A Functional Size Measurement Procedure for MVC Applications from Source Code: Design, Automation and Empirical Evaluation

Christian Quesada-López
Center for ICT Research
University of Costa Rica
San Pedro, Costa Rica
cristian.quesadalopez@ucr.ac.cr

ABSTRACT

Software size has proved to be one of the main effort-and-cost drivers. It is widely accepted that software size is one of the key factors that has the potential to affect the effort and cost of the software projects. Functional size methods are hardly automatable and generally require a lengthy and costly process. FSM methods define generic concepts and measurement rules. The setup of a measurement procedure for each input to the measurement process is hence needed. A measurement procedure is defined as a set of operations described explicitly in order to measure according to a specific measurement method. A size estimation procedure based on the use of software development productivity models allows the management of development costs. Although accurate size estimation and effort prediction are very important for the success of any project, many practitioners have experienced difficulties in applying them. Thus, automated and simplified FSM methods are required.

This research aims to proposing a functional size measuring procedure for Model-View-Controller (MVC) applications from source code. The research project includes the design, automation and empirical validation of a functional size measuring procedure, according to the ISO/IEC 20926 FPA CPM method. This proposal describes the research agenda of the PhD project. Research objective, background, relevant, prior work, research methods, threats to validity, current status and future plans are described in details. Since this is an ongoing work, this proposal looks for feedback from the expert community in order to improve its consistency as well as the reliability of the empirical validation.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics – *Product metrics*

General Terms

Measurement, Management, Experimentation.

Keywords

Functional size measurement, functional size procedures (FSM), software size estimation, function points, software development productivity models, source code, empirical validation.

ESEM'14, September 18–19, 2014, Torino, Italy.

1. INTRODUCTION

1.1 Motivation

Software estimation process is a key factor for software project success [1]. The complexity to provide accurate software size estimation and effort prediction models in software industry is well known. The need for accurate size estimates and effort predictions for projects is one of the most important issues in the software industry [2]. Inaccurate estimates are often the main cause for a great number of issues related to low quality and missed deadlines [3]. Software size measurement is an important part of the software development process. Functional size measures are used to measure the logical view of the software from the users' perspective by counting the amount of functionality to be delivered. These measures can be used for a variety of purposes, such as project estimation, quality assessment, benchmarking, and outsourcing contracts. According to [4], functional size measurements can be used for:

- Budgeting software development or maintenance
- Tracking the progress of a project
- Negotiating modifications to the scope of the software
- Determining the proportion of the functional requirements satisfied
- Estimating the total software asset of an organization
- Managing the productivity of software development, operation or maintenance
- Analyzing and monitoring software defect density.

The use of functional size measures has been extensively discussed in the literature. These measures can be used for generating a variety of productivity, financial and quality indicators in different phases of the software development process. Software size has proved to be one of the main effort-and-cost drivers. It is widely accepted that software size is one of the key factors that has the potential to affect the effort and cost of the software projects [3] [5] [6] [7] [8].

An automatic method of counting function points will increase the use of this technique, because automation reduces the cost of counting and the inconsistency of manual counts. An automated function point measurement can become a standard component of the software development and maintenance process. Besides, automatic counting could generate consistent and reliable historical project data for benchmarking. Finally, IT organizations whose manage many software projects can estimate the functional size of their application portfolio more accurately and usually within a short time frame [9].

Monitoring an application's functional size over time is challenging even for those companies that are very well advanced in the use of FP Analysis and that have set processes to count. A functional size estimation method based on input provided by source code analysis can help the process of regularly updating the baseline counts and taking into account changes made during application maintenance and during small application enhancement projects [10].

1.2 Background

Functional Size Measurement (FSM) is defined as the process of measuring functional size. The ISO/IEC 14143-1 standard [4] defines the concepts related to FSM and describes the general principles for applying an FSM method. Functional size methods are hardly automatable and generally require a lengthy and costly process. FSM methods define generic concepts and measurement rules. The setup of a measurement procedure for each input to the measurement process is hence needed. A functional size measurement procedure is defined as a set of operations described explicitly in order to measure according to a specific measurement method [11].

