# Functional Complexity Measurement: Proposals and Evaluations

Luigi Lavazza
Dipartimento di Informatica e Comunicazione
Università degli Studi dell'Insubria
Varese, Italy
luigi.lavazza@uninsubria.it

Gabriela Robiolo
Departamento de Informática
Universidad Austral
Buenos Aires, Argentina
grobiolo@austral.edu.ar

***Abstract* — Several definitions of measures that aim at representing the size of software requirements are currently available. These measures have gained a quite relevant role, since they are one of the few types of objective data upon which effort estimation can be based. However, traditional Functional Size Measures do not take into account the amount and complexity of elaboration required, concentrating instead on the amount of data accessed or moved. This is a problem, when it comes to effort estimation, since the amount and complexity of the required data elaboration affect the implementation effort, but are not adequately represented by the current measures, including the standardized ones. Recently, a few approaches to measuring aspects of user requirements that are supposed to be related with functional complexity and/or data elaboration have been proposed by researchers. The authors of this paper have also proposed a measure of the functional complexity as specified in user requirements. In this paper we take into consideration some of these proposed measures and compare them with respect to their ability to predict the development effort, especially when used in combination with COSMIC measures of functional size.**

***Keywords-Functional size measurement; Function Points; COSMIC function points; effort estimation; functional complexity measurement.***

## I. INTRODUCTION

COSMIC function points [8][12] are growingly used for measuring the functional size of applications, i.e., to measure the size of functional user requirements. The measure of functional size is typically used to drive the estimation of the development effort. To this end, effort models require several inputs in addition to the functional size, including the complexity of the software to be [3][7]. In fact, problem complexity is recognized as one of the elements that contribute to the comprehensive notion of software size [9].

The need to account for software complexity when estimating the development effort does not depend on the functional size measurement method used: for instance, when more traditional measures of the functional size –like IFPUG function points [12]– are used, complexity has to be accounted for as well.

Actually, both COSMIC and IFPUG function points fail to represent the amount and complexity of data elaboration required. COSMIC function points concentrate on the measure of the data movements, neglecting the data elaboration. More precisely, the model of software used by the COSMIC method –illustrated in Figure 1–includes data elaboration, but no indication on how to measure it is provided. The COSMIC measurement manual [8] simply assumes that every data movement accounts for some amount of data elaboration, and that such amount is proportional to the number of data movements, so that by measuring data movements one measures also data manipulation.

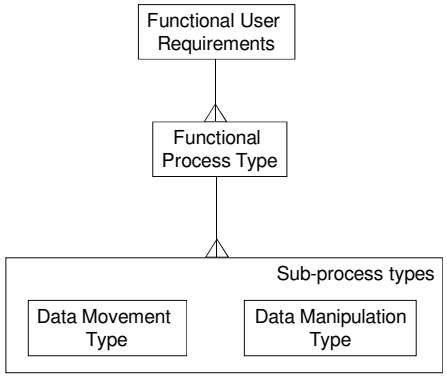


Figure 1. The COSMIC generic software model.

Before proceeding, it is useful to spend some words on the fact that throughout the paper we treat the terms "complexity" and "amount of data elaboration" as synonyms. This is due to the fact that complexity is an inherently elusive concept, and also to the fact that at the functional requirements level it is not clear what should be the difference between the amount and the complexity of data elaboration: for instance, in many cases, complexity is considered proportional to the number of alternatives in a process execution, but this number is also clearly related to the size of the process.

When dealing with effort estimation, the most popular methods require an evaluation of the complexity of the application. Currently such evaluation is of a purely qualitative nature. For instance, COCOMO II [7] provides a table that allows the user to evaluate complexity on an ordinal scale (from "very low" to "extra high") according to five aspects (control operations, computational operations, device-dependent operations, data management operations, user interface management operations) that have to be evaluated in a qualitative and subjective way: e.g., the characterization of computational operations corresponding

 ISBN: 978-1-61208-165-6 

to the "Nominal" complexity is "Use of standard math and statistical routines. Basic matrix/vector operations" [7].

It is quite clear that it would be greatly beneficial to replace such subjective and approximate assessment of complexity with a real measure, based on objective and quantitative evaluations, since this would enable the construction of more objective and accurate models of effort.

