This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Improvements to the Function Point Analysis Method: A Systematic Literature Review

Marcos de Freitas Junior, Marcelo Fantinato, and Violeta Sun

***Abstract*—Function point analysis (FPA) is a standardized method to systematically measure the functional size of software. This method is proposed by an international organization and it is currently recommended by governments and organizations as a standard method to be adopted for this type of measurement. This paper presents a compilation of improvements, focused on increasing the accuracy of the FPA method, which have been proposed over the past 13 years. The methodology used was a systematic literature review (SLR), which was conducted with four research questions aligned with the objectives of this study. As a result of the SLR, of the 1600 results returned by the search engines, 454 primary studies were preselected according to the criteria established for the SLR. Among these studies, only 18 specifically referred to accuracy improvements for FPA, which was the goal of this study. The low number of studies that propose FPA improvements might demonstrate the maturity of the method in the current scenario of software metrics. Specifically in terms of found issues, it was found that the step for calculating the functional size exhibited the highest number of problems, indicating the need to revise FPA in order to encompass the possible improvements suggested by the researchers.**

***Index Terms*—Accuracy improvement, function point analysis (FPA), systematic literature review (SLR).**

I. Introduction

THE systematic measurement of different aspects of the product being developed is a constant concern in various areas of engineering [1], including software engineering [2]. A key aspect to be measured in software engineering projects [3] is the functional size of the software, for which one of the most used methods is function point analysis (FPA) [4].

FPA is a standardized method aimed at establishing a software size measurement from its functional requirements, considering the features to be implemented in it. It was designed to be applied regardless of the programming language and implementation technology. FPA was proposed by Albrecht [4] as an outcome from a project at IBM between 1974 and 1978. The method was expanded and published in 1984 in the IBM internal report "IBM CIS & A Guideline 313, AD/M Productivity Measurement and Estimate Validation."

In 1986, the International Function Point User Group (IFPUG) [5] was created as a nonprofit organization to promote and disseminate the effective management of the software development and maintenance through FPA. The IFPUG is currently the FPA regulator agency, responsible for the improvement and development of the rules set out in the Counting Practices Manual (CPM) [6], currently in version 4.3.1. Since the creation of the IFPUG, the original FPA method is known as the IFPUG's FPA, referred in this paper only as FPA.

FPA is currently standardized by ISO/IEC 20926:2010. This standard specifies the set of definitions, rules and steps for applying it. There are similar methods, also standardized by other ISO/IEC standards, derived from the original FPA, such as COSMIC, FiSMA, Mark-II, and NESMA, which are not in the scope of this study presented in this paper.

FPA is widely used as a reference to derive other measures such as development effort, productivity, or cost. The key relevance of FPA can be seen in cases such as in Brazil, where the Federal Government determined that all software development procurement for the government must comply with an objective metric [7]. Moreover, the government recommends that FPA be applied to members of SISP (Management System for Information and Computer Resources) and that its script for software metrics be adopted when hiring services of development and maintenance of software [8].

However, there are some controversies regarding the FPA method, when evaluated by different researchers, in terms of advantages and limitations. Kampstra and Verhoef [9] and Kemerer [10], for example, report that the FPA method does not produce consistent results when applied by different metrics. Meli [11], however, points to a mismatch between the complexities established for the base functional components (BFC) and the possible productivity estimates. However, such findings did not limit the application of FPA in several organizations interested in the method in order to obtain cost and effort estimates from the functional size to conduct productivity studies, among other analysis.

In order to contribute to this area, a systematic literature review (SLR) was conducted to identify evidence of the method's potential limitations as well as the solutions proposed. The SLR refers to a compilation of improvements, focused on increasing the accuracy of FPA, which have been proposed over the past 13 years. Therefore, the main objective of this paper is to present the results of the study so that other researchers interested in this area may have an overview of the limitations and improvements of FPA presented in the literature. According to our investigations, no previous study has been ever conducted specifically with this purpose. Two related SLR has been published [12], [13], but they are related to wider software metrics context in general. These SLR include analyses related to FPA but without focusing on its limitations and improvements, as done here.

Manuscript received September 24, 2014; revised April 13, 2015 and June 21, 2015; accepted June 26, 2015. Review of this manuscript was arranged by Department Editor T. R. Browning.

The authors are with the School of Sciences, Arts and Humanities, University of São Paulo, Sao Paulo 03828-000, Brazil (e-mail: marcos.freitas@usp.br; m.fantinato@usp.br; violeta@usp.br).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEM.2015.2453354

The following sections are presented: the FPA procedure; the SLR methodology; the analysis results; a discussion of the results achieved; and the conclusions of this paper.

II. FPA Method Procedure

The procedure to apply FPA, according to the IFPUG [6], includes the following steps.

1) *Define the counting boundary, scope, and purpose:* A software boundary is defined by establishing a logical border between the software being measured, its users and other software systems [6]. It depends only on the user's external business view; hence, it is determined based on it. The boundary can be subjective, consequently, it is often difficult to delineate where one software ends and another begins. Therefore, instead of considering technical aspects, the boundary should consider a business perspective [14].
2) *Measure data functions:* A data function addresses user's functional requirements related to storing or referencing data. It can be classified into two different types of BFC: internal logical file (ILF) and external interface file (EIF) [6]. The steps to identify the existing data functions on the software boundary are: a) group data tables from the logical side; b) classify them as ILF or EIF; c) determine the number of data element types (DET) and record element types (RET) of each ILF and each EIF; and d) determine the complexity (high, medium, and low) and the functional size of the data function of each BFC [6].
3) *Measure transactional functions:* A transactional function is the functionality provided by software to the user for data processing. It can be classified into three different types of BFC: external input (EI), external output (EO), and external inquiry (EQ) [6]. The steps to measure a transactional function are: a) decompose the requirements into the smallest units of activity; b) classify each transactional function in EI, EO, or EQ; c) determine the number of DET and of file type reference (FTR) of each EI, EO, and EQ; and d) determine the complexity (high, medium, and low) and the functional size of the transactional function of each BFC [6].
4) *Calculate the functional size:* Different formulas can be used to calculate the functional size considering the project type, considering the scope and purpose of the counting: development, improvement, or application [6].
5) *Document the counting and report the results:* Recording the premises and interpretations during the counting ensures the tracking of the functional size resulting from a counting using FPA. This makes the method more consistent and indicates the possible improvements that can be done in the documentation provided for the counting [6].

The complexity of the data and transactional functions is given by the matching Tables I–III after determining the number of DET, RET, and FTR, with different weights. The functional size of the data and transactional functions is given by the matching Table IV.

During Step 4 (calculate function size), after the first calculation of the functional size using some of the provided formula, which is called the nonadjusted functional size, it is possible—optionally—to adjust the measured functional size by applying the value adjustment factor (VAF). The VAF indicates the general functionality provided to the user of the application. The VAF is calculated based on an assessment of 14 general system characteristics (GSC) for software [6]. The calculation of VAF is as follows:

$$\text{VAF} = (\text{ TDI }^* \ 0.01) + 0.65 \quad (1)$$

where TDI is the total of degree of influence defined for each nonfunctional requirement. Degree of influence is a numerical indicator of the amount of impact of each GSC, from "0" to "5." These indicators are used to compute the VAF.