A functional size measurement (FSM) procedure based on the use of software development productivity models allows the management of development costs [12]. The FSM procedure require a rigorous and systematic definition, a clear definition of the of the base functional components (BFC) that contribute to software size, the mapping rules between the BFC and the FSM method and an empirical validation in order to verifying whether the FSM procedure complies with specific performance properties like repeatability, reproducibility and accuracy [13] and specific perception properties like perceived ease of use, perceived usefulness and intention to use [14].

Jacquet and Abran [15] [16] suggest a process model for functional size measurement methods. The model details the steps from the design, its application, the analysis of its measurement results and the exploitation of these results in subsequent prediction models, such as in quality and estimation models. Fetcke et al. [17] proposed a generalized representation for functional size measurement that defines the main concepts used by FSM methods to represent the functional view of a software application.

1.3 Research Objective

This research proposes a functional size measuring procedure for Model-View-Controller (MVC) applications from source code. The research project includes the design, automation and empirical validation of the functional size measuring procedure, according to the ISO/IEC 20926 FPA CPM method. The design and the application of the proposed measurement procedure follows the steps of a process model for software measurement proposed in [15] [16] [17].

Therefore, the objective of this research is: 1) to define a FSM procedure for the automatic measurement of the functional size from MVC source code applications using FPA CMP; and 2) to evaluate the quality of this measurement procedure by looking at its design, application, and the results obtained. To achieve these objectives, a measurement procedure MVC-FPA will be systematically defined and empirically validated and a prototype tool that implements this procedure will be developed.

2. FOUNDATIONS

2.1 Software Size Measurement

It is widely accepted that software size is one of the key factors that has the potential to affect the effort and cost of the software projects [3] [5] [6] [7] [8]. A software application can be measured from two viewpoints: the user viewpoint (problem domain) and the developer viewpoint (system domain) [18]. Functional size measures support the user's perspective and technical size measures support the developer's perspective. There is plenty of research about the relation of these measures, but these views are not necessarily correlated to each other. For example, we might find a system which is very "large" in terms of functionality, but which is relatively small in terms of technical items. Reverse engineering from source code could be a better solution [10].

2.2 Functional Size Measurement (FSM)

Functional Size Measurement (FSM) is defined as the process of measuring functional size. The ISO/IEC 14143-1 standard [4] defines the concepts related to FSM and describes the general principles for applying an FSM method. These series of standards provide a framework in which a new FSM method can be developed, tested and refined. It is composed by the following six parts:

- Part 1: Definition of concepts and requirements for FSM method [4].
- Part 2: Conformance evaluation of software sizing methods to ISO/IEC 14143-1. Provide a framework for conformity evaluation of a candidate FSM method. Describes a process for conformity evaluation [19].
- Part 3: Verification of a functional size measurement method. Establishes a framework for verifying the statements of an FSM Method and for conducting tests requested by the verification about performance properties: repeatability and reproducibility, accuracy, convertibility, discrimination threshold, applicability to Functional Domains [20].
- Part 4: Functional size measurement reference model. It is an input to the evaluation process of an FSM Method [21].
- Part 5: Determination of functional domains for use with functional size measurement. Describes the characteristics of Functional Domains and the procedures by which characteristics of Functional User Requirements (FUR) can be used to determine Functional Domains [22].
- Part 6: provide a guide for use of ISO/IEC 14143 series and related International Standards [23].

After the ISO/IEC 14143 standard series, several functional size measurement (FSM) methods have been proposed to quantify the software functional size based on functional user requirements, including: ISO FSM methods had been proposed ISO/IEC 19761 (COSMIC), ISO/IEC 20926 (IFPUG), ISO/IEC 20968 (Mk II), ISO/IEC 24570 (NESMA), and ISO/IEC 29881 (FiSMA).

