# A Metric of Software Size as a Tool for IT Governance

Marcus Vinícius Borela de Castro, Carlos Alberto Mamede Hernandes
Tribunal de Contas da União (TCU)
Brasília, Brazil
{borela, carlosmh}@tcu.gov.br

*Abstract*— **This paper proposes a new metric for software functional size, which is derived from Function Point Analysis (FPA), but overcomes some of its known deficiencies. The statistical results show that the new metric, Functional Elements (EF), and its submetric, Functional Elements of Transaction (EFt), have higher correlation with the effort in software development than FPA in the context of the analyzed data. The paper illustrates the application of the new metric as a tool to improve IT governance specifically in assessment, monitoring, and giving directions to the software development area.**

***Index Terms*—Function Points, IT governance, IT performance, Software engineering, Software metrics.**

## I. Introduction

ORGANIZATIONS need to leverage their technology to create new opportunities and produce change in their capabilities [1, p. 473]. According to ITGI [2, p. 7], information technology (IT) has become an integral part of business for many companies with key role in supporting and promoting their growth. In this context, IT governance fulfills an important role of directing and boosting IT in order to achieve its goals aligned with the company's strategy.

In order for IT governance to foster the success of IT and of the organization, ISO 38500 [3, p. 7] proposes three main activities: to assess the current and future use of IT; to direct the preparation and implementation of plans and policies to ensure that IT achieves organizational goals; to monitor performance and compliance with those policies (Fig. 1).

A metric of software size can compose several indicators to help reveal the real situation of the systems development area for the senior management of an organization, directly or through IT governance structures (e.g., IT steering committee). Measures such as the production of software in a period (e.g., measure of software size per month) and the productivity of an area (e.g., measure of software size per effort) are examples of indicators that can support the three activities of governance proposed by ISO 38500.

For the formation of these indicators, one can use Function Point Analysis (FPA) *to get* function points (FP) as a metric of software size. Created by Albrecht [4], FPA has become an

This work has been supported by the Brazilian Court of Audit (TCU).

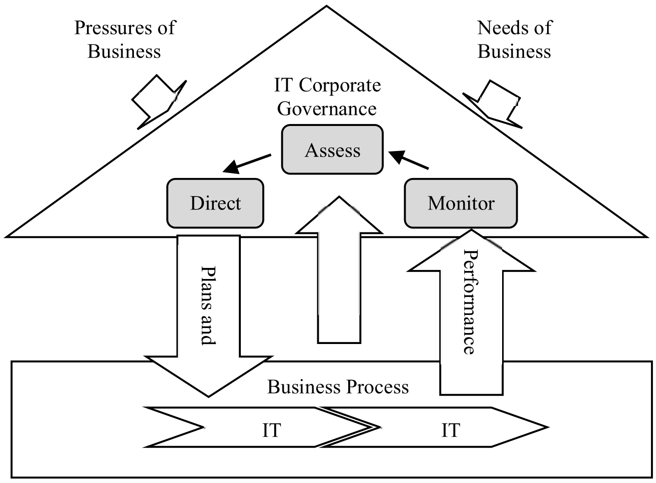


Fig. 1. Cycle Assess-Direct-Monitor of IT Governance
Source:: ISO 38500 [3, p. 7]

international standard for measuring the functional size of a software with the ISO 20926 [5] designation. Its rules are maintained and enhanced by a nonprofit international group of users called International Function Point Users Group (IFPUG), responsible for publishing the Counting Practices Manual (CPM), now in version 4.3.1 [6].

Because it has a direct correlation with the effort expended in software development [7]-[8], FPA has been used as a tool for information technology management, not only in Brazil but worldwide. As identified in the *Quality Research in Brazilian Software Industry report*, 2009 [9, p. 93], FPA is the most widely used metric to evaluate the size of software among software companies in Brazil, used by 34.5% of the companies. According to a survey carried out by Dekkers and Bundschuh [10, p. 393], 80% of projects registered on the International Software Benchmarking Standards Group (ISBSG), release 10, which applied metric used the FPA.

The FPA metric is considered a highly effective instrument to measure contracts [11, p. 191]. However, it has the limitation of not treating non-functional requirements, such as quality criteria and response-time constraints. Brazilian federal government institutions also use FPA for procurement of development and maintenance of systems. The *Brazilian Federal Court of Audit* (TCU) points out FPA as an example of metric to be used in contracts. [2] The *metrics roadmap of SISP* [12], a federal manual for software procurement, recommends its application to federal agencies.

Despite the extensive use of the FPA metric, a large number of criticism about its validity and applicability, described in Section II-B, put in doubt the correctness of its use in contracts and the reliability of its application as a tool for IT management and IT governance.

So the question arises for the research: is it possible to propose a metric for software development, with the acceptance and practicality of FPA, that is, based on its concepts already widely known, without some of the flaws identified in order to maximize its use as a tool for IT governance, focusing on systems development and maintenance?

The specific objectives of this paper are: 1) to present an overview of software metrics and FPA; 2) to present the criticisms to the FPA technique that motivated the proposal of a new metric; 3) to derive a new metric based on FPA; 4)to evaluate the new metric against FPA in the correlation with effort; 5) to illustrate the use of the proposed metric in IT governance in the context of systems development and maintenance.

Following, each objective is covered in a specific section.

## II. DEVELOPMENT

### A. *Software Metrics*

#### 1) *Conceptualization, categorization, and application*

Dekkers and Bundschuh [10, p. 180-181] describe various interpretations for metric, measure, and indicator found in the literature. Concerning this study, no distinction is made among these three terms. We used Fenton and Pfleeger’s definition [13, p. 5] for measure: a number or symbol that characterizes an attribute of a real world entity, object or event, from formally defined rules. Kitchenham *et al.* [14] present a framework for software metrics with concepts related to the formal model in which a metric is based, for example, the type of scale used.

According to Fenton and Pfleeger [13, p. 74], software metrics can be applied to three types of entities: processes, products, and resources. The authors also differentiate direct metrics when only one attribute of an entity is used, from indirect metrics, the other way around [13, p. 39]. Indirect metrics are derived by rules based on other metrics. The speed of delivery of a team (entity type: resource) is an example of indirect metric because it is calculated from the ratio of two measures: size of developed software (product) development and elapsed time (process). The elapsed time is an example of direct metric. Moser [15, p. 32] differentiates size metrics from quality metrics: size metrics distinguish between the smallest and the largest whereas quality metrics distinguish between good and bad. Table I consolidates the mentioned categories of software metrics.

[2] There are several rulings on the subject: 1.782/2007, 1.910/2007, 2.024/2007, 1.125/2009, 1.784/2009, 2.348/2009, 1.274/2010, 1.647/2010, all of the Plenary of the TCU.

Moser [15, p.31] notes that, given the relationship between a product and the process that produced it, a product measure can be assigned to a process, and vice versa. For example, the percentage of effort in testing, which is a development process attribute, can be associated with the generated product as an indicator of its quality. And the number of errors in production in the first three months, a system attribute (product), can be associated to the development process as an indicative of its quality.