There are 14 GSCs [6]: 1) data communications; 2) distributed data processing; 3) performance; 4) heavily used configuration; 5) transaction rate; 6) online data entry; 7) end-user efficiency; 8) online update; 9) complex processing; 10) reusability; 11) installation ease; 12) operational ease; 13) multiple sites; and 14) facilitate change.

TABLE I
Data Function Complexity

		DET		
		1–19	20–50	>50
Record	1	Low	Low	Average
Element	2-5	Low	Average	High
Types (RET)	>5	Average	High	High

TABLE II
EI Functional Complexity

		DET		
		1–5	6–15	>15
File Type	0–1	Low	Low	Average
Reference	2	Low	Average	High
(FTR)	>2	Average	High	High

TABLE III
EO and EQ Functional Complexity

		DET		
		1–5	6–19	>19
File Type	0–1	Low	Low	Average
Reference	2–3	Low	Average	High
(FTR)	>3	Average	High	High

TABLE IV
Function Type Weight According to Complexity

Function Type	Low	Medium	High
ILF	7	10	15
EIF	5	7	10
EI	3	4	6
EO	4	5	7
EQ	3	4	6

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

III. Methodology

An SLR is a way to identify, evaluate, and interpret all available research—through primary studies—relevant to a specific research question, a thematic area, or a particular phenomenon of interest [15]. An SLR is a type of secondary study. Secondary studies rely on studies—observational or experimental—which relate directly to the research question to be answered [16].

An SLR differs from traditional reviews (such as simple literature reviews) and surveys with comments made by domain experts because it is replicable, scientific, and used as a transparent approach to avoid possible bias [17]. This allows the researchers to conduct a critical analysis of the collected data to solve problems identified in the literature and identify potential problems that can be studied in the future [17].

By using the SLR, the research, selection, analysis and the organization of studies are facilitated due to predefined steps and criteria [15]. Thus, the application of the review is systematized, enabling to find relevant content about a topic of interest in a repository with a great amount of information.

The SLR start is to formulate the research questions, which need be answered based on data gathering and data analysis [16]. This SLR was conducted based on the procedures proposed by Kitchenham [15]. According to these procedures, the SLR can be grouped into three main steps, namely: 1) plan the SLR; 2) conduct the SLR; and 3) report the SLR. These procedures are detailed in the following sections.

A. Identify the Need for an SLR

Three main points justify the relevance to conduct this SLR, as described as follows:

1) *Lack of SLR specifically focused on FPA improvements:* Given that there is no SLR aimed at investigating the limitations and improvement suggestions related to FPA, this SLR would compile and analyze existing knowledge. Thus, the interested researchers could use it as a knowledge base for future research in the area.
2) *Support related areas:* This SLR can support other areas besides the FPA area itself. For example, software estimation could benefit from such a study, since many methods used to derive the development effort are based on functional software size calculated by FPA. Thus, the observation of the main improvements proposed for FPA would allow identifying which ones could increase the accuracy of the functional sizes found. Consequently, the input parameters of many software estimation methods based on FPA could also be improved. These techniques receive, as an input parameter, the software functional size (in function points) and generate, as an output parameter, the estimated development effort.
3) *Advance on the improvement of FPA tools:* FPA professionals proposed tools that automate the calculation of FPA or store the result of this calculation. The study of the proposed FPA improvements could enable these tools to incorporate such improvements, to make the functional size calculation, generated by automated tools, more precise.

TABLE V
Research Questions

RQ	Research Question	Motivation
1	Are there FPA problems being reported in terms of accuracy?	Identify and summarize potential FPA problems, pointed out by researchers.
2	Are there proposals to improve the FPA's accuracy? Which improvements are being proposed?	Identify and summarize the approaches proposed to improve the FPA accuracy.
3	Are the proposed improvements effective?	Verify if the proposed improvements are more effective when compared to the standard FPA.
4	What are the limitations for the proposed improvements?	Identify potential problems or weaknesses for the proposed improvements.

B. Research Questions

Considering the relevance presented in the previous section, Table V describes four research questions to guide this SLR. The research questions, as the main objective of this SLR, were defined considering the need to identify the proposed improvements to FPA, in order to identify possible trends for functional software measurement that use this method.

Thus, this study sought to "identify the major FPA issues reported by researchers in terms of accuracy" (RQ1), to identify "what improvements are being proposed for solving these problems" (RQ2). Moreover, it also sought to determine whether "the proposed improvements are effective" (RQ3) and whether there are "potential disadvantages presented within the context of software functional measurement" (RQ4).

Research questions were defined based on [15]: 1) a specific intervention, i.e., a software technology aimed at specific problems; b) a population, i.e., what is affected by the intervention; and 3) the results, which are the research implications related to the practitioners' important factors. Therefore, specifically for this SLR, the research questions were defined based on: 1) *intervention*—FPA method; 2) *population*—studies defining approaches to improve results generated with FPA; and 3) *results*—comprehensive overview of the proposed approaches to improve results generated with FPA.

C. Selection of Data Sources and Studies

The search for primary studies, conducted from October to December 2014 and renewed in April 2015, was performed using four bibliographic databases: ACM Digital Library (DL), IEEEXplore, Scopus, and Science Direct. They were chosen because they have at least one online search engine with the following options: 1) advanced search by keywords and 2) filtering of results by publication year and by area or type of publication. Table VI shows the specific search strings used in each of the four basic databases.

Two iterations were carried out. In the former, the results closest to the objective of this SLR were selected. This required a careful reading of the paper titles. In cases of doubts related to the applicability of the papers for this SLR, the abstracts were also read. In this first iteration, 454 items were selected:

TABLE VI
DATABASES, SEARCH STRINGS, AND RESULTS

DB	Search String	Total	1st[2]	2nd
Scopus[1]	(TITLE-ABS-KEY("function point" OR "function points" OR "function-point" OR "function-points")) AND PUBYEAR > 2001 AND (LIMIT-TO(DOCTYPE, "cp") OR LIMIT-TO(DOCTYPE, "ar") OR LIMIT-TO(DOCTYPE, "ip")) AND (LIMIT-TO(LANGUAGE, "English")) AND (LIMIT-TO(SUBJAREA, "COMP") OR LIMIT-TO(SUBJAREA, "ENGI") OR LIMIT-TO(SUBJAREA, "BUSI")) AND (LIMIT-TO (SRCTYPE, "p") OR LIMIT-TO(SRCTYPE, "j") OR LIMIT-TO(SRCTYPE, "k")) AND (LIMIT-TO(LANGUAGE, "English"))	702	315	17
IEEE Xplore	("function point") OR ("function points") OR ("function-point") OR ("function-points") Year: 2002–2014; Content Type: Conference, Journals & Magazines, Early Access Articles, Standards	187	44	8
ACM DL	("function point" OR "function points" OR "function-point" OR "function-points"). Publication Year: 2002 ... 2014	148	39	6
Science Direct	pub-date > 2001 and "function point" OR "function points" OR "function-point" OR "function-points" Journals(Comp. Sci., Eng.)	563	56	3

[1][Legend] ABS: abstract; KEY: keywords; PUBYEAR: publication year; DOCTYPE: document type; cp: conference paper; ar: article; ip: article in press; SUBJAREA: subject area; COMP: computer science; ENGI: engineering; BUSI: business, management and accounting; SRCTYPE: source type; p: conference proceedings; j: journal; k: book series; pub-date: publication date.
[2]The papers returned in more than one database were only considered the first time they were identified during the 1st iteration, since duplicate results are of no interest in this SLR.