Several different possible measures of functional complexity were proposed. For instance, in [5] the number of inputs and outputs, the number of decision nodes, the sum of predicates of all decision nodes, the depth of decision tree and the length of paths are considered as possible indicators of complexity.

In [6], Cao et al. propose the usage of the number of data groups (NOD), the number of conditions (NOC) and entropy of system (EOS). They also study how these measures (also in combination with COSMIC FP) are correlated with the development effort.

Another measure of complexity, the Paths, was defined on the basis of the information typically available from use case descriptions [21]. The measure of the complexity of use cases is based on the application of the principles of McCabe's complexity measure [18] to the descriptions of use cases in terms of scenarios. In fact, use cases are usually described giving a main scenario, which accounts for the 'usual' behaviour of the user and system, and a set of alternative scenarios, which account for all the possible deviations from the normal behaviour that have to be supported by the system. Robiolo and Orosco [21] apply to the use case textual descriptions the same measure applied by McCabe to code. Every different path in a given use case scenario contributes to the measure of the use case's complexity. The definition of Paths conforms to several concepts enounced by Briand et al. [4]: Paths represent "an intrinsic attribute of an object and not its perceived psychological complexity as perceived by an external observer", and they represent complexity as "a system property that depends on the relationship between elements and is not an isolated element's property". A detailed description of the Paths measure and its applicability to use cases described in UML can be found in [15].

Previous work showed that effort models that take into consideration complexity measures are more precise than those based on the functional size only. In particular, the authors of this paper showed that development effort correlates well with COSMIC function points and Path [15], and that the inclusion of a Path-based complexity measure improves the models based on size, whatever size measure is used (IFPUG Function Points, CFP, or even Use Case Points) [16].

In this paper we enhance the dataset used in [16] with some measures that represent potential complexity dimensions, build effort estimation models that exploit these measures, and discuss the precision of fit of these models.

The results of the measurements and analyses reported in the paper contribute to enhancing the knowledge of how it is possible to measure functional complexity at the requirements level, and what is the contribution of such measure to effort estimation.

## II. THE EXPERIMENTAL EVALUATION

In the research work reported here, we used measures that are conceptually very close to those proposed in previous studies [5][6]. However, we did not stick exactly to the previous proposals, essentially for practical reasons. We used Paths instead of NOC because both measures capture essentially the same meaning, and the measures of Paths were already available. Similarly, we used the number of data groups instead of NOD, because –having measured the size of the applications in CFP, the documentation on the data groups was already available, thus the measurement could be performed very easily.

Finally, we decided to use another "by product" of CFP measurement, namely the number of functional processes, as a simplified measure of size.

### A. The Dataset

In order to evaluate the measures mentioned above with respect to their usability as effort predictors, we collected all such measures for a set of projects. We could not use data from the best known repositories –such as the PROMISE or ISBSG– because they do not report the size of each project according to different FSM methods; moreover, the Paths measure is very recent, and no historical data exist for it.

TABLE 1. THE DATASET

| ProjID | Actual effort | Path | CFP | Func. Proc. | Data groups | Pers. DG |
|---|---|---|---|---|---|---|
| P1 | 410 | 71 | 143 | 39 | 21 | 7 |
| P2 | 473.5 | 73 | 118 | 28 | 15 | 9 |
| P3 | 382.4 | 60 | 109 | 24 | 15 | 12 |
| P4 | 285 | 49 | 74 | 25 | 14 | 8 |
| P5 | 328 | 34 | 48 | 12 | 17 | 7 |
| P6 | 198 | 35 | 67 | 10 | 15 | 7 |
| P7 | 442.02 | 50 | 81 | 16 | 12 | 6 |
| P8 | 722.65 | 97 | 115 | 27 | 19 | 10 |
| P9 | 392 | 83 | 105 | 24 | 22 | 11 |
| P10 | 272 | 42 | 73 | 21 | 9 | 9 |
| P11 | 131 | 18 | 51 | 13 | 5 | 5 |
| P12 | 1042 | 118 | 85 | 30 | 29 | 12 |
| P13 | 348 | 32 | 46 | 12 | 12 | 6 |
| P14 | 242.5 | 68 | 96 | 26 | 18 | 9 |
| P15 | 299.76 | 33 | 54 | 12 | 12 | 4 |
| P16 | 147 | 20 | 53 | 14 | 15 | 4 |
| P17 | 169 | 17 | 30 | 5 | 10 | 6 |