Fenton and Pfleeger [13, p. 12] set three goals for software metric: to understand, to control, and to improve the targeted entity. They call our attention to the fact that the definition of the metrics to be used depends on the maturity level of the process being measured: the more mature, more visible, and therefore more measurable [13, p. 83]. Chikofsky and Rubin [16, p. 76] highlight that an initial measurement program for a development and maintenance area should cover five key dimensions that address core attributes for planning, controlling, and improvement of products and processes: size, effort, time, quality, and rework. The authors remind us that what matters are not the metric itself, but the decisions that will be taken from them, refuting the possibility of measuring without foreseeing the goal [16, p. 75].

According to Beyers [17, p. 337], the use of estimates in metric (e.g., size, time, cost, effort, quality, and allocation of people) can help in decision making related to software development and to the planning of software projects.

#### 2) *FPA overview*

According to the categorization of in previous section, FPA is an indirect measure of product size. It measures the functional size of an application (system) as a gauge of the functionality requested and delivered to the user of the software. [3] This is a metric understood by users, regardless of the technology used.

According to Gencel and Demirors [18, p. 4], all functional metrics ISO standards estimate software size based on the functionality delivered to users, [4] differing in the considered objects and how they are measured.

TABLE I
EXAMPLES OF CATEGORIES OF SOFTWARE METRICS

| Criterion | Category | Source |
|---|---|---|
| Entity | Of process<br>Of product<br>Of resource | [13, p. 74] |
| Number of attributes involved | Direct<br>Indirect | [13, p. 39\ |
| Target of differentiation | Size<br>Quality | [15, p. 32] |

[3] The overview presented results from the experience of the author Castro with FPA. In 1993, he coordinated the implementation of FPA in the area of systems development at the Brazilian Superior Labor Court (TST). At TCU, he works with metric, albeit sporadically, without exclusive dedication.

[4] Besides FPA, there are four other functional metrics that are ISO standards, as they meet the requirements defined in the six standards of ISO 14143: MKII FPA, COSMIC-FFP, FISMA, and NESMA. Non-functional attributes of a development process (e.g., development team experience, chosen methodology) are not in the scope of functional metrics. Functional requirements are only one dimension of several impacting the effort. All of them have to be taken into account in estimates. Estimates and non-functional requirements evaluations are not the goal of this paper.

Functionalities can be of two types: transactions, that implement data exchanges with users and other systems, and data files, which indicate the structure of stored data. There are three types of transactions: external inquiry (EQ), external outputs (EO), and external inputs (EI), as the primary intent of the transaction is, respectively, a simple query, a more elaborate query (e.g., with calculated totals) or data update. There are two types of logical data files: internal logical files (ILF) and external interface files (EIF), as their data are, respectively, updated or just referenced (accessed) in the context of the application.

Fig. 2 illustrates graphically these five function types. To facilitate understanding, we can consider an example of EI as an employee inclusion form which includes information in the employees data file (ILF) and validates the tax code (CPF) informed by the user accessing the external file taxpayers (EIF), external to the application. Also in the application we could have, hypothetically, an employee report, a simple query containing the names of the employees of a given organizational unit (EQ) and a more complex report with the number of employees per unit (EO).

In the FPA calculating rule, each function is evaluated for its complexity and takes one of three classifications: low, medium or high complexity. Each level of complexity is associated with a size in function points. Table II illustrates the derivation rule for external inquiries, according to the number of files accessed (File Type Referenced - FTR) and the number of fields that cross the boundary of the application (Data Element Type - DET).

As for EQ, each type of functionality (EO, EI, ILF, and EIF) has its specific rules for derivation of complexity and size, similar to Table II. Table III summarizes the categories of attributes used for calculating function points according to each type of functionality.

The software size is the sum of the sizes of its functionalities. This paper is not an in-depth presentation of concepts associated with FPA. Details can be obtained in the *Counting Practices Manual*, version 4.3.1 [6].

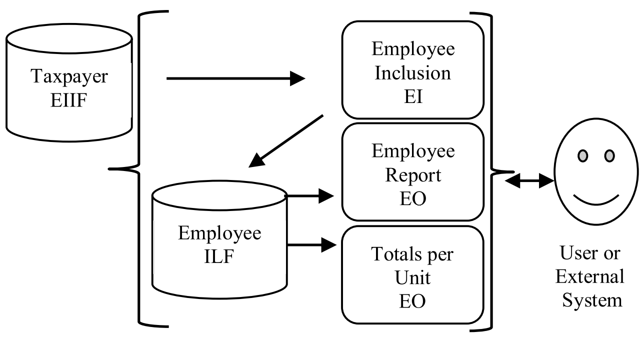


Fig. 2. Visualization of the five types of functions in FPA

TABLE II
DERIVATION RULE FOR COMPLEXITY AND SIZE IN FUNCTION POINTS OF AN EXTERNAL INQUIRY

| FTR (file) \ DET (field) | 1 - 5 | 6 - 19 | 20 or more |
|---|---|---|---|
| 1 | low (3) | low (3) | medium (4) |
| 2 - 3 | low (3) | medium (4) | high (6) |
| 4 or more | medium (4) | high (6) | high (6) |

## *B. Criticisms to the FPA technique that motivated the proposal of a new metric*

Despite the extensive use of the metric FPA, mentioned in Section I, there are a lot of criticism about its validity and applicability that call into question the correctness of its use in contracts and the reliability of its application as a tool for IT management and governance ( [19], [13], [20], [21], [14], [22]; [23], [24], [25]).

Several metrics have been proposed taking FPA as a basis for their derivation, either to adapt it to particular models, or to improve it, fixing some known bugs. To illustrate, there is Antoniol *et al.* [26] work proposing a metric for object-oriented model and Kralj *et al.* [22] work proposing a change in FPA to measure more accurately high complexity functions (item 4 below).

The objective of the metric proposed in this paper is not to solve all faults of FPA, but to help to reduce the following problems related to its definition:

1) low representation: the metric restricts the size of a function to only three possible values, according to its complexity (low, medium, or high). But there is no limit on the number of possible combinations of functional elements considered in calculating the complexity of a function in FPA;

2) functions with different functional complexities have the same size: as a consequence of the low representation. Pfleeger *et al.* [23, p. 36] say that if H is a measure of size, and if A is greater than B, then $H_A$ should be greater than $H_B$. Otherwise, the metric would be invalid, failing to capture in the mathematical world the behavior we perceive in the empirical world. Xia *et al.* [25, p. 3] show examples of functions with different complexities that were improperly assigned the same value in function points because they fall into the same complexity classification, thus exposing the problem of ambiguous classification;

3) abrupt transition between functional element ranges: Xia *et al.* [25, p. 4] introduced this problem. They present two logical files, B and C, with apparent similar complexities, differing only in the number of fields: B has 20 fields and C has 19 fields. The two files are classified as low (7 fp, function points) and medium complexity (10 fp), respectively. The difference lies in the transition of the two ranges in the complexity derivation table: up to 19 fields, it is considered low complexity; from 20 fields, it is considered medium complexity. The addition of only one field leading to an increase in 3 pf is inconsistent, since varying from 1 to 19 fields does not involve any change in the function point size. A similar result occurs in other ranges of transitions;