315 from Scopus, 44 from IEEEXplore, 39 from ACM Digital Library (DL), and 56 from Science Direct.

In the second iteration, the entire papers were read in order to determine whether the selected studies in the first iteration were in fact aligned to the aims of this SLR. Also, in this second iteration, the references of the selected papers were studied, mainly regarding the related works, to search for additional studies not returned through the databases used. In the end, 18 items were selected, including duplications: 17 from Scopus, 8 from IEEEXplore, 6 from ACM, 3 from Science Direct, and only 1 from checking references from previous selected papers. Duplicated papers were discarded.

D. Inclusion and Exclusion Criteria and Quality Criteria

The following inclusion and exclusion criteria were defined to refine the selection of papers considering the objectives of this SLR and to ensure the quality of studies selected.

1) Inclusion criteria.
    a) Paper focused on FPA whose main objective is to identify any accuracy-related limitations or deficiency in the method.
    b) Paper focused on FPA whose main objective is to propose improvements in the existing method procedures to solve the accuracy problems identified.
    c) Full paper published and available in online scientific databases.
    d) Paper published after Jan 1, 2002, since the last CPM version (4.X) was published by the IFPUG in 2002. It was, therefore, considered that the proposed possible improvements previous to the current version of CPM would not bring significant results for the objective of this SLR, since it would be based on an outdated CPM version.
    e) Paper published in a journal or proceedings of the event, with peer review.
    f) Paper published in English.
2) Exclusion criteria.
    a) Paper proposing new methods for functional measuring as FPA alternative.
    b) Paper that presents a secondary study.
    c) Paper investigating the relationship of FPA with a particular design paradigm (e.g., object orientation), modeling language (e.g., UML), or a technology to interpret the existing CPM rules for a given context, without proposing accuracy improvements to the method.
    d) Paper whose main objective is to investigate the relationship of FPA (or to compare it) with other functional measurement methods.
    e) Paper whose main objective is to address the convertibility between function points measured through FPA and any other metric generated by other methods.
    f) Paper whose main goal is to make improvements in FPA procedures to improve any issue other than the improvement of accuracy.

The inclusion criteria (b)–(e) were applied during the first iteration, during the selection of studies, by reviewing the paper title and abstract as well as by the research filter of the search engines. The inclusion criteria (a) and (f) and all exclusion criteria were applied during the second iteration, when the content was evaluated.

E. Data Extraction

After the selection of primary studies, they were again read in full and summarized for a better understanding. In parallel, the data needed to answer the research questions of this SLR, as presented in Table V, were extracted from the studies.

During the data extraction, structured forms were completed. In addition to basic information about the papers, the summary of this paper and the observations of the researchers conducting the SLR about the content and conclusions of the study were stored.

Table VII lists the 18 selected primary studies resulting from the criteria application.

IV. RESULTS OBTAINED

In this section, the SLR results are presented considering each research question developed for this SLR. For each research question, the results are divided into three categories

TABLE VII
SELECTED PRIMARY STUDIES

Ref	Selected study	Year	Type[3]	Source[4]
[18]	The software maintenance project effort estimation model based on function points	2003	J	S
[19]	Fuzzy modeling for FPA	2003	J	S; A
[20]	Adapting FPA to estimate data mart size	2004	C	S; I
[21]	Improved standard FPA method-solving problems with upper boundaries in the complexity rating process	2005	J	S; D
[22]	Modification of standard function point complexity weights system	2005	J	S; D
[23]	Neurofuzzy approach to calibrate function points	2007	C	R
[24]	A new calibration for function point complexity weights	2008	J	S; D
[25]	Integrating function point project information for improving the accuracy of effort estimation	2008	C	S; I; A
[26]	A neurofuzzy model for function point calibration	2008	J	S; A
[27]	Software security characteristics for FPA	2009	C	S; I
[28]	Mapping GSC to nonfunctional requirements	2009	C	S; I
[29]	Study of FPA based on fuzzy interpolation	2010	J	S
[30]	A Software size estimation method based on improved FPA	2010	C	S; I
[31]	Extended FPA prototype with security costing estimation	2010	C	S; I
[32]	Error correction in function point estimation using soft computing method	2011	C	S, A
[33]	Enhancement of software projects' FPA based on conditional nonfunctional judgments	2012	C	S; I
[34]	Research on VAF of the IFPUG method based on fuzzy analytic hierarchy process	2012	C	S; I; A
[35]	Estimating the size of data mart projects	2013	C	S; A

[3]Type: J Journal; C Conference.
[4]Source: S Scopus; I IEEE; A ACM; D Science Direct; R References.

TABLE VIII
STUDIES DIVIDED INTO CATEGORIES OF TYPES OF PROPOSED IMPROVEMENTS

Studies	Weights and complexities	Technological independence	Adjusted functional size
[23], [24], [26], [30]	X	X	
[19], [21], [22], [25], [29], [32]	X		
[20], [35]		X	X
[18], [27], [28], [31], [33], [34]			X

of improvement proposals, which are related to: 1) weights and complexities determined for each BFC; 2) technological independence of the method; and 3) calculating the adjusted functional size. These three categories were determined after grouping the improvements proposed in each primary study.

Table VIII lists the studies mapped considering the three categories. "Weights and complexities" contains all studies that refer to the number of function points allocated to each complexity, as initially determined by Albrecht [4]. "Technological independence" contains all studies that refer to the fact that FPA functional measure is not linked to any technology, generating a functional size based only on the user's view. Finally, "adjusted functional size" contains all studies that refer to GSC in terms of limitations and proposals to include other nonfunctional requirements.

Following, each of the four research questions are discussed in relation to the results found in the studies of this SLR, grouped in these three categories.

All six studies classified as "technological independence" were identified as addressing this only secondarily; i.e., reporting problems linked to technological independence without proposing any improvement. Therefore, these studies are discussed regarding only the context of the question RQ1.

A. RQ1—Are There FPA Problems Being Reported?

The analysis of the selected studies showed FPA problems related to the three categories presented in Table VIII. While 56% of the studies are related to "weights and complexities," 44% are related to "adjusted functional size" and 33% related to "technological independence." Some of the works are classified in more than one category. Following, the main problems reported in the 18 studies are briefly described.

1) Weights and Complexities: All the ten studies of this category point to FPA failures in correctly scoring the BFC of the software. According to their authors, the weights and complexities determined (see Table IV) for the BFCs are not suitable, given that a same data function and/or a same transactional function with different combinations of DET and RET/FTR (see Tables I–III) can be classified with the same complexity. Consequently, they can result in data functions and/or transactional functions with the same number of function points. They also point out that, in some situations, functionalities that have very similar DET and RET / FTR can be classified with different complexities, and hence, receive different weights according to FPA.

Ya-Fang *et al.* [30] highlighted the weights of BFC, as currently set by the IFPUG, do not reflect the software functional size. They believe the range of values for the complexities for data and transactional functions is too large, allowing huge variations within a given complexity. According to Junior *et al.* [19], this inconsistency becomes even worse when a large number of BFC lies within the boundary areas of the specified intervals, as the inaccurate classification of various system functionalities would distort its functional size.