We measured 17 small business projects, which were developed in three different contexts: an advanced undergraduate academic environment at Austral University, the System and Technology (S&T) Department at Austral

University and a CMM level 4 Company. The involved human resources shared a similar profile: advanced undergraduate students who had been similarly trained worked both at the S&T Department and at the CMM level 4 Company. All the selected projects met the following requisites:

a) Use cases describing requirements were available.
b) All projects were new developments.
c) The use cases had been completely implemented, and the actual development effort in PersonHours was known.

The dataset is reported in TABLE 1. Note that we distinguished the number of persistent data groups (column *Pers. DG*) from the total number of data groups, which includes also transient data groups. Our hypothesis is that persistent data groups are more representative of the amount of data being handled by the application.

### B. Analysis of the dataset using log-log transformations

As a first approach to evaluating the correlation of effort with other measures, we used linear regression after log-log transformation, as is usually done in studies concerning effort (see for instance COCOMO [3][7]).

We started by checking the correlation between effort and CFP. The results are not very good: after eliminating outliers, we got a model featuring adjusted $R^2 = 0.335$.

Then we moved to univariate analysis of the correlation between Effort and each variable mentioned in TABLE 1:

– Path [Path]
– COSMIC Function Points [CFP]
– Functional Processes [FPr]
– Data Groups [DG]
– Persistent Data Groups [PDG]

We also systematically tested the correlation between effort and the following *density* measures:

– Path per Functional Process [Path/FPr]
– Path per CFP [Path/CFP]
– Data Groups per Functional Process [DG/FPr]
– Data Groups per CFP [DG/CFP]
– Persistent Data Groups per Functional Process [PDG/FPr]
– Persistent Data Groups per CFP [PDG/CFP]

These density measures introduce the concept of complexity per size unit. The complexity of a system is a property that depends on the relationships among system's elements [4]. So, the measures listed above represent the density of relationships among elements per unit size. As size units we adopted both the fine grained CFP and the coarse grained number of functional processes. In fact, the number of functional processes is suggested as a reasonable approximation of the size in CFP in [8].

Quite interestingly, we got significant models only based on variables involving Paths. The results are synthetically reported in TABLE 2. For each model, we have also assessed the precision of the fit by using what are considered the de facto currently used goodness-of-fit indicators in Empirical Software Engineering, i.e., the Mean Magnitude of Relative Error (MMRE) and the percentage of data points whose actual effort falls within 75% and 125% of the estimated value (pred(25)) and the error range.

In TABLE 2 are reported only the models that satisfy the applicability conditions of linear regression (e.g., the residuals are normally distributed), are statistically significant (e.g., their p-value is < 0.05), and have coefficient of determination (Adjusted $R^2$) sufficiently high (>0.6).

TABLE 2. CORRELATIONS WITH EFFORT (LOG-LOG UNIVARIATE REGRESSION)

| Var. | Adj. $R^2$ | p-value | Outl. | MMRE | Pred(25) | Error range |
|---|---|---|---|---|---|---|
| Path | 0.79 | $< 10^{-5}$ | 2 | 22.7 | 70.6 | -35%.. 82% |
| Path/FPr | 0.73 | $< 10^{-3}$ | 5 | 37.2 | 58.8 | -48% .. 169% |
| Path/CFP | 0.65 | $< 10^{-4}$ | 0 | 24.1 | 52.9 | -43% .. 66% |

The regression line of the model representing Effort vs. Paths –which appears as the best univariate model– is illustrated in Figure 2.