4) limited sizing of high (and low) complexity functions: FPA sets an upper (and a lower) limit for the size of a function

TABLE III
CATEGORIES OF FUNCTIONAL ATTRIBUTES FOR EACH TYPE OF FUNCTIONALITY

| Function | Functional Attributes |
|---|---|
| Transactions: EQ, EO, EI | referenced files (FTR) and fields (DET) |
| Logical files: ILF, EIF | logical registers (Record Element Type - RET) and fields (DET) |

in 6, 7, 10 or 15, according to its type. Kralj *et al.* [22, p. 83] describe high complexity functions with improper sizes in FPA. They propose a change in the calculation of FPA to support larger sizes for greater complexity;

5) undue operation on ordinal scale: as previously seen, FPA involves classifying the complexity of functions in low, medium or high complexity, as a ordinal scale. These labels in the calculated process are substituted by numbers. An internal logical file, for example, receives 7, 10 or 15 function points, as its complexity is low, medium or high, respectively. Kitchenham [20, p. 29] criticizes the inadequacy of adding up values of ordinal scale in FPA . He argues that it makes no sense to add the complex label with the simple label, even if using 7 as a synonym for simple and 15 as a synonym for complex;

6) inability to measure changes in parts of the function: this characteristic, for example, does not allow to measure function points of part of a functionality that needs to be changed in one maintenance operation. Thus, a function addressed in several iterations in an agile method or other iterative process is always measured with full size, even if the change is considered small in each of them. For example, consider three maintenance requests at different moments for a report already with the maximum size of 7 fp, which initially showed 50 distinct fields. Suppose each request adds a single field. The three requests would be dimensioned with 7 fp each, the same size of the request that created the report, and would total 21 fp. Aware of this limitation, PFA [6, vol. 4, p. 94] points to the Netherlands Software Metrics Association (NESMA) metric as an alternative for measuring maintenance requests. NESMA presents an approach to solve this problem. According to the *Function Point Analysis for Software Enhancement* [27], NESMA measures a maintenance request as the multiplication of the original size of a function by a factor of impact of the change. The impact factor is the ratio of the number of attributes (e.g., fields and files) included, changed or deleted by the original number of attributes of the function. The adjustment factor assumes multiple values of 25%, varying up to a maximum of 150%.

Given the deficiencies reported, the correlation between the size in function points of software and the effort required for the development tends not to be appropriate, since FPA has these deficiencies in the representation of the real functional size of software. If there are inaccuracies in the measuring of the size of what must be done, it is impossible to expect a proper definition of the effort and therefore accuracy in defining the cost of development and maintenance. The mentioned problems motivated the development of this work, in order to propose a quantitative metric, with infinite values, called Functional Elements (EF).

### C. *Derivation process of the new metric*

The proposed metric, Functional Elements, adopts the same concepts of FPA but changes the mechanism to derive the size of functions. The use of concepts widely known to metric specialists will enable acceptance and adoption of the new metric among these professionals.

The reasoning process for deriving the new metric, as described in the following sections, implements linear regression similar to that seen in Fig. 3. The objective is to derive a formula for calculating the number of EF for each type of function (Table VII in Section II-C-4) from the number of functional attributes considered in the derivation of its complexity, as indicated in Table III in Section II-A-2. In this paper, these attributes correspond to the concept of functional elements, which is the name of the metric proposed.

The marked points in Fig. 3 indicate the size in fp (Z axis) of an external inquiry derived from the number of files (X axis) and the number of fields (Y axis), which are the attributes used in the derivation of its complexity (see Table II in Section II-A-2). The grid is the result of a linear regression of these points, and represents the new value of the metric.

#### 1) *Step 1 - definition of the constants*

If the values associated with the two categories of functional attributes are zero, the EF metric assumes the value of a constant. Attributes can be assigned value zero, for example, in the case of maintenance limited to the algorithm of a function not involving changes in the number of fields and files involved.

The values assigned to these constants come from the NESMA functional metric mentioned in Section 2-B. This metric was chosen because it is an ISO standard and supports the maintenance case with zero-value attributes. For each type of functionality, the proposed metric uses the smallest possible value by applying NESMA, that is, 25% of the number of fp of a low complexity function of each type: EIF - 1.25 (25% of 5); ILF - 1.75 (25% of 7); EQ - 0.75 (25% of 3); EI - 0.75 (25% of 3), and EO - 1 (25% of 4).

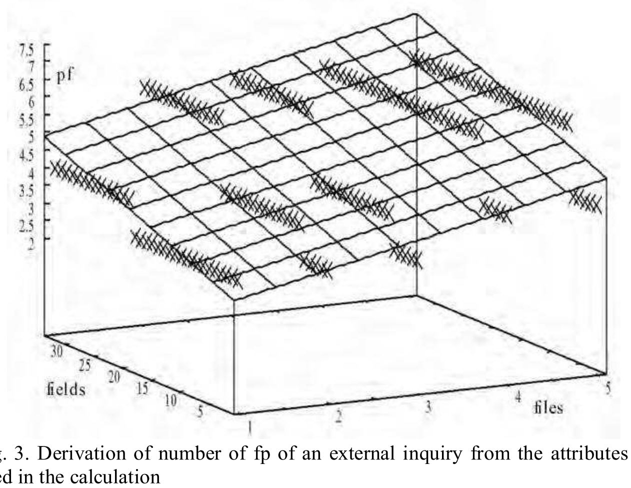


Fig. 3. Derivation of number of fp of an external inquiry from the attributes used in the calculation

*2) Step 2 - treatment of ranges with unlimited number of elements*

In FPA, each type of function has its own table to derive the complexity of a function. Table II in Section II-A-2 presents the values of the ranges of functional attributes for the derivation of the complexity of external inquiries. The third and last range of values of each functional element of all types of functions is unlimited. We see *20 or more* TD in the first cell of the fourth column of the same table, and *4 or more* ALR in the last cell of the first column.

The number of elements in the greater range, that is, the highest value among the first two ranges, was chosen for setting a upper limit for the third range. In the case of ranges for external inquiries, the number of fields was limited to 33, having 14 elements (*20 to 33*) as the second range (*6 to 19*), the largest one. The number of referenced files was limited to 5, using the same reasoning.

The limitation of the ranges is a mathematical artifice to prevent imposing an upper limit for the new metric (4th criticism in Section II-B).

*3) Step 3 - generation of points for regression*

The objective of this phase was to generate, for each type of function, a set of data records with three values: the values of the functional attributes and the derived fp, already decreased from the value of the constant in step 1. Table IV illustrates some points generated for the external inquiry.

An application developed in MS Access generated a dataset with all possible points for the five types of functions, based on the tables of complexity with bounded ranges developed in the previous section. Table V shows all considered combinations of ranges for EQ.

*4) Step 4 - linear regression*

The several points obtained by the procedure described in the previous section were imported into MS Excel for linear regression using the *ordinary least squares method* (OLS). The regression between the size fp, which is the dependent variable, and the functional attributes, which are the dependent variables, held constant with value zero, since these constants were already defined in step 1 and decreased from the expected value in step 3. The statistical results of the regression are shown in Table VI for each type of function.