According to Xia *et al.* [23], [24], [26], BFC' weights do not reflect current software industry trends, representing a potential issue to derive software costs and effort. Al-Hajri *et al.* [22] and Ahmed *et al.* [25] agree, considering that the classification of the BFC with low, medium, or high complexity do not fully reflect the effort required to develop the user's software. Al-Hajri *et al.* [22] reported that FPA suffers from weight weakness, where they found that the choices of weights for calculating function points were determined subjectively from IBM experience.

Chen *et al.* [29] argued that FPA not being able to accurately make the translation from element type number and complexity grade into function point represents a weakness.

2) Technological Independence: Six studies address FPA issues related to the independence from any technology or technical factors. Calazans *et al.* [20] reported that technologies have specific features that must be considered in the measurement, and that PFA, when defined three decades ago, considered the existing hardware and software of that time, which are now obsolete. Xia *et al.* in [23], [24], [26] and Ya-Fang *et al.* [30] stated that FPA is not suitable to increase the accuracy of estimates as they consider it outdated and does not reflect the actual software development process, including—for example—the advent of the object-orientated paradigm. Ferreira and Marques-Neto [35] highlighted that data marts are software examples that have specific features and that should be taken into account in the measurement.

3) Adjusted Functional Size: Finally, eight studies address the problems related to the calculation of the adjusted functional size of software. According to Bharadwaj and Nair [28], for example, adjusting the software calculation as proposed does not consider some important nonfunctional requirements for the current context, such as efficiency, usability, maintainability, and portability. Moreover, they believe that the GSC Transaction rate is no longer applicable to the current context due to technological progress, high internet speed rates, high disk access speed, and CPUs with high megahertz processing [28].

Abdullah *et al.* [27], [31] argued FPA is not covering all the GSCs needed to determine the adjusted functional size as, for example, aspects related to the information security area. Thus, they strongly recommend the incorporation of new GSCs for FPA focused on this area.

Ahn *et al.* [18] raised some issues related to GSCs: 1) they do not consider relevant features to software maintenance; 2) they were proposed for the context of new projects; 3) GSCs such as data communications', distributed data processing', and performance' are inapplicable to software maintenance; and 4) they cannot be objectively measured.

Peng *et al.* [34] reported that the assignment of the VAF for the 14 GSCs did not take into account the heterogeneity of different types of systems such as, for example, real-time systems. For them, the distribution of VAFs should be more heterogeneous in order to better reflect some specific situations. For example, some GSCs that have greater relevance to a specific type of software should have greater influence on the adjusted size of such software when compared to other GSCs.

Addressing another aspect, Matijevic *et al.* [33] reported that, as currently set, determining the influence of each GSC is a difficult task to be performed by the responsible for determining each influence factor of the GSCs. According to Matijevic *et al.* [33], the user cannot generally easily identify the value to be assigned to a particular GSC. In addition, according to this study, this is a serious situation given that a nonfunctional requirement can influence the complexity of a software development project more than a functional requirement.

TABLE IX
AI Techniques Proposed for the Category "Weights and Complexities"

Studies	Technique Used
[19], [32]	Fuzzy logic
[22]	Artificial neural networks
[23], [24], [26]	Fuzzy logic; artificial neural networks; statistical regression
[25]	Genetic algorithms
[29]	Fuzzy logic; interpolation methods
[30]	Fuzzy logic; artificial neural networks

B. RQ2—What Improvements are Being Proposed for FPA?

Next, the main improvement proposals for FPA are presented, considering the problems exposed for RQ1, in view of the categories "weights and complexities" and "adjusted functional size." For "technological independence," no improvement is shown, since there was no study showing improvements for the problems pointed out.

1) Weights and Complexities: In general, the improvements for FPA relative to "weights and complexities" propose ways for generating additional weights and complexities to those already existing in the method, or for calibrating the weights already derived by the IFPUG.

Among the ten studies reporting problems related to the weights or complexities of each BFC, nine propose FPA improvements to include the use of fuzzy logic, artificial neural networks, or genetic algorithms, from the artificial intelligence (AI) area. In some of these studies, other techniques are also proposed. Table IX lists the respective techniques proposed in each of these nine studies as a solution to the problems related by research question RQ1.

The use of fuzzy logic is proposed in seven studies for deriving the functional size, that is, for deriving the weights for the BFC [19], [23], [24], [26], [29], [30], [32]. On another hand, artificial neural networks are proposed in five studies [22]–[24], [26], [30]. Out of these ten studies in total, four propose applying some AI technique alone, without its use compromising the accuracy and precision of the functional sizes derived from the techniques determined by them [19], [22], [25], [32]. In the other studies, the use of fuzzy logic is proposed in combination with other techniques [23], [24], [26], [29], [30], as described as follows.

Xia *et al.* in [23], [24], and [26] proposed combining three techniques in a neurofuzzy function point calibration model. They propose to combine fuzzy logic, artificial neural networks, and statistical regression. Statistical regression is a mathematical technique used to represent the relation between the selected values and observed values of statistical data. Artificial neural networks are proposed to facilitate learning based on previous data. And fuzzy logic is proposed to make rational decisions

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

in an environment of uncertainty and imprecision. The fuzzy logic concept was incorporated into the neural network to generate new values for the weights or the complexities. Similarly, Ya-Fang *et al.* [30] also proposed a combination of fuzzy logic with artificial neural networks so that, from the rules and linguistic terms defined by fuzzy logic, the neural networks could learn from previous data.

Chen *et al.* [29] proposed the combination of fuzzy logic with interpolation methods. It is proposed because although fuzzy logic can address the problems reported in the research question RQ1, the distribution of values for the complexities remains a problem.

Kralj *et al.* [21], unlike the studies listed in Table IX, are not using AI techniques. As alternative, they proposed to build a technique based on the growth rate of the functional size, as measured by the Common Software Measurement International Consortium (COSMIC) and Mk II FPA. These functional measurement techniques are those that exhibit, respectively, the lowest and highest functional size growth rate according to this study.

Ahmed *et al.* [25] proposed that new weights for FPA were calculated based on an adapted genetic algorithm. The proposed algorithm is based on: an initial data population; a set of chromosomes and their genes and containing actual parameters with values relating to the current complexities of FPA; cross-over operations; and mutation operations. Its purpose is to generate new and better solutions, based on the genetic algorithms technique and evolutionary computation, in terms of weights more appropriate for FPA.

2) Adjusted Functional Size: Calazans *et al.* [20] and Ferreira and Marques-Neto [35] based their work on the data warehouse software analysis and determined that the calculation of FPA adjusted functional size could be improved to measure this type of software more accurately. Therefore, 25 new GSCs are proposed. Moreover, Calazans *et al.* [20] proposed the interpretation of the FPA rules specifically for data warehouse software to standardize the counting for this type of software. However, such improvements are outside the analysis scope of this SLR, according to the criteria presented in Section III-D.

Abdullah *et al.* [27], [31] in turn, suggested the inclusion of 58 new GSCs, all focused on the software security context, such as: encryption, sensitive data exchange, and protection of technology security. For each GSC proposed, a way to score its influence on the project being measured was also proposed. Five scales were determined, ranging from the minimal influence to the maximum influence so that users could determine the influence that a given GSC would have. Parameters were also proposed to assist the user in the scoring.