2.3 A Process Model for FSM

Jacquet and Abran [15] [16] suggest a process model for functional size measurement methods that details the steps from the design, its application, the analysis of its measurement results and the use of these results in subsequent prediction models, such as quality and estimation models. In the first step, a measurement method is designed (definition of the objective, the concept to be measured, selection of the metamodel, the BFC relationships, the rules to measure, the numerical assignment rules for the metamodel, and all tasks associated with a measurement procedure). In the second step, the measurement method rules are applied to measure the size of software applications (software documentation gathering, construction of the software model and assignment of the numerical rules to the model, automation and data storage). Next, the results provided by the measurement method are presented and audited (description of the measurement process and evaluation of correctness). Finally, the measurement result is exploited in a quantitative or qualitative model (productivity models, effort-estimation models, quality models). Figure 1 shows the detailed steps of the software measurement process model.

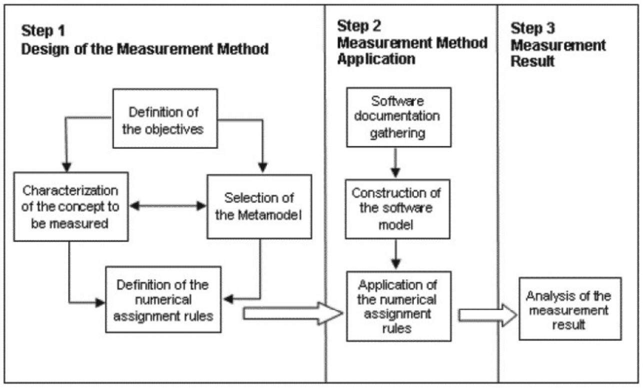

Figure 1. A Process Model Detailed Steps for FSM [14].

2.4 A generalized representation for FSM

Fetcke et al. [17] proposed a generalized representation for functional size measurement that defines the main concepts used by FSM methods to represent the functional view of a software application. The model proposes two levels of abstraction: the first step is the identification (abstract model, concepts such as the user, application, transaction and data) and the second step is the measurement (mapping between functional elements in the abstract model and the numerical rules).

2.5 Function Points Analysis (FPA)

Function point analysis (FPA) [6] [24] [25] was the first proposal for a FSM and it is one of the most used method to measure functional size in the industry. The function point analysis (FPA) method was developed in an attempt to overcome some of the difficulties associated with lines of code as a measure of size [24]. FPA measure the functional size of software from the user perspective [6]. The International Function Point Users Group (IFPUG) FPA manual is one of the most used Functional Size Measurement (FSM) methods in the software industry. This IFPUG FPA is an ISO standard [25]. ISO/IEC 20926:2009 specifies the set of definitions, rules and steps for applying the IFPUG functional size measurement (FSM) method.

In FPA the user requirements are classified and counted in a set of basic functional size components (BFC). These elementary units are called data and transactional functions. They represent data and operations that are relevant to the users. Data functions (DF) are classified into internal logic files (ILF) and external interface files (EIF). Transactional functions are classified into external inputs (EI), external outputs (EO), and external inquires (EQ). Each BFC contributes in the FPA counting that depends on its complexity. Complexity weigh is calculated according to given tables. Unadjusted Function Points is obtained by the summing of all BFC. Details about FPA method can be found in FPA manual [25].

2.6 MVC Architectural Pattern

Model-View-Controller (MVC) was first introduced by Trygve Reenskaug at Xerox Research Center in 1979 [26]. MVC architectural pattern is a well-known and accepted standard for software development [27] [28]. MVC separates an application into three main components: the model, the view, and the controller. MVC helps to decouple data access and business logic from the manner in which it is displayed to the user and helps to create applications that separate the data logic, business logic, and UI logic, while providing a loose coupling between these elements. The user interface (UI) logic belongs to the view component, the input logic belongs to the controller component, and business logic belongs to the model component. This separation helps you manage complexity when an application is built [27] [28]. MVC is a very popular and well accepted practice in developing J2EE and .Net applications.