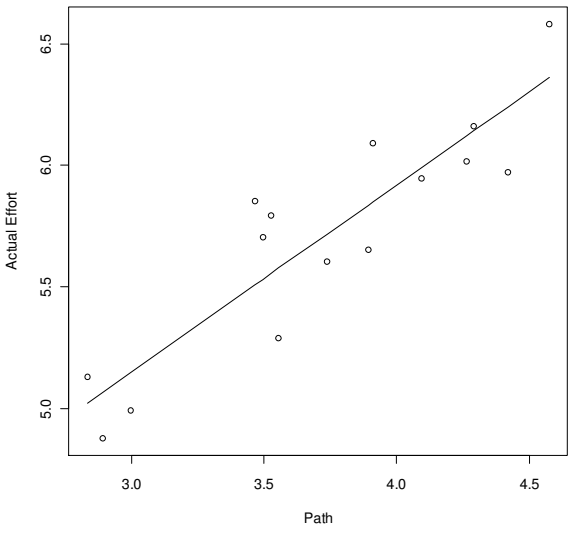


Figure 2. Effort vs. Path: log-log regression line.

The distribution of relative residuals is given in Figure 3.

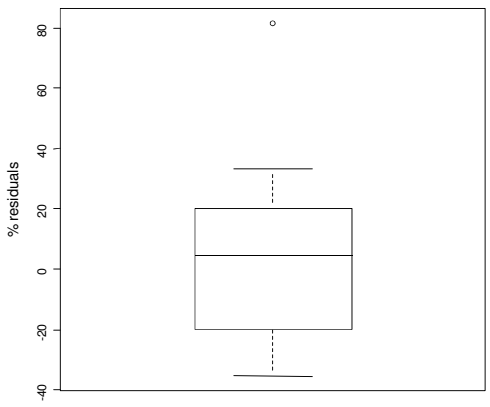


Figure 3. Log-log regression of effort vs. Path: distribution of relative residuals.

We then proceeded to the analysis via multiple regression. Again, we systematically tested the correlation of Effort with any combination of the aforementioned variables.

The statistically significant models obtained are reported in TABLE 3.

TABLE 3. CORRELATIONS WITH THE ACTUAL EFFORT (LOG-LOG MULTIPLE REGRESSION)

| Var. | Adj. $R^2$ | Pr(>\|t\|) | Outl. | MMRE | Pred(25) | Error range |
|---|---|---|---|---|---|---|
| FPr, Path /FPr | 0.71 | $< 10^{-3}$ | 1 | 19.4 | 70.6 | -37%.. 67% |
| FPr, Path /CFP | 0.78 | $< 10^{-3}$ | 0 | 18.7 | 82.4 | -31% .. 80% |
| Path/FPr, PDG/FPr | 0.64 | $< 10^{-3}$ | 1 | 22.9 | 64.7 | -45% .. 53% |
| Path/CFP, DG/FPr | 0.69 | < 0.03 | 1 | 22.6 | 64.7 | -30% .. 73% |
| Path/CFP, DG/CFP | 0.72 | <0.02 | 1 | 21.3 | 64.7 | -41%.. 69% |
| Path/CFP, PDG/FPr | 0.75 | <0.02 | 0 | 18.5 | 70.6 | -35%.. 74% |
| Path/CFP, PDG/CFP | 0.80 | $< 10^{-2}$ | 0 | 17.8 | 82.4 | -36%.. 72% |
| DG, Path/FPr, DG/FPr | 0.75 | $< 10^{-2}$ | 0 | 18.6 | 70.6 | -29%.. 76% |
| Path/FPr, DG/CFP, PDG/FPr | 0.67 | <0.05 | 3 | 23.9 | 58.8 | -66%.. 79% |

It is quite interesting to see that none of the obtained models uses size in CFP as an independent variable. On the contrary, most of the other variables (including size expressed as number of Functional Processes, computation density, amount of data and data density) can be used to build valid and significant models.

It is also interesting to see that these models appear quite good both in terms of their ability to explain the variation of effort depending on the variation of the size and complexity measures (as indicated by the values of the adjusted $R^2$) and in terms of precision of the fit (as indicated by MMRE, pred(25) and the relative error range).

Although it is quite clear that some models appear better than others, e.g., with respect to precision of fit and adjusted $R^2$, it is not so obvious which one is best.