Ahn *et al.* [18] proposed the inclusion of ten new GSCs focused on software maintenance. The new GSCs are grouped into three categories: 1) people perspective—engineer's skills; 2) product perspective—technical characteristics; and 3) process perspective—environment characteristics. Considering these new GSCs, the SMPEEM (Software Maintenance Project Effort Estimation Model) effort derivation model is proposed.

The determination of scales, from one to five points, to score the influences was also proposed by Calazans *et al.* [20] and Ferreira and Marques-Neto [35] so that the influence of the GSCs of the data warehouse software could be determined. These values would then be incorporated into the functional size measured to reflect these aspects.

Peng *et al.* [34] proposed to calculate the degree of influence (DI) for each GSC considering the importance of the GSC for the specific type of software being analyzed. They propose the fuzzy AHP approach, which aims at defining the weight of each DI assigned to a certain GSC, considering a set of defined procedures. The new DI is calculated as a result of the product between the flowing two variables: a score of 1 to 5, as provided for in the original FPA; and a constant calculated by the fuzzy AHP approach that represents the degree of importance of that GSG for that particular software.

Matijevic *et al.* [33] proposed an alternative way to score each GSC to determine the DI. Since the influence of each GSG cannot be easily determined by the applicators of the method, they propose using the CS-AHP algorithm (a derivation of his S-AHP method) to process different types of requirements and combine them with FPA. Through this algorithm, the user determines the degree of importance a GSG has with another one. For example, usability can have a weight of five in relation to scalability; while performance may have a weight of three in relation to the same scalability. Finally, based on the importance values of the GSCs determined by the user, a matrix is assembled to calculate an overall importance ranking of all the GSCs. From this ranking, the highest value is considered proportionally as the highest possible value for a GSC; which is five. The others are adjusted proportionally, considering the new scale of five. According to these authors, implementing this procedure when determining the DI for each GSC would enable the user to more easily or more accurately determine the total of degree of influence (TDI) of each GSC. This would enable its VAF as well as the adjusted functional size of software to have greater accuracy.

C. RQ3 and RQ4—Are the Proposed Improvements Effective? Are There Limitations to the Improvements Proposed?

Following, the main advantages and the main limitations pointed out in the studies evaluated are presented, considering—in a combined manner—the research questions RQ3 and RQ4. The results are again presented only for the categories "weights and complexities" and "adjusted functional size," since for the category "technological independence" no improvement was verified for the problems indicated.

1) Weights and Complexities: In general, the studies proposing improvements exclusively for the weights or complexities concluded that the improvements resulted in more accurate functional size estimates. Table X summarizes the improvement percentages shown by each study in this category, mainly related to mean magnitude of relative error (MMRE).

The MMRE percentage, used to evaluate the improvement most of these proposals, measures, for a given project, the

TABLE X
IMPROVEMENTS OBSERVED REGARDING "WEIGHTS AND COMPLEXITIES"

Ref.	Percentage of improvement	#Projects	Data source	Technology
[19]	8.8% improvement in MMRE3	9	Provided by systems analysts trained in FPA	4GL, Java, VB
[21]	13% improvement in effort derivation when compared with the value derived from the traditional method described in the CPM	20	ISBSG database[5]	VB, Access, Natural 2
[22]	30% improvement in MMRE	206	ISBSG database	Not informed
[23]	22% improvement in MMRE	184	ISBSG database	Not informed
[24]	22% improvement in MMRE	184	ISBSG database	Not informed
[25]	50% improvement in MMRE	184	ISBSG database	Not informed
[26]	22% improvement in MMRE	600	ISBSG database	Not informed
[29]	There was no proposal validation	-	-	-
[30]	3.6% improvement in MMRE[6]	Not informed	Not informed	Not informed
[32]	There was no proposal validation	-	-	-

[5]International Software Benchmarking Standards Group (www.isbsg.org).
[6]Value not shown directly in the study evaluated. For the sake of comparison in this SRL, the MMRE percentage regarding the original method was obtained by calculating the arithmetic average of the improvement percentages for all projects measured in the scope of the study.

difference between actual and estimated effort relative to the actual effort. The mean takes into account the numerical value of every observation in the data distribution, and is sensitive to individual predictions with large MRE [36].

Although the evidences shown that the proposed FPA improvements for scoring the BFC can provide greater accuracy in the development effort derivation, some limiting factors were also highlighted, as presented as follows. In all the other studies, no limitation or disadvantage is explicitly presented by the authors.

The approaches proposed by Junior *et al.* [19] and Kralj *et al.* [21] are designed to projects containing mostly high-complexity functions. Thus, the accuracy improvement rates may not have the same performance when there is a large proportion of functions with simple and medium complexities in the project in question. Kralj *et al.* [21], for instance, experimented their approach in a propitious setting for which it was designed: projects containing approximately 85% of high-complexity functions.

On the other hand, the approaches proposed by Al-Hajri *et al.* [22], Xia *et al.* [23], [24], [25], and Ahmed *et al.* [26] are not aimed at any certain scenario, providing the possibility of application to different types of projects regarding the levels of complexity. However, these proposals still holds some of the effects observed to the original FPA: "a same data function and/or a same transactional function with different combinations of DET and RET/FTR can be classified with the same complexity"; since these proposals only provide the calibration for the current weights of the original FPA technique, without having proposed new complexities that would confer greater granularity to FPA. In summary, from these ones, the approach proposed by Al-Hajri *et al.* [22] is, in fact, the one presenting the greater possibility of applying to different projects.

Ya-Fang *et al.* [30], for example, pointed out that using backpropagation—a training method for artificial neural networks—shows a slow speed for convergence rate. This can compromise their method in terms of speed to obtain the results. Following this same idea, although with no limitations highlighted, Xia *et al.* [23], [24], [26] also proposed backpropagation as a training method for artificial neural networks, which may be understood as a limiting factor for their work, due to this method's low speed of convergence rate.

TABLE XI
IMPROVEMENTS OBSERVED REGARDING "ADJUSTED FUNCTIONAL SIZE"

Study	Percentage of improvement
[18]	There was no result regarding the MMRE. It was concluded that the inclusion of the ten GSCs are not as influential as one expected.
[20]	There was no result regarding the MMRE. It was concluded that there is an improvement of the proposed method over the traditional method.
[27]	There was no validation of the proposed method.
[28]	There was no validation of the proposed method.
[31]	There was no validation of the proposed method.
[33]	There was no validation of the proposed method.
[34]	There was no result regarding the MMRE. About a 10% improvement was observed (for five specific projects) for estimating effort when considering the new DI adapted.
[35]	There was no result regarding the MMRE. An improvement was observed over both: the traditional method and the proposal of Calazans, Oliveira and Santos [20].

For the method proposed by Chen *et al.* [29], the results were obtained only based on tests performed on software of the university Education Management Information System. This may have distorted the results and increased the reliability of the proposed method due to the low number of applications measured in the tests performed in the study conducted.

2) Adjusted Functional Size: For the studies that propose FPA improvements related to GSC, results related to the advantages achieved were presented only for the measurement applied in the data warehouse software [20], [35]. For the other studies [27], [28], [31], [33], no validation reports of the proposed methods were identified in order to verify if there was any improvement in the method proposed regarding FPA. Table XI summarizes the results presented by the authors of these studies regarding the advantages achieved.