Table VII shows the derived formula for each type of function with coefficient values rounded to two decimal place values. Each formula calculates the number of functional elements, which is the proposed metric, based on the functional attributes impacting the calculation and the constants indicated in step 1. The acronym EFt and EFd represent the functional elements associated with transactions (EQ, EI, and EO) and data (ILF and EIF), respectively.

The functional elements metric, EF, is the sum of the functional elements transaction, EFT, with the functional

TABLE IV
PARTIAL EXTRACT OF THE DATASET FOR EXTERNAL INQUIRY

| FTR | DET | PF (decreased of constant of step 1) |
|---|---|---|
| 1 | 1 | 2.25 |
| 1 | 2 | 2.25 (...) |
| 1 | 33 | 3.25 |
| 2 | 1 | 2.25 (...) |

TABLE V
COMBINATIONS OF RANGES FOR CALCULATING FP OF EQ

| Function type | Initial FTR | Final FTR | Initial DET | Final DET | Original FP | PF decreased of constant |
|---|---|---|---|---|---|---|
| EQ | 1 | 1 | 1 | 5 | 3 | 2.25 |
| EQ | 1 | 1 | 6 | 19 | 3 | 2.25 |
| EQ | 1 | 1 | 20 | 33 | 4 | 3.25 |
| EQ | 2 | 3 | 1 | 5 | 3 | 2.25 |
| EQ | 2 | 3 | 6 | 19 | 4 | 3.25 |
| EQ | 2 | 3 | 20 | 33 | 6 | 5.25 |
| EQ | 4 | 5 | 1 | 5 | 4 | 3.25 |
| EQ | 4 | 5 | 6 | 19 | 6 | 5.25 |
| EQ | 4 | 5 | 20 | 33 | 6 | 5.25 |

TABLE VI
STATISTICAL REGRESSION - COMPARING RESULTS PER TYPES OF FUNCTIONS

| | ILF | EIF | EO | EI | EQ |
|---|---|---|---|---|---|
| $R^2$ | 0.96363 | 0.96261 | 0.95171 | 0.95664 | 0.96849 |
| Records | 729 | 729 | 198 | 130 | 165 |
| Coefficient *p-value* (FTR or RET) | 3.00E-21 | 1.17E-21 | 7.65E-57 | 1.70E-43 | 4.30E-60 |
| Coefficient *p-value* (DET) | 2.28E-23 | 2.71E-22 | 1.44E-59 | 2.76E-39 | 2.95E-45 |

TABLE VII
CALCULATION FORMULAS OF FUNCTIONAL ELEMENTS BY TYPE OF FUNCTION[5]

| Function type | Formula |
|---|---|
| ILF | $EFd = 1.75 + 0.96 * RET + 0.12 * DET$ |
| EIF | $EFd = 1.25 + 0.65 * RET + 0.08 * DET$ |
| EO | $EFt = 1.00 + 0.81 * FTR + 0.13 * DET$ |
| EI | $EFt = 0.75 + 0.91 * FTR + 0.13 * DET$ |
| EQ | $EFt = 0.75 + 0.76 * FTR + 0.10 * DET$ |

elements of data, EFd, as explained in the formulas of Table VII. So the proposed metric is: EF = EFt + EFd.

The EFt submetric considers logical files (ILF and EIF) as they are referenced in the context of transactions. Files are not counted in separate as in the EFd submetric. Similar to two other ISO standard metrics of functional size [10, p. 388], MKII FPA [28] and COSMIC-FFP [29], EFt does not take into account logical files. EFt is indicated for the cases where the effort of dealing with data structures (EFd) is not subject to evaluation or procurement.

In the next section, the EF and EFt metrics were tested, counting and not counting logical files, respectively. Results show stronger correlation with effort for EFt. Although not evaluated, the EFd submetric has its role as it reflects the structural complexity of the data of an application.

## *D. Evaluation of the new metric*

The new EF metric and its submetric EFt were evaluated for their correlation with effort in comparison to the FPA metric.[6] The goal was not to evaluate the quality of these correlations, but to compare their ability to explain the effort.

We obtained a spreadsheet from a federal government agency with records of Service Orders (OS) contracted with private companies for coding and testing activities. An OS

[5] The size of a request for deleting a function is equal to the constant value, since no specific attributes are impacted by this operation.

[6] Kemerer [8, p. 421] justified linear regression as a means of measuring this correlation.

contained one or more requests for maintenance or development of functions of one system, such as: create a report, change a transaction. The spreadsheet showed for each OS the real allocated effort and, for each request, the size of the function handled. The only fictitious data were the system IDs, functionality IDs and OS IDs, as they were not relevant to the scope of this paper. Each system was implemented in a single platform: Java, DotNet or Natural. The spreadsheet showed the time spent in hours and the number of people allocated for each OS. The OS effort, in man-hours, was derived from the product of time by team size. Table VIII presents the structure of the received data.

Data from 183 Service Orders were obtained. However, 12 were discarded for having dubious information, for example, undefined values for function type, number of fields, and operation type. The remaining 171 service orders were related to 14 systems and involved 505 requests that dealt with 358 different functions. To achieve higher quality in the correlation with effort, we decided to consider only the four systems associated with at least fifteen OS, namely, systems H, B, C, and D. Table IX indicates the number of OS and requests for each system selected.

The data were imported into MS Excel to perform the linear regression using the ordinary least squares method after calculating the size in EF and EFt metrics for each request in an MS-Access application developed by the authors.[7] The regression considered the effort as the independent variable and the size calculated in the PF, EF, and EFT metrics as the dependent ones. As there is no effort if there is no size, the regression considered the constant with value zero, that is, the straight line crosses the origin of the axes. Independent regressions were performed for each system, since the variability of the factors that influence the effort is low within a single system, because the programming language is the same and the technical staff is generally also the same.[8] Fig. 4 illustrates the dispersion of points (OS) on the correlation between size and effort in EFt (man-hour) and the line derived by linear regression in the context of system H.

The coefficient of determination $R^2$ was used to represent the degree of correlation between effort and size calculated for each of the evaluated metrics. According to Sartoris [30, p. 244], $R^2$ indicates, in a linear regression, the percentage of the variation of a dependent variable Y that is explained by the variation of a second independent variable X. Table IX shows the results of the linear regressions performed.

From the results presented on Table IX, comparing the correlation of the metrics with effort, we observed that:

1) correlations of the new metrics (EF, EFt) were considered significant at a confidence level of 95% for all systems (*p-value* less than 0.05).[9] However, the correlation of FPA was not significant for system B (*p-value* 0.088 > 0.05);

2) correlations of the new metrics were higher in both systems with the highest number of OS (H and B). A better result in larger samples is an advantage, because the larger the sample size, the greater the reliability of the results, since the *p-value* has reached the lowest values for these systems;

3) although no metric got a high coefficient of determination ($R^2 > 0.8$), the new metrics achieved medium correlation ($0.8 > R^2 > 0.5$) in the four systems evaluated, whereas FPA obtained weak correlation ($0.2 > R^2$) in system B. We considered the confidence level of 91.2% in this correlation (*p-value* 0.88);

4) the correlation of the new metrics was superior in three out of the four systems (H, B, and D). (A correlation C1 is classified as higher than a correlation C2 if C1 is significant and C2 is not significant or if both correlations are significant and C1 has a higher $R^2$ than C2.)