A possible way for identifying the best model is by comparison of the relative absolute residuals (since we are considering the ability to predict effort, we have to look at *relative* absolute residuals, since an error of, say, two PersonMonths can be irrelevant or very important, depending on the total effort). The models that feature the highest values of the adjusted $R^2$ are those based on

a) Paths
b) Path per CFP and PersistentDataGroups per CFP
c) Functional Processes and Path per CFP

The boxplots representing relative absolute residuals of these models are reported in Figure 4 .

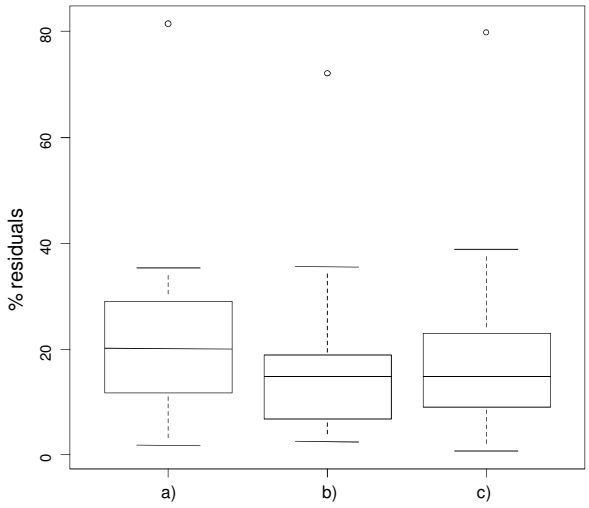


Figure 4. Model comparison: relative absolute residuals.

The comparison of boxplots does not allow selecting a model as clearly the best, although it seems that the univariate model is a bit less precise than both the other two models. In order to evaluate whether a model can be elected the best, Kitchenham et al. [14] suggest to use paired tests of the absolute residuals. We then proceeded to compute the paired tests. We used t-tests when appropriate (i.e., when the distributions were close to normal) and the Wilcoxon signed rank test otherwise. Also the paired tests did not indicate a clear winner. Therefore, we must conclude that further research is needed to understand if it is possible to build a model that explains in the best possible manner the dependency of effort from size and complexity measures.

### C. *Analysis of the dataset using plain linear regression*

Having performed the analysis on log-log transformed data, we checked if valid and significant models can be built using ordinary least squares (OLS) linear regression, i.e., without log-log (or any other) transformation.

We found that a linear model linking Effort and Paths exists: it features adjusted $R^2 = 0.71$, p-value $< 10^{-3}$, MMRE = 23.5%, Pred(25) = 58.8%, Error range = -33% .. 81%.

The models involving two independent variables are summarized in TABLE 4.

TABLE 4. CORRELATIONS WITH THE ACTUAL EFFORT (OLS MULTIPLE REGRESSION)

| Var. | Adj. $R^2$ | Pr(>\|t\|) | Outl. | MMRE | Pred(25) | Error range |
|---|---|---|---|---|---|---|
| CFP, Path /CFP | 0.82 | $< 10^{-3}$ | 4 | 18.5% | 76.5% | -20%.. 84% |
| FPr, Path/CFP | 0.64 | $< 10^{-2}$ | 3 | 20% | 76.5% | -31%.. 76% |

It is interesting to note that in this case the best model involves the usage of a size measure (CFP) and a complexity density measure (Paths/CFP).

 ISBN: 978-1-61208-165-6 

## III. DISCUSSION

The only models based on a single variable that feature adjusted $R^2$ greater than 0.6 involve either Path or Path density. Anyway, Path seems to be better than both Path per CFP and Path per Functional Processes as far as $R^2$, MMRE and Pred(25) are concerned. The reason why Path appears as a good predictor of effort is probably that this measure summarizes the needed information concerning both size (a la COSMIC) and amount of required elaboration.

Concerning models using two independent variables, we can observe that they appear of similar precision (e.g., MMRE ranges from 17.8% to 23.9%).

As already mentioned, there is no statistically evidence that any of these models features a better fitting that univariate models.

Also in these models, Path per CFP appears as an independent variable in several good models, together with

– the density of data (DataGroups per CFP, DataGroups per FunctionalProcess, PersistentDataGroups per CFP or PersistentDataGroups per FunctionalProcess);
– the number of functional processes.

Interestingly, the persistent data groups (a concept close to unweighted data functions in Function Point Analysis) appear to be a better predictor than the whole number of data groups (i.e., including transient ones).