Regarding the application for measuring the data warehouse software, both methods proposed showed higher accuracy in estimating the functional size, when compared to the standard FPA method [20]. application of the Tukey's test, Calazans *et al.* [20]

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

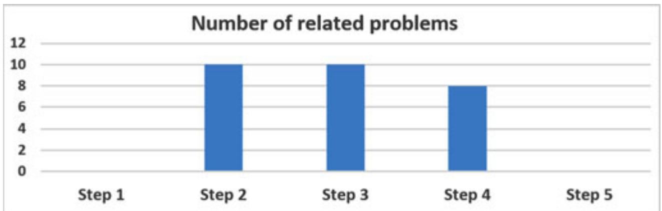

Fig. 1. Number of issues identified by the studies for the five steps of FPA.

concluded that there are significant differences between both methods, and that the proposed method has higher accuracy when compared to the standard method for measuring a data warehouse software.

Ferreira and Marques-Neto [35] found that, after the application of their method, the effort derived from the measured functional size is more accurate when compared to the standard method and to the results obtained by Calazans *et al.* [20].

Ahn *et al.* [18] validated their approach using a survey method and regression analysis. Questionnaires were used to collect actual data in four South Korean organizations. As a result, they concluded that the proposed model is good to estimate the effort of maintenance projects and that they can be adjusted in the range of 20% by the ten new GSCs of SMPEEM. However, the new GSCs proposed are not as influential as one expected.

In turn, Peng *et al.* [34] founded that the effort derived through their method showed an improvement of approximately 10% compared to the original FPA. However, these results were based on only an experiment conducted with five projects. In all eight of these studies, no limitation or disadvantage is explicitly presented by the authors.

V. Analysis of Results

In this section, the SLR results presented in the previous section are analyzed and discussed in an attempt to answer each of the research questions developed for this SLR.

A. *RQ1—Are There FPA Problems Being Reported?*

Among the 18 primary studies, ten of them are related to how FPA proposes the functional size calculation as originally determined by the IFPUG. The authors of these papers reported that there is a method failure in scoring the BFC of the software, as reported in Section IV. Therefore, these ten works are related to Step 2 and Step 3 of the FPA procedure described in Section II, as highlighted in the graph of Fig. 1.

Although steps 2 and 3 concentrate the issues identified, they all refer to a single subactivity to be carried out in these two steps, namely, the last subactivity, which is to determine the complexity (high, medium, and low) and the functional size of the data function and/or of the transactional function of each BFC (see Fig. 2). The absence of presenting problems related to the other subactivities of these two steps shows that after the subsequent versions of CPM, these other FPA rules became clearer and more objective, reducing the heterogeneity of functional sizes found by different professionals when based on the same scope.

All the other eight studies have problems related to the GSC currently in place, used in Step 4, because they are considered quite limited. These authors believe that the current GSCs do not include characteristics related to other technologies, as for instance related to data warehouse software or to many other nonfunctional requirements such as software security. As highlighted in the graph of Fig. 1, these eight studies are related to Step 4 of the FPA procedure described in Section II.

Six studies also reported issues related to the technological independence of FPA assuming the current technological context is not reflected in the technique proposed in the 70s. In fact, these issues are not directly related to any FPA step but the technique as a whole.

Throughout the complete period analyzed (2002–2014), only 18 studies were identified pointing to the need for FPA improvements. This may be considered a low number, and perhaps an indication that the method is already at a fairly advanced level of maturity. Regarding the time distribution, the studies identified are evenly distributed, with an average of one to two papers per year. This distribution, although small, indicates that researchers are constantly interested in FPA limitations and improvements.

B. *RQ2—What Improvements are Being Proposed for FPA?*

Different techniques have been proposed to improve FPA. Fig. 3 summarizes the techniques used as proposals for solving these problems. In general, the preponderance of techniques from the AI area is noted, especially those related to the category "weights and complexities." As for the studies of the category "adjusted functional size," most do not have a specific technique for solving the problems. For these studies, the proposal of new GSCs was mostly based on the analysis of nonfunctional requirements.

No improvement was identified for the category "Technological independence," although related issues have been reported. To address this type of problem, high-magnitude changes would probably need to be proposed, given the diversity and complexity of currently existing technology scenarios.

C. *RQ3—Are the Proposed Improvements Effective?*

Among the ten studies that showed some kind of evaluation of the proposed improvement, all showed that this improvement was more effective than the original FPA, as detailed in Tables X and XI.

Regarding the category "weights and complexities," the majority of the proposed works (90%) have used techniques from the AI area. In general, positive results were presented, with great improvement in deriving the effort. Consequently, the use of AI techniques is presented as a trend for the development of more accurate weights and complexities for FPA.

Regarding the category "adjusted functional size," from the seven proposals, only two studies presented data about the evaluation of the proposed improvement. Based on these two studies, there is evidence that the improvements are effective for the context addressed by them, as shown in Table XI. However, further studies should be conducted considering mainly the cases described in the other studies so that different contexts can

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

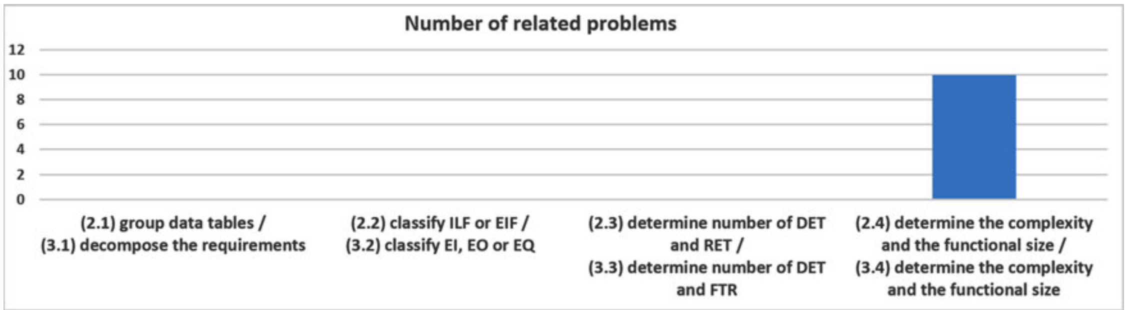

Fig. 2. Number of issues identified by the studies for the Steps 2 and 3.

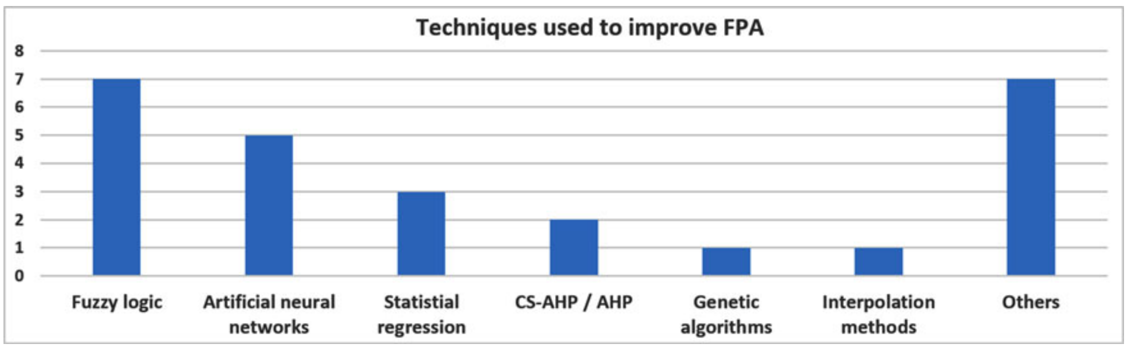

Fig. 3. Techniques used in the improvements for the issues identified in FPA.

be tested and validated and no bias impair the final validation results.