TABLE VIII
STRUCTURE OF THE RECEIVED DATA TO EVALUATE THE METRIC

| Abbreviation | Description | Domain |
|---|---|---|
| OS | Identification Number of a service order | up to 10 numbers |
| Function | Identification Number of a function | up to 10 numbers |
| Type | Type (categorization) of a functionality according to FPA | ALI, AIE, EE, SE or CE |
| Operation | Operation performed, which may be inclusion (I) of a new feature or change (A) of a function (maintenance) | I or A |
| Final FTR RET | Value at the conclusion of the request implementation: if the function is a transaction, indicates the number of referenced logical files (FTR); if it is a logical file, indicates the number of logical records (RET) | up to 3 numbers |
| Operation FTR RET | Number of FTR or RET that were included, changed or deleted in the scope of a maintenance of a functionality (only in change operation) | up to 3 numbers |
| Original FTR RET | Number of FTR or RET originally found in the functionality (only in change operation) | up to 3 numbers |
| Final DET | Number of DET at the conclusion of the request implementation | up to 3 numbers |
| Operation DET | Number of DET included, changed or deleted in the scope of a functionality maintenance (only in change operation) | up to 3 numbers |
| Original TD | Number of DET originally found in a functionality (only in change operation) | up to 3 numbers |
| FP | Number of function points of the functionality at the conclusion of the request | up to 2 numbers |
| %Impact | Percentage of the original function impacted by the maintenance, as measured by NESMA [27] | 25, 50, 75, 100, 125, 150 |
| PM | Number of maintenance points of the functionality handled, as measured by NESMA [27] | up to 4 numbers |
| System | Identification of a system | one char |
| Hours | Hours dedicated by the team to implement the OS | up to 5 numbers |
| Team | Number of team members responsible for the implementation of the OS | up to 2 numbers |

[7] A logistic nonlinear regression with constant was also performed using Gretl, a free open source tool (http://gretl.sourceforge.net). However, the $R^2$ factor proved that this alternative was worse than the linear regression for all metrics.

[8] The factors that influence the effort and the degree of this correlation are discussed in several articles. We suggest the articles available in the BestWeb database (http://www.simula.no/BESTweb), created as a result of the research of Jorgensen and Shepperd [31].

[9] To be considered a statistically significant correlation at a confidence level of X%, the *p-value* must be less than 1 - X [30, p.11]. For a 95% confidence level, the *p-value* must be less than 0.05.

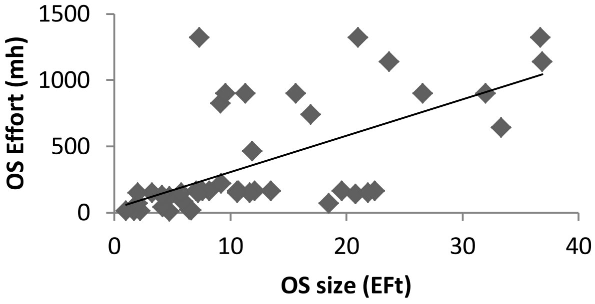


Fig.4. Dispersion of points (OS) of H system: effort (man-hour) x size (Functional Element of Transaction)

TABLE IX
RESULTS OF LINEAR REGRESSIONS - EFFORT *VERSUS* METRICS OF SIZE

| System | | H | B | C | D |
|---|---|---|---|---|---|
| Quantity of OS | | 45 | 25 | 21 | 15 |
| Quantity of Requests | | 245 | 44 | 60 | 20 |
| FP | $R^2$ | 59.3% | 11.2% | 67.7% | 51.8% |
| | *p-value* (test-f) | 4.6E-10 | 8.8E-02 | 3.3E-06 | 1.9E-03 |
| EF | $R^2$ | 65,1% | 60.3% | 53.0% | 54.7% |
| | *p-value* (test-f) | 1.5E-11 | 2.3E-06 | 1.4E-04 | 1.2E-03 |
| | Proportion to FP's $R^2$ | +10% | +438% | -22% | +5% |
| EFt | $R^2$ | 66.1% | 60.3% | 53.0% | 54.7% |
| | *p-value* (test-f) | 8.5E-12 | 2.3E-06 | 1.4E-04 | 1.2E-03 |
| | Proportion to FP's $R^2$ | +11% | +438% | -22% | +5% |

Given the observations listed above, , we conclude for the analyzed data that the metrics proposed, EF and EFt, have better correlation with effort in comparison to FPA. A higher correlation of the EFt metric in comparison to the EF was perceived for system H. Only system H allowed a differentiation of the result for the two metrics by presenting requests for changing logical files in its service orders. Therefore, we see that the EFt submetric tends to yield better correlations if compared to the EF. This result reinforces the hypothesis that the EFd submetric, which composes the EF metric, does not impact the effort, at least not for coding and testing, which are tasks addressed in the evaluated service orders.

Table X contains the explanation of how the proposed metrics, EF and EFt, address the criticisms presented in Section II-B.

### E. Illustration of the use of the new metrics in IT governance

Kaplan and Norton [31, p. 71] claim that what you measure is what you get. According to COBIT 5 [34, p. 13], governance aims to create value by obtaining the benefits through optimized risks and costs. In relation to IT governance, the metrics proposed in this paper not only help to assess the capacity of IT but also enable the optimization of its processes to achieve the results.

Metrics support the communication between the different actors of IT governance (see Fig. 5) by enabling the translation of objectives and results in numbers. The quality of a process can be increased by stipulating objectives and by measuring results through metrics [15, p. 19]. So, the production capacity of the process of information systems development can be enhanced to achieve the strategic objectives with the appropriate use of metrics and estimates.

TABLE X
JUSTIFICATIONS OF HOW THE NEW METRICS ADDRESS THE CRITIQUES PRESENTED IN SECTION II-B

| Critique | Solution |
|---|---|
| Low representation | Each possible combination of the functional attributes considered in deriving the complexity in FPA is associated with a distinct value. |
| Functions with different complexities have the same size | Functionalities with different complexities, as determined by the number of functional attributes, assume a different size. |
| Abrupt transition between functional element ranges | By applying the formulas of calculation described in Section II-C-4, the variation in size is uniform for each variation of the number of functional attributes, according to its coefficients. |
| Limited sizing of high (and low) complexity functions | There is no limit on the size assigned to a function by applying the calculation formulas described in Section II-C-4. |
| Undue operation on ordinal scale | The metrics do not have a ordinal scale with finite values, but rather a quantitative scale with infinite discrete values, which provide greater reliability in operations with values. |
| Inability to measure changes in parts of the function | Enables the measurement of changes in part of a functionality considering in the calculation only the functional attributes impacted by the amendment. |

Software metrics contribute to the three IT governance activities proposed by ISO 38500, mentioned in Section I: to assess, to direct and to monitor. These activities correspond, respectively, to the goals of software metrics mentioned in Section II-A-1: to understand, to improve, and to control the targeted entity of a measurement.

