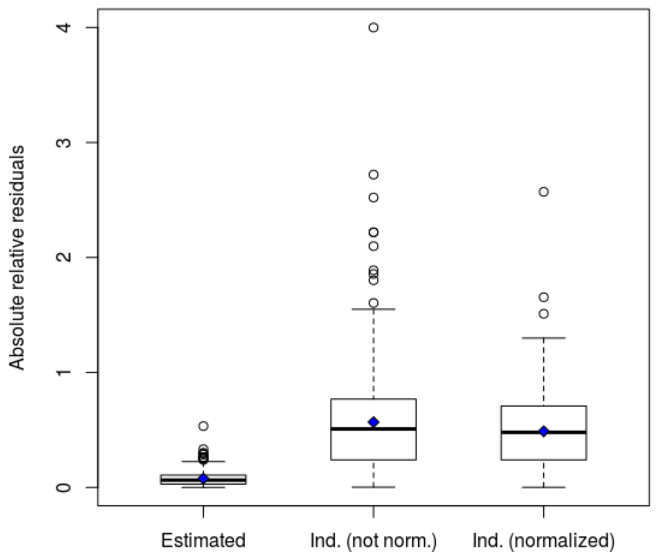

Figure 8. Distributions of absolute relative estimation errors of the NESMA methods.

omitted to keep the figure readable).

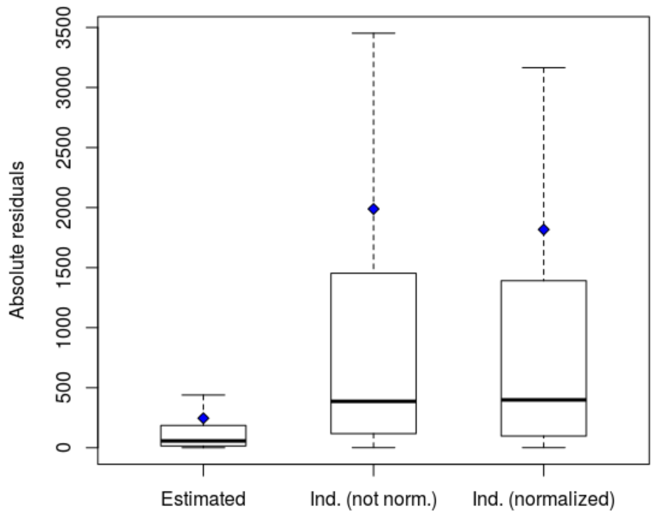

Figure 9. Distributions of absolute relative estimation errors of the NESMA methods.

TABLE II. RESULTS WITH DIFFERENT NESMA METHODS

		Indicative	
	Estimated	Not Norm.	Normalized
Mean AR	264	1989	1817
Median AR	57	386	398
Mean MRE	7.7%	57%	49%
Median MRE	6.3%	51%	48%

The MAR and MMRE obtained with the different NESMA methods are given in Table II.

V. THREATS TO VALIDITY

Given the type of study we presented, there are two main threats to validity that need attention.

First, we should consider the correctness of the given data. In fact, the data in the analyzed dataset were derived from the analysis and measurement of functional requirements: both analysis and measurement could be affected by error, which would end up in the dataset. Concerning this threat, we are reasonably sure that the used data are of good quality, since they were collected by professionals in industrial contexts where functional size measures are quite important, hence great attention is posed in the measurement activities. Evan so, we cannot exclude that some errors are present; however, in such case most errors are expected to affect both IFPUG measures and NESMA estimates, in approximately the same way. Hence, these errors should not be able to affect our results to a large extent.

Second, we need to consider external validity, i.e., whether we can generalize the results of our study outside the scope and context that characterize the considered software projects. On the one hand, our dataset is much larger than the datasets usually involved in software engineering empirical studies; besides, our dataset includes data from fairly large projects (e.g., over 20000 FP). In this sense, our dataset represents a large and varied enough sample. On the other hand, all the considered projects are from the economic, financial and banking domain, hence we cannot be sure that the results of our study apply equally well in other domains. In this respect, readers are reminded that previous studies (e.g., [9]) show some difference in accuracy when estimates concern real-time software applications.