This analysis is not applicable for the category "technological independence" since no improvement was proposed to address this problem.

D. RQ4—What are the Limitations for the Proposed Improvements?

Regarding proposals to improve the category "weights and complexities," the results show some studies not able to solve this problem completely. Though the proposed advances have alleviated the problem of enabling that a same data function or a transaction function, with different combinations of DET and RET / FTR, may be able to receive the same weights and/or complexities. All the works of this category proposed FPA improvements in order to make it more granular and be able to rate various combinations of DET and RET / FTR with different values with respect to the original method. However, the undesired effect is not completely eliminated.

For the improvement proposals related to the category "adjusted functional size," the strategy of including nonfunctional requirements related to some specific additional aspects such as security could make the measuring procedure still more costly and complex. Especially when the goal of FPA is to essentially measure the software considering only the functional requirements of software. Taking all the selected studies, the inclusion of 93 new GSCs was proposed to make the adjusted functional size calculation more accurate, comparing to the original 14 ones. These new GSCs would require additional time for careful analysis by the professionals responsible for measuring the software, adding costs to the measurement procedure. In general, the proposed improvements might offer more accuracy in the adjusted functional size when compared to the original FPA only when the improved method is implemented in a specific scenario, as for instance, in data warehouse software. In a general context, the improvement tends to keep the method's standard accuracy. Thus, these improvements have a limited scope, not increasing the accuracy of FPA in any settings.

VI. Conclusion

The purpose of this SLR was to identify improvements proposed for FPA to make the results generated more accurate. Four different bibliographic databases were researched. From the search conducted in the databases, 1600 papers were returned, out of which, after applying the defined systematic protocol, 18 primary studies were identified, read, and reported in this paper. The low number of studies that propose FPA improvements might demonstrate the maturity of the method in the current scenario of software metrics.

Among the various steps of the FPA measurement procedure, there is evidence, identified through this SLR, that steps "2—measure transactional functions," "3—measure data functions," and "4—calculate the functional size" are the ones most in need of adjustments. Eighteen studies with improvement proposals were found for these steps of the method. The high proportion of improvement proposals for these steps may suggest to the

IFPUG, for example, that they should be revised in order to reflect the quantification of the user's functional requirements more accurately.

The improvements proposed in the selected 18 studies, were analyzed and grouped into three categories: 1) "weights and complexities" determined for each BFC; 2) 'technological independence" of the method; and 3) calculating the 'adjusted functional size."

The category 'weights and complexities" has a clear picture in this study. It is the one with the highest number of studies reporting problems and also suggestions for improvement. On the other hand, almost all the improvement proposals resort to IA techniques, alone or in conjunction to some other technique. In general, the evaluated studies show positive results, indicating that the use of this type of solution can be quite beneficial for this type of problem. In conclusion, this study indicates the use of AI techniques as a potential way to obtain a more accurate functional size by applying FPA. However, this type of solution necessarily must be accompanied by computational support required for the application of AI techniques, which can be viewed as a possible drawback since FPA has originally the simplicity of being able to be used independently of specialized software.

For the category 'technological independence," only problems were reported by the authors of the selected studies, but no improvement proposal was. In general, FPA may be considered as an outdated method, even being technologically independent, since it was proposed almost 40 years ago. Consequently, though questionable, FPA could be not suitable anymore to produce accurate numbers since hardware and software has evolved substantially these last decades. As a possible explanation for the fact that this issue is only raised, but no solution is presented, it is considered that it would be very difficult to solve it. To address this type of problem, high-magnitude changes would probably need to be proposed, given the great diversity and complexity of currently existing technology scenarios.

Finally, for the category 'adjusted functional size," almost half of the selected studies present some problem related to this problem, and also some improvement proposal. However, differently of the first category previously discussed, in this case, there is no clear indication of how to solve the reported problems. In general, the different studies propose the inclusion of several new GSCs. Although these new GCSs could make the final metric more accurate, this type of solution could lead to a series of side effects, such as: 1) increasing effort and cost to apply FPA; 2) increasing reliance on experts to help in counting these additional GCSs; 3) increasing the need to apply FPA later to get the values for these additional GCSs; and 4) increasing the incentive to apply FPA only partially, without using all GCSs. In fact, the IFPUG already shows signs that they intend to eliminate definitely this problem within FPA instead of solving it, due to its high complexity. Currently, obtaining the adjusted functional size is an optional step in the current version of the CPM. In addition, IFPUG proposed the SNAP (software nonfunctional assessment process) technique, which aims to measure the nonfunctional requirements of the software's user, in addition to measure the functional software requirements of the software through FPA. Thus, the next step might be to definitively eliminate the FPA's GCSs.

REFERENCES

[1] M. J. Liberatore and B. Pollack-Johnson, "Improving project management decision making by modeling quality, time, and cost continuously," *IEEE Trans. Eng. Manag.*, vol. 60, no. 3, pp. 518–528, Aug. 2013.

[2] H. D. Frederiksen and L. Mathiassen, "A contextual approach to improving software metrics practices," *IEEE Trans. Eng. Manag.*, vol. 55, no. 4, pp. 602–616, Nov. 2008.

[3] T. Ravichandran, "Organizational assimilation of complex technologies: An empirical study of component-based software development," *IEEE Trans. Eng. Manag.*, vol. 52, no. 2, pp. 249–268, May 2005.

[4] A. J. Albrecht, "Measuring application development productivity," in *Proc. IBM Appl. Develop. Symp.*, Montara, CA, USA, 1979, pp. 83–92.

[5] IFPUG. (2014). International Function Point Users Group. [Online]. Available: http://www.ifpug.org

[6] IFPUG, "Function point counting practices manual, release 4.3.1," International Function Point Users Group, Westerville, OH, USA, 2010.

[7] MPOG, "Instrução normativa SLTI/MPOG no. 4," (in Portuguese), Brazil's Ministry of Planning, Budget and Management, Brasília, Brazil, 2010.

[8] MPOG, "Portaria SLTI/MPOG no. 31," (in Portuguese), Brazil's Ministry of Planning, Budget and Management, Brasília, Brazil, 2010.

[9] P. Kampstra and C. Verhoef, "Reliability of function point counts," Department of Computer Science, VU University Amsterdam, Amsterdam, The Netherlands, 2010.

[10] C. F. Kemerer, "Reliability of function points measurement: A field experiment," *Commun. ACM*, vol. 36, no. 2, pp. 85–97, 1993.

[11] R. Meli, "Functional metrics: Problems and possible solutions," in *Proc. 1st Eur. Software Meas. Conf.*, Antwerp, Belgium, 1998, pp. 503–514.

[12] M. Jørgensen and M. Shepperd, "A systematic review of software development cost estimation studies," *IEEE Trans. Softw. Eng.*, vol. 33, no. 1, pp. 33–53, Jan. 2007.

[13] K. Petersen, "Measuring and predicting software productivity: A systematic map and review," *Inf. Softw. Technol.*, vol. 53, no. 4, pp. 317–343, 2011.