Regarding the directions of IT area, Weill and Ross [36, p. 188] state that the creation of metrics for the formalization of strategic choices is one of four management principles that summarize how IT governance helps companies achieve their strategic objectives. Metrics must capture the progress toward strategic goals and thus indicate if IT governance is working or not [36, p. 188].

Kaplan and Norton [37, pp. 75-76] claim that strategies need to be translated into a set of goals and metrics in order to have everyone's commitment. They claim that the Balanced Scorecard (BSC) is a tool which provides knowledge of long-term strategies at all levels of the organization and also promotes the alignment of department and individual goals with those strategies. According to ITGI [2, p. 29], BSC, besides being a holistic view of business operations, also contributes to connect long-term strategic objectives with short-term actions.

To adapt the concepts of the BSC for the IT function, the perspectives of a BSC were re-established [38, p. 3]. Table XI presents the perspectives of a BSC-IT and their base questions.

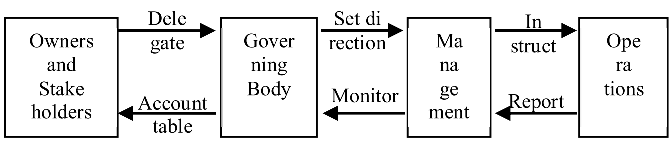


Fig.5. Roles, activities and relationships of IT governance.
Source: ISACA [35, p. 24]

TABLE XI
PERSPECTIVES OF A BSC-IT

| Perspective | Base question | BSC corporative perspective |
|---|---|---|
| Contribution to the business | How do business executives see the IT area? | Financial |
| Customer orientation | How do customers see the IT area? | Customer |
| Operational excellence | How effective and efficient are the IT processes? | Internal Processes |
| Future orientation | How IT is prepared for future needs? | Learning |

Source: inspired in ITGI [2, p. 31]

According to ITGI [2, p. 30], BSC-IT effectively helps the governing body to achieve alignment between IT and the business. This is one of the best practices for measuring performance [2, p. 46]. BSC-IT is a tool that organizes information for the governance committee, creates consensus among the stakeholders about the strategic objectives of IT, demonstrates the effectiveness and the value added by IT and communicates information about capacity, performance and risks [2, p. 30].

Van Grembergen [39, p.2] states that the relationship between IT and the business can be more explicitly expressed through a cascade of scorecards. Van Grembergen [39, p.2] divides BSC-IT into two: BSC-IT-Development and BSC-IT-Operations. Rohm and Malinoski [40], members of the Balanced Scorecard Institute, present a process with nine steps to build and implement strategies based on scorecard. Bostelman and Becker [41] present a method to derive objectives and metrics from the combination of BSC and the Goal Question Metric (GQM) technique proposed by Basili and Weiss [42]. The association with GQM method is consistent to what ISACA [43, p. 74] says: good strategies start with the right questions. The metric proposed in this paper can compose several indicators that can be used in BSC-IT - Development.

Regarding the activities of IT monitoring and assessment [3, p. 7], metrics enable the monitoring of the improvement rate of organizations toward a mature and improved process [1, p. 473]. Performance measurement, which is object of monitoring and assessment, is one of the five focus areas of IT governance, and it is classified as a driver to achieve the results [2, p. 19].

To complement the illustration of the applicability of the new metric for IT governance, Table XII shows some indicators based on EF. [10] The same indicator can be used on different perspectives of a BSC-IT-Development, depending on the targeted entity and the objective of the measurement, such as the following examples. The productivity of a resource (e.g., staff, technology) may be associated with the *Future Orientation* perspective, as it seeks to answer whether IT is prepared for future needs. The same indicator, if associated with an internal process, encoding, for example, reflects a vision of its production capacity, in the *Operational Excellence* perspective. In the *Customer Orientation* perspective, production can be divided by client, showing the proportion of IT production to each business area. The evaluation of the variation in IT production in contrast to the production of business would be an example of using the indicator in the *Contribution to the Business* perspective.

The choice of indicators aimed to encompass the five fundamental dimensions mentioned in Section II-A-1: size, effort, time, quality, and rework. A sixth dimension was added: the expected benefit. According to Rubin [44, p. 1], every investment in IT, from a simple training to the creation of a corporate system, should be aligned to a priority of the business whose success must be measured in terms of a specific value. Investigating the concepts and processes associated with the determination of the value of a function (or a system or the IT area) is not part of the scope of this work. This is a complex and still immature subject. The dimension of each indicator is shown in the third column of Table XII.

Some measurements were normalized by being divided by the number of functional elements of the product or process, tactics used to allow comparison across projects and systems of different sizes. The ability to standardize comparisons, as in a BSC, is one of the key features of software metrics [45, p. 493]. It is similar to normalize construction metrics based on square meter, a common practice [46, p. 161].

As Dennis argues [47, p. 302], one should not make decisions based on a single indicator, but from a vision formed by several complementary indicators. As IT has assumed greater prominence as a facilitator to the achievement of business strategy, the use of dashboards to monitor its performance, under appropriate criteria, has become popular among company managers [43, p. 74]. Abreu and Fernandes [48, p. 167] propose some topics that may compose such strategic and tactical control panels of IT.

Fig. 6 illustrates the behavior of the indicators shown in Table XII with annual verification for hypothetical systems A,

TABLE XII
DESCRIPTION OF ILLUSTRATIVE INDICATORS

| Metric | Unit | Dimension | Description of the calculation for a system |
|---|---|---|---|
| Functional size | EF | Size | sum of the functional size of the functionalities that compose the system at the end of the period |
| Production in the period | EF | Effort | sum of the functional size of requests for inclusion, deletion, and change implemented in the period |
| Production on rework | EF | Rework | sum of the functional size of requests for deletion and change implemented in the period |
| Productivity | Functional Elements / Man–hour | Effort | sum of the functional size of requests implemented in the period / sum of the efforts of all persons allocated to the system activities in the period |
| Error density | Failures / Functional Element | Quality | number of failures resulting from the use of the system in a period / size of the system at the end of the period |
| Delivery speed | Functional Elements / Hour | Time | sum of the size of the features implemented in the period / elapsed time |
| Density of the expected benefit | $ / EF | Expected benefit | benefit expected by the system in the period / system size |

[10] The illustration is not restricted to EF, as the indicators could use others software size metrics.

B, C, and D.[11] The vertical solid line indicates how the indicator to the system was in the previous period, allowing a view of the proportion of the increasing or decreasing of the values over the period. In the productivity column (column 4), a short line at its base indicates, for example, a pattern value obtained by benchmark. The vertical dashed line metric associated with the production in the period (2) indicates the target set in the period for each system: system A reached it, system D exceeded it, and systems B and C failed.