Path per Functional Process provides –as Path per CFP– good models in combination with the density of data (PersistentDataGroups per FunctionalProcess) or the number of functional processes.

It should be noted that we found some models based *exclusively* on density (such as the second and third in TABLE 2 or the seventh in TABLE 3). These models are rather unexpected, as they say that the size of the programs is not important at all. This result is probably due to the fact that the variation of size was relatively little in the set of projects that we analysed. Additional research is needed to explore this point.

Finally, Path per Functional Process appears also as an argument in models featuring three independent variables. So, the complexity density (i.e., Paths divided by a size measure) appears in *all* the models.

When considering models obtained via OLS regression (i.e., without log-log transformation) we find again an elaboration density measure (Path per CFP), this time in combination with a size measure (CFP or Functional Processes).

## IV. RELATED WORK

A few attempts to account for data elaboration in FSM have been done.

3D Function Points [22] consider three dimensions of the application to be measured: Data, Function, and Control. The Function measurement considers the complexity of algorithms; and the Control portion measures the number of major state transitions within the application.

Bernárdez et al. [2] measured the cyclomatic complexity of a use case in order to validate the use case definition, while Levesque [17] measured the conditions of inputs in a sequential diagram in order to add the concept of complexity to the COSMIC method.

Bashir and Thomson [1] used traditional regression analysis to derive two types of parametric models: a single variable model based on product complexity and a multivariable model based on product complexity and requirements severity. Generally, the models performed well according to a number of accuracy tests. In particular, product complexity explained more than 80% of variation in estimating effort. They concluded that product complexity as an indicator for project size is the dominant parameter in estimating design effort.

Our results are in agreement with those by Bashir and Thomson, in fact several of our models explain 80% (or just slightly less) of the variation of effort.

Hastings and Sajeev [11] proposed a Vector Size Measure (VSM) that incorporates both functionality and problem complexity in a balanced and orthogonal manner. VSM is used as the input to a Vector Prediction Model (VPM) which can be used to estimate development effort early in the software life cycle. The results indicate that the proposed technique allows for estimating the development effort early in the software life cycle with errors not greater than 20% across a range of application types.

Our results are in accordance with the consideration expressed by Morasca on the definition of measures [19] as it appears that the notion of complexity may be represented by taking into account several basic indicators (size, control flow, data, ...) that can be used individually (i.e., without the need to build a derived measure defined as a weighted sum) in estimation models.

Finally, Gencel and Demirors [10] point out that we still need a new Base Functional Component (BFC) Types for the boolean operations of Functional User Requirements, which are often not considered to be algorithmic operations, but which are related to complexity. This point of view highlights the necessity of considering the complexity of elaboration required in FSM, and they suggested introducing as a new BFC type which differs from authors' proposal.

## V. CONCLUSIONS

The work reported here moves from the consideration that development effort depends (also) on the complexity or the amount of computation required, but no suitable measure has emerged as a reliable way for capturing such complexity. In fact, very popular methods like COCOMO II [3][7] still use just an ordinal scale measure for complexity, based on the subjective evaluation performed by the user.

We approached the problem of measuring the required functional complexity by considering (a subset of) the approaches presented in the literature, and testing them on a set of projects that were measured according to the COSMIC FSM.

The results of our analysis do not allow us to draw definite conclusions about the best set of measures to use for effort estimation. However, we observed that all the most significant models obtained were based on a notion of computation density, which is based on the measure of Paths

 ISBN: 978-1-61208-165-6 

[21], i.e., the number of computation flows in functional processes.

Since Paths are quite easy to measure [15] and appear as good effort predictors, we suggest that future research on COSMIC based effort estimation takes into consideration the possibility of involving a Path based measure of functional complexity.

We plan to continue experimenting with measures of functional complexity. Since in this type of experimentations a critical point is the difficulty to get measures, we kindly invite all interested readers that are involved in effort estimations to perform functional complexity measurement and share the data with us and the research community.

## ACKNOWLEDGMENT

The research presented in this paper has been partially funded by the IST project QualiPSo [20], sponsored by the EU in the 6th FP (IST-034763), the project "Metodi e tecniche per l'analisi, l'implementazione e la valutazione di sistemi software" funded by the Università degli Studi dell'Insubria, and the Research Fund of School of Engineering of Austral University.