[14] C. E. Vazquez, G. S. Simoes, and R. M. Albert, *Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software* (in Portuguese). São Paulo, Brazil: Érica, 2013.

[15] B. Kitchenham, "Procedures for performing systematic reviews—Joint technical report," Keele University, Keele ,U.K. and National ICT Australia Ltd., Sydney, Australia, Tech. Rep. TR/SE-0401 and NICTA TR 0400011T.1, 2004.

[16] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering—Joint technical report," Keele University, Keele ,U.K. and University of Durham, Durham, U.K., Tech. Rep. EBSE-2007-01, 2007.

[17] J. Biolchini, P. G. Mian, A. C. C. Natali, and G. H. Travassos, "Systematic review in software engineering," Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, Tech. Rep. RT-ES 679/05, 2005.

[18] Y. Ahn, J. Suh, S. Kim, and H. Kim, "The software maintenance project effort estimation model based on function points," *J. Softw. Maintenance Evol.*, vol. 15, no. 2, pp. 71–85, 2003.

[19] O. S. L. Junior, P. P. M. Farias, and A. D. Belchior, "Fuzzy modeling for function points analysis," *Softw. Quality Control*, vol. 11, no. 2, pp. 149–166, 2003.

[20] A. T. Calazans, K. M. Oliveira, and R. R. Santos, "Adapting function point analysis to estimate data mart size," in *Proc. 10th Int. Symp. Softw. Metrics*, Chicago, IL, USA, 2004, pp. 300–311.

[21] T. Kralj, I. Rozmanb, M. Heričkob, and A. Živkovičb, "Improved standard FPA method: Resolving problems with upper boundaries in the rating complexity process," *J. Syst. Softw.*, vol. 77, no. 2, pp. 81–90, 2005.

[22] M. A. Al-Hajri, A. A. A. Ghani, M. N. Sulaiman, and M. H. Selamat, "Modification of standard function point complexity weights system," *J. Syst. Softw.*, vol. 74, no. 2, pp. 195–206, 2005.

[23] W. Xia, L. F. Capretz, and D. Ho, "Neuro-fuzzy approach to calibrate function points," in *Proc. 8th WSEAS Int. Conf. Fuzzy Syst.*, Vancouver, Canada, 2007, pp. 116–119.

[24] W. Xia, L. F. Capretz, D. Hoc, and F. Ahmedd, "A new calibration for function point complexity weights," *Inf. Softw. Technol.*, vol. 50, no. 7–8, pp. 670–683, 2008.

[25] F. Ahmed, S. Bouktif, A. Serhani, and I. Khalil, "Integrating function point project information for improving the accuracy of effort estimation," in *Proc. 2nd Int. Conf. Adv. Eng. Comput. Appl. Sci.*, Valencia, Spain, 2008, pp. 193–198.

[26] W. Xia, D. Ho, and L. F. Captrez, "A neuro-fuzzy model for function point calibration," *WSEAS Trans. Inf. Sci. Appl.*, vol. 5, no. 1, pp. 22–30, 2008.

[27] N. A. S. Abdullah, R. Abdullah, M. H. Selamat, and A. Jaafar, "Software security characteristics for function point analysis," in *Proc. IEEE 16th Int. Conf. Ind. Eng. Eng. Manag.*, Hong Kong, 2009, pp. 394–397.

[28] A. K. Bharadwaj and T. R. G. Nair, "Mapping general system characteristics to non-functional requirements," in *Proc. IEEE Int. Adv. Comput. Conf.*, Patiala, India, 2009, pp. 1821–1825.

[29] Q. Chen, R. Cheng, S. Fang, and Y. Ou, "Study on function point analysis based on fuzzy-interpolation," *J. Comput. Inf. Syst.*, vol. 6, no. 5, pp. 1369–1375, 2010.

[30] F. Ya-Fang, L. Xiao-Dong, Y. Ren-Nong, D. Yi-Lin, and L. Yan-Jie, "A software size estimation method based on improved FPA," in *Proc. 2nd World Congr. Softw. Eng.*, Wuhan, China, 2010, pp. 228–233.

[31] N. A. S. Abdullah, R. Abdullah, M. H. Selamat, and A. Jaafar, "Extended function point analysis prototype with security costing estimation," in *Proc. Int. Symp. Inf. Technol.*, Kuala Lumpur, Malaysia, 2010, pp. 1297–1301.

[32] K. K. Rao and G. S. Raju, "Error correction in function point estimation using soft computing technique," in *Proc. Int. Conf. Adv. Comput. Artif. Intell.*, Rajpura, India, 2011, pp. 194–198.

[33] T. Matijevic, I. Ognjanovic, and R. Sendelj, "Enhancement of software projects' function point analysis based on conditional non-functional judgments," in *Proc. Mediterranean Conf. Embedded Comput.*, Bar, Montenegro, 2012, pp. 283–287.

[34] H. Peng, G. X. Yang, and L. Cai, "Research on VAF of IFPUG method based on fuzzy analytic hierarchy process," in *Proc. IEEE/ACIS 11th Int. Conf. Comput. Inf. Sci.*, Shanghai, China, 2012, pp. 593–597.

[35] W. G. Ferreira and H. T. Marques-Neto, "Estimating the size of data mart projects," in *Proc. 28th Annu. ACM Symp. Appl. Comput.*, Coimbra, Portugal, 2013, pp. 1147–1148.

[36] E. Mendes and S. Abrahao, "Web development effort estimation: An empirical analysis," in *Handbook of Research of Web Information Systems Quality*, C. Calero, M. A. Moraga, and M. Piattini, Eds. Hershey, PA, USA: IGI Global, 2008.

Marcos de Freitas Junior received the Technological degree in systems analysis and development from the Thermomechanical Technology Faculty, Salvador Arena Foundation, São Bernardo do Campo, Brazil. He is currently working toward the Master of Science degree in information systems at the University of São Paulo, São Paulo, Brazil.

He is currently a Metric Specialist at "ti Métricas," Brazil. His main research interests include software metrics, FPA counting automation, and software cost and time estimation.

Marcelo Fantinato received the Bachelor degree in computer science in 1999 from the State University of Maringa, Maringa, Brazil, and the Master of Engineering degree in 2002 and the Ph.D. degree in computer science in 2007 from the University of Campinas, Campinas, Brazil.

He is an Associate Professor at the University of São Paulo, São Paulo, Brazil. He is currently the Coordinator of the Master of Science Graduate Program in Information Systems. He worked as a Software Testing Specialist at the CPqD Foundation (2001–2006) and as a Specialist in research and development at Motorola (2006–2008), Brazil. He has a Green Belt certification in the Motorola Six Sigma program for process improvement. His main research interests include business process management, service-oriented computing, software product line, software quality, and software testing.

Violeta Sun received the Bachelor in Business Administration degree in 1986 from the Fundação Armando Álvares Penteado, São Paulo, Brazil and the Master in Business Administration degree in 2005 and the Ph.D. degree in 2010 from the University of São Paulo, São Paulo. A part of her Ph.D. was developed at the University of Oslo, Oslo, Norway.

She is an Assistant Professor at the University of São Paulo. Her main research interests include project management, information technology management, IT infrastructure, IT governance, management systems in health, and computer application in public administration.