In one illustrative and superficial analysis of the indicators for system C, one can associate the cause of not achieving the production goal during that period (2) with the decrease of the delivery speed (6) and the increase of the production on rework (3), resulted, most likely, from the growth in the error density (5). The reduction on the delivery speed (6) which can be associated with decreased productivity (4) led to a low growth of the functional size of the system (1) during that period. These negative results led to a decrease in the density of the expected benefit (7).

Fig. 6 represents an option of visualization of the governance indicators shown in Table XII: a multi-metrics chart of multi-instances of a targeted entity or a targeted attribute. The vertical column width is variable depending on the values of the indicators (horizontal axis) associated with the different instances of entities or attributes of interest (vertical axis). The same vertical space is allocated for each entity instance. The width of the colored area, which is traced from the left to the right, indicates graphically the value of the indicator for the instance.

In the hands of the governance committee, correct indicators can help senior management, directly or through any governance structure, to identify how IT management is behaving and to identify problems and the appropriate course of action when necessary.

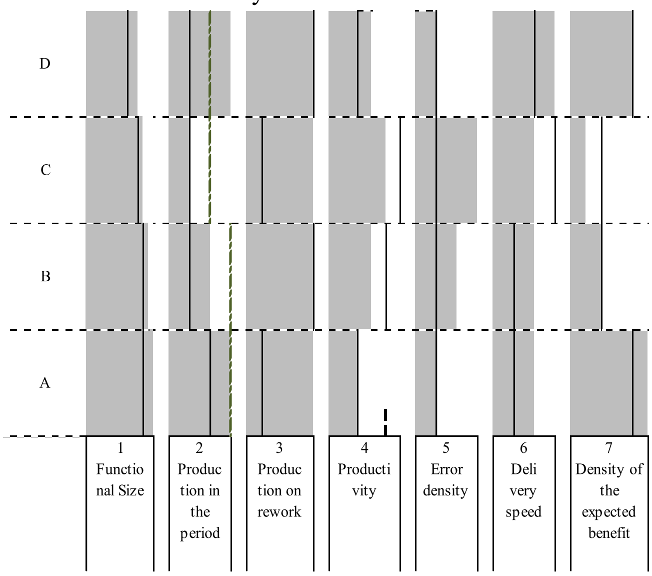


Fig. 6. Annual indicators of systems A, B, C and D

[11] The fictitious values associated with the indicators were adjusted so that all vertical columns had the same maximum width. The adjustment was done by correlating the maximum value for the indicator with the width defined for the column. The other values were derived by a simple rule of three.

## III. Final Considerations

The five specific objectives proposed for this work in Section I were achieved, albeit with limitations and with possibilities for improvement that are translated into proposals for future work.

The main result was the proposition of a new metric EF and its submetric EFt. The new metrics, free of some deficiencies of the FPA technique taken as a basis for their derivation, reached a higher correlation with effort than the FPA metric, in the context of the analyzed data.

The paper also illustrated the connection found between metrics and IT governance activities, either in assessment and monitoring, through use in dashboards, or in giving direction, through use in BSC-IT.

There are possibilities for future work in relation to each of the five specific objectives.

Regarding the conceptualization and the categorization of software metrics, a comprehensive literature research is necessary to the construction of a wider and updated categorization of software metrics.

Regarding the presentation of the criticisms to FPA, only the criticisms addressed by the new proposed metrics were presented. Research in the theme, as a bibliographic research to catalog the criticisms, would serve to encourage other propositions of software metrics.

Regarding the process of creating the new metric, it could be improved or it could be applied to other metrics of any area of knowledge based on ordinal values derived from tables of complexity as FPA (e.g., metric proposed by KARNER [49]: Use Case Points). Future works may also propose and evaluate changes in the rules and in the scope of the new metrics.

Regarding the evaluation of the new metric, the limitation in using data from only one organization could be overcome in new works. Practical applications of the metric could also be illustrated. New works could compare the results of EF with the EFt submetric as well as compare both with other software metrics. Different statistical models could be used to evaluate its correlation with effort even in specific contexts (e.g., development, maintenance, development platforms). We expect to achieve a higher correlation of the new metric with effort in agile methods regarding to the APF, considering its capacity of partial functionality sizing. (6th criticism in Section II-B.)

Regarding to the connection with IT governance, a work about the use of metrics in all IT governance activities is promising. The proposed graph for visualization of multiple indicators of multiple instances through columns with varying widths along their length can also be standardized and improved in future work.[12]

A suggestion for future work is noteworthy: the definition

[12] In http://learnr.wordpress.com (accessed on 04 November 2012) is located a graph that functionally reminds the proposed one: heatmap plotting. However it is different in the format and in the possibilities of evolution. As we did not find any similar graph, we presume to be a new format for viewing the behavior of Multiple Indicators about Multiple Instances through Columns with Varying Widths along their Extension (MIMICoVaWE). An example of evolution would be a variation in the color tone of a cell according to a specific criterion (eg in relation to achieving of a specified goal).

of an indicator that shows the level of maturity of a company regarding to the use of metrics in IT governance. Among other aspects, it could consider: the breadth of the entities evaluated (e.g., systems, projects, processes, teams), the dimensions treated (e.g., size, rework, quality, effectiveness) and the effective use of the indicators (e.g., monitoring, assessment).

Finally, we expect that the new metric EF and its submetric EFt help increase the contribution of IT to the business, in an objective, reliable, and visible way.

## REFERENCES


[1] H. A. Rubin, ―Software process maturity: measuring its impact on productivity and quality,” in Proc. of the 15th int. conf. on Softw. Eng., IEEE Computer Society Press, pp. 468-476, 1993.

[2] ITGI - IT Governance Institute. *Board briefing on IT Governance,* 2nd ed, Isaca, 2007.

[3] ISO/IEC, *38500: Corporate governance of information technology,* 2008.

[4] A. J. Albrecht, "Measuring application development productivity" in *Guide/Share Application Develop. Symp. Proc.,* pp.83-92 1979.

[5] ISO/IEC, *20926: Software measurement - IFPUG functional size measurement method,* 2009.

[6] IFPUG - International Function Point Users Group, *Counting Practices Manual*, Version 4.3.1, IFPUG, 2010.

[7] A. Albrecht and J. Gaffney Jr., ―Software function, source lines of code, and development effort prediction: A software science validation,” *IEEE Trans. Software Eng.*,vol. 9, pp. 639-648, 1983.

[8] C. F. Kemerer, ―An empirical validation of software cost estimation models,” *Communications of the ACM,* vol. 30, no. 5, pp. 416-429, 1987.

[9] Brazil. MCT - Ministério da Ciência e Tecnologia. ―Quality Research in Brazilian Software Industry; Pesquisa de Qualidade no Setor de Software Brasileiro – 2009,” Brasília. [Online]. 204p. Available: http://www.mct.gov.br/upd_blob/0214/214567.pdf

[10] M. Bundschuh and C. Dekkers, *The IT measurement compendium: estimating and benchmarking success with functional size measurement,* Springer, 2008.

[11] C. E. Vazquez, G. S. Simões and R. M. Albert, *Function Point Analysis: Measurement, Estimates and Project Management Software; Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software*. Editora Érica, São Paulo, 2005.