## REFERENCES

[1] Bashir, H. and Thomson,V. Models for estimating design effort and time. Elsevier. Design Studies, vol.22, n.2, 2001.

[2] Bernárdez B., Durán A., and Genero M. Empirical Evaluation and Review of a Me-trics–Based Approach for Use Case Verification. Journal of Research and Practice in Information Technology, vol. 36 n. 4, 2004.

[3] Boehm, B.W., Horowitz, E., Madachy, R., Reifer, D., Clark, B.K., Steece, B., Winsor Brown, A., Chulani S., and Abts, C. Software Cost Estimation with Cocomo II. Prentice Hall, 2000.

[4] Briand L.C., Morasca S., and Basili V.R. Property-Based Software Engineering Measurement. IEEE Transactions on Software Engineering, Vol. 22, 1996.

[5] Cao, De Tran, Lévesque, G., and Abran, A. From Measurement of Software Functional Size to Measurement of Complexity, ICSM 2002, Montreal, Canada , 2002.

[6] Cao, De Tran, Lévesque, G., and Meunier, J-G. A Field Study of Software Functional Complexity Measurement, 14th International Workshop on Software Measurement, IWSM/METRIKON'04, Berlin, 3-5 November 2004.

[7] COCOMO II Model Definition Manual. http:// csse.usc.edu/csse/research/COCOMOII/ cocomo_downloads.htm

[8] COSMIC – Common Software Measurement International Consortium, 2009. The COSMIC Functional Size Measurement Method - version 3.0.1 Measurement Manual (The COSMIC Implementation Guide for ISO/IEC 19761: 2003), May 2009.

[9] Fenton, N.E. Software Metrics: A Rigorous Approach. Chapman and Hall, London, 1991.

[10] Gencel, C. and Demirors, O. Functional Size Measurement Revisited. ACM Transactions on Software Engineering and Methodology, 17(3), 2008.

[11] Hastings, T. and Sajeev, A. A Vector-Based Approach to Software Size Measurement and Effort Estimation. IEEE Transactions on Software Engineering, vol.27 n.4. 2001.

[12] ISO/IEC19761:2003, Software Engineering – COSMIC-FFP – A Functional Size Measurement Method, ISO.

[13] ISO/IEC 20926: 2003, Software engineering – IFPUG 4.1 Unadjusted functional size measurement method – Counting Practices Manual, International Organization for Standardization, Geneve.

[14] Kitchenham, B., Pfleeger, S.L., McColl, B., and Eagan, S., An empirical study of maintenance and development accuracy, Journal of Systems and Software, vol. 64, 2002.

[15] Lavazza, L. and Robiolo, G. Introducing the Evaluation of Complexity in Functional Size Measurement: a UML-based Approach, 4th International Symposium on Empirical Software Engineering and Measurement - ESEM 2010, Bozen, 16-17 September 2010.

[16] Lavazza, L. and Robiolo, G., The Role of the Measure of Functional Complexity in Effort Estimation, 6th International Conference on Predictive Models in Software Engineering (PROMISE 2010), Timisoara, Romania, 12-13 September 2010.

[17] Levesque G., Bevo V., and Cao, De Tran, Estimating Software size with UML Models. In Proceedings of the 2008 C3S2E conference, ACM International Conference Pro-ceeding Series, vol. 290, 2008.

[18] McCabe, T.J. A complexity measure. IEEE Transactions on Software Engineering, vol.2, n.4, 2005.

[19] Morasca, S. On the use of weighted sums in the definition of measures. ICSE Workshop on Emerging Trends in Software Metrics (WETSoM '10), Cape Town, South Africa, May 04, 2010.

[20] QualiPSo project portal. http://www.qualipso.eu/

[21] Robiolo, G. and Orosco, R. Employing use cases to early estimate effort with simpler metrics. Innovations Syst. Softw. Eng, vol.4, 2008.

[22] Whitmire, A., An Introduction to 3D Function Points, Software Development, vol. 3 n.4, 1995.

 ISBN: 978-1-61208-165-6 