VI. Conclusions

In this paper, we addressed the evaluation of the accuracy of functional size estimates that can be achieved via the NESMA estimation methods. To this end, we compared functional size measures obtained via the standard IFPUG Function Point Analysis process, and estimates obtained via the NESMA indicative and NESMA estimated methods. Both measures and estimates were computed for a dataset containing data from 479 software projects. Based on the results of the analysis, we can draw a few relevant conclusions:

– The NESMA estimated method is definitely more accurate than the NESMA indicative method.
– The NESMA estimated method provides reasonably accurate estimates: the mean absolute residual is 264 FP, quite small, considering that the average size of estimated projects is 3567 FP.
– 75% of applications were estimated by the NESMA estimated method with errors not greater than (or extremely close to) 10%.
– The NESMA method tends to underestimate. This can be dangerous, since at the initial stages of development one could be induced to believe that the development process will be shorter and cheaper than actually required.

Future work includes experimenting with additional estimation methods and investigating whether and how estimation accuracy can be improved.

Acknowledgment

This work was partly supported by the “Fondo di ricerca d’Ateneo” funded by the Università degli Studi dell’Insubria, by Zhejiang Provincial Science Foundation of China under Grant No. LY19F020046 and LY17F020023, and by the Chinese Scholarship Council under Grant No. 201708330399

References

[1] A. J. Albrecht, “Measuring application development productivity,” in Proceedings of the joint SHARE/GUIDE/IBM application development symposium, vol. 10, 1979, pp. 83–92.
[2] International Function Point Users Group (IFPUG), “Function point counting practices manual, release 4.3.1,” 2010.
[3] H. van Heeringen, E. van Gorp, and T. Prins, “Functional size measurement-accuracy versus costs-is it really worth it?” in Software Measurement European Forum (SMEF 2009), 2009.
[4] A. Timp, “uTip – Early Function Point Analysis and Consistent Cost Estimating,” 2015, uTip # 03 (version # 1.0 2015/07/01).
[5] International Standardization Organization (ISO), “ISO/IEC 20926: 2003, Software engineering IFPUG 4.1 Unadjusted functional size measurement method Counting Practices Manual,” 2003.
[6] T. Fetcke, “The warehouse software portfolio: A case study in functional size measurement,” Département d’informatique, Université du Quebec à Montréal, Tech. Rep. 1999 20, 1999. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.195.5828&rep=rep1&type=pdf Retrieved: September 2019
[7] M. Shepperd and S. MacDonell, “Evaluating prediction systems in software project estimation,” Information and Software Technology, vol. 54, no. 8, 2012, pp. 820–827.
[8] L. Lavazza and S. Morasca, “On the evaluation of effort estimation models,” in Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering.   ACM, 2017, pp. 41–50.
[9] L. Lavazza and G. Liu, “An empirical evaluation of simplified function point measurement processes,” International Journal on Advances in Software Volume 6, Number 1 & 2, 2013, 2013, pp. 1–13.
[10] nesma, “Early Function Point Analysis.” [Online]. Available: https://nesma.org/themes/sizing/function-point-analysis/early-function-point-counting/ Retrieved: September 2019
[11] F. G. Wilkie, I. R. McChesney, P. Morrow, C. Tuxworth, and N. Lester, “The value of software sizing,” Information and Software Technology, vol. 53, no. 11, 2011, pp. 1236–1249.
[12] L. Santillo, M. Conte, and R. Meli, “Early & quick function point: sizing more with less,” in 11th IEEE International Software Metrics Symposium (METRICS’05).   IEEE, 2005, pp. 41–41.
[13] P. Morrow, F. G. Wilkie, and I. McChesney, “Function point analysis using nesma: simplifying the sizing without simplifying the size,” Software Quality Journal, vol. 22, no. 4, 2014, pp. 611–660.