[12] Brazil. SISP - Sistema de Administração dos Recursos de Tecnologia da Informação. (2012). ―Metrics Roadmap of SISP - Version 2.0; Roteiro de Métricas de Software do SISP – Versão 2.0,” Brasília: Ministério do Planejamento, Orçamento e Gestão.Secretaria de Logística e Tecnologia da Informação. [Online]. Available: http://www.sisp.gov.br/ct-gcie/download/file/Roteiro_de_Metricas_de_Software_do_SISP_-_v2.0.pdf

[13] N. E. Fenton and S. L. Pfleeger, *Software metrics: a rigorous and practical approach*, PWS Publishing Co, 1998.

[14] B. Kitchenham, S. L. Pfleeger and N. Fenton, ―Towards a framework for software measurement validation,” *IEEE Trans. Softw. Eng.*, vol. 21, no. 12, pp. 929-944, 1995.

[15] S. MOSER, ―Measurement and estimation of software and software processes,” Ph.D. dissertation, University of Berne, Switzerland, 1996.

[16] E. Chikofsky and H. A. Rubin, ―Using metrics to justify investment in IT,” *IT professional*, vol. 1, no. 2, pp. 75-77. 1999.

[17] C. P. Beyers, "Estimating software development projects." in *IT measuremen,*. Addison-Wesley Longman Publishing Co., Inc., pp. 337-362, 2002.

[18] C. Gencel and O. Demirors, ―Functional size measurement revisited,” *ACM Transactions on Software Engineering and Methodology (TOSEM)* vol. 17, no. 3, p. 15, 2008.

[19] A. Abran and P. N. Robillard, "Function Points: A Study of Their Measurement Processes and Scale Transformations", *Journal of Systems and Software*, vol. 25, pp.171 -184 1994.

[20] B. Kitchenham, ―The problem with function points,” *IEEE Software,* vol. 14, no. 2, pp. 29-31, 1997.

[21] B. Kitchenham, K. Känsälä, ―Inter-item correlations among function points,” in Proc.15th Int. Conf. on Softw. Eng., , IEEE Computer Society Press, pp. 477-480, 1993.

[22] T. Kralj, I. Rozman, M. Heričko and A.Živkovič, ―Improved standard FPA method—resolving problems with upper boundaries in the rating complexity process,” *Journal of Systems and Software*, vol. 77, no. 2, pp. 81-90, 2005.

[23] S. L. Pfleeger, R. Jeffery, B. Curtis and B. Kitchenham, ―Status report on software measurement,” *IEEE Software*, vol. 14, no. 2, pp. 33-43, 1997.

[24] O. Turetken, O. Demirors, C. Gencel, O. O. Top, and B. Ozkan, ―The Effect of Entity Generalization on Software Functional Sizing: A Case Study,” in *Product-Focused Software Process Improvement*, Springer Berlin Heidelberg, pp. 105-116, 2008.

[25] W. Xia, D. Ho, L. F. Capretz, and F. Ahmed, ―Updating weight values for function point counting,” *International Journal of Hybrid Intelligent Systems*, vol. 6, no. 1, pp. 1-14, 2009.

[26] G. Antoniol, R. Fiutem and C. Lokan, "Object-Oriented Function Points: An Empirical Validation," *Empirical Software Engineering*, vol. 8, no. 3, pp. 225-254, 2003

[27] NESMA - Netherlands Software Metrics Association, ―Function Point Analysis For Software Enhancement,” [Online], Available: http://www.nesma.nl/download/boeken_NESMA/N13_FPA_for_Software_Enhancement_(v2.2.1).pdf

[28] ISO/IEC, *20968: MkII Function Point Analysis - Counting Practices Manual*, 2002.

[29] ISO/IEC, *19761: COSMIC: a functional size measurement method,* 2011.

[30] A. Sartoris, *Estatística e introdução à econometria; Introduction to Statistics and Econometrics*, Saraiva S/A Livreiros Editores, 2008.

[31] M. Jorgensen and M. Shepperd, ―A systematic review of software development cost estimation studies,” *IEEE Trans. Softw. Eng., vol.* 33, no. 1, pp. 33-53, 2007.

[32] M. L. Orlov, ―Multiple Linear Regression Analysis Using Microsoft Excel,” Chemistry Department, Oregon State University, 1996.

[33] R. S. Kaplan and D. P. Norton, ―The balanced scorecard - measures that drive performance,” Harvard business review, vol. 70, no. 1, pp. 71-79, 1992.

[34] Isaca, *COBIT 5: Enabling Processes*, Isaca, 2012.

[35] Isaca, *COBIT 5: A Business Framework for the Governance and Management of IT*, Isaca, 2012.

[36] P. Weill and J. W. Ross, *IT governance: How top performers manage IT decision rights for superior results*, Harvard Business Press, 2004.

[37] R. S. Kaplan and D. P. Norton, ―Using the balanced scorecard as a strategic management system,” *Harvard business review,* vol.74, no. 1, pp. 75-85, 1996.

[38] W. Van Grembergen and R. Van Bruggen, ―Measuring and improving corporate information technology through the balanced scorecard,” *The Electronic Journal of Information Systems Evaluation*, vol. 1, no. 1. 1997.

[39] W. Van Grembergen, ―The balanced scorecard and IT governance,” *Information Systems Control Journal*, Vol 2, pp.40-43, 2000.

[40] H. Rohm and M. Malinoski, ―Strategy-Based Balanced Scorecards for Technology,” Balanced Scorecard Institute, 2010.

[41] S. A. Becker and M. L. Bostelman, "Aligning strategic and project measurement systems," *IEEE Software*, vol. 16, no. 3, pp. 46-51, May/Jun 1999.

[42] V. R. Basili, and D. M. Weiss, "A Methodology for Collecting Valid Software Engineering Data," *IEEE Trans. Softw. Eng.*, vol. SE-10, no. 6, pp. 728-738, Nov. 1984.

[43] Isaca,. *CGEIT Review Manual 2010*, ISACA.

[44] H. A. Rubin, ―How to Measure IT Value,” CIO insight, 2003.

[45] B. Hufschmidt, ―Software balanced scorecards: the icing on the cake,” in *IT measurement*, Addison-Wesley Longman Publishing Co., Inc., pp. 491-502. 2002.

[46] C. A. Dekkers, ―How and when can functional size fit with a measurement program?," in *IT measurement,* Addison-Wesley Longman Publishing Co., Inc., pp. 161-170, 2002.

[47] S. P. Dennis, ―Avoiding obstacles and common pitfalls in the building of an effective metrics program,” in *IT measurement*, Addison-Wesley Longman Publishing Co., Inc., pp. 295-304, 2002.

[48] A. A. Fernandes and V. F. Abreu, *Deploying IT governance: from strategy to process and services management; Implantando a governança de TI: da estratégia à gestão de processos e serviços*, Brasport, 2009.

[49] G. Karner. ―Metrics for Objectory,” Diploma thesis, University of Link ping, Sweden, No. LiTH-IDA-Ex9344:21, December 1993.