3. RELATED WORK

Although functional size measurement (FSM) methods have been available for three decades, FSM is not widely adopted in the software community. Several studies searched for problems in the structure and the practice of FSM methods and discussed the challenges for the FSM community [29]. One of the responses to these problems has been the introduction of some degree of formality into functional size measurement. These studies were mainly concerned with objective measurements, automation of the measurement process and the semantic and syntactic elements to define a FSM method without ambiguities but further empirical validation is needed [29].

Several literature surveys have analyzed FSM procedures according to Lother and Dumke proposal [30] to classify them by: FSM method and release, input artifact (requirements, design models, source code, test cases), abstract model, measurement process, mapping and measurement rules, functional domains (management information systems, real time embedded), tool support, validation procedure, empirical data, context (multi-company, local company, consulting company, in-house, academic environment), and others. These studies have identified key remaining challenges in FSM yet to be addressed by further formalization studies [29].

Marín et al. [31], presents a survey of functional FSM procedures in order to provide researchers with an overview of the current state of the functional size measurement procedures based on COSMIC and to provide practitioners with information about the functional size measurement procedures that are available. Ozkan and Demirors [29], identify challenges in FSM that potentially remain to be addressed by further formalization studies.

Currently, we are conducting a systematic literature review for automated FSM procedures. For the first run, we have identified 31 papers for automated FSM procedure proposals. Now, we are executing the snowballing process (backward and forward) in order to identify related papers. Furthermore, in a previous literature survey, we identified 15 proposals for automated function point counting from source code and we have identified 5 Ph.D. thesis on FSM procedures.

Our study is mainly concerned with the definition of a formal functional size measurement procedure without ambiguities, the automation for the measurement process and the empirical validation for the procedure.

3.1 OMG's Automated Function Points (AFP)

Recently, the Object Management Group (OMG) has release the Automated Function Point (AFP) specification [9]. AFP provides a standard for automating function point measure according to the counting guidelines of the International Function Point User Group (IFPUG). Currently, there are some commercial software packages that claim to automatically count function points, but this is the first standard that ensures the repeatability and consistency of the counting technique.

The arrival of an automatic method of counting function points will increase the use of this technique, because automation reduces the cost of counting and inherent inconsistency of manual counts. Automated Function Point measurement can become a standard component of the software development and maintenance process. Automatic counting could generate more consistent and reliable historical project data for benchmarking. AFP specification defines a method for automating the counting of Function Points that is generally consistent with the Function Point Counting Practices

Manual, Release 4.3.1 (IFPUG CPM). It may differ from those in the IFPUG CPM at points where subjective judgments have to be replaced by the rules needed for automation. The specification is applicable to the functional sizing of transaction-oriented software applications, and in particular those with data persistency.

The specification is derived from IFPUG CPM. However, explicit counting rules were specified in this document in order to provide for rigorous automation that may not be in strict conformance with guidance in IFPUG's manual, thus there is no claim of strict conformance with the IFPUG CPM standard. This specification conforms to OMG's Knowledge Discovery Meta-model (KDM) and Structured Metrics Meta-model (SMM) in its specification and representation of the Automated Function Point counting and scoring process. This process ensures automation, consistency and verifiability.

4. RESEARCH OBJECTIVES

The general objective of this research is to propose a functional size measuring procedure for MVC applications from source code. The research project includes the design, automation and empirical validation of the functional size measuring procedure, according to the ISO/IEC 20926 FPA CPM method. The design and application of the proposed measurement procedure follows the steps of a process model for software measurement proposed in [15] [16] [17]. This research analyzes and adapts the FPA CPM method in order to measure the functional size of a MVC software application from its source code.

Our objective written in GQM [32] form is as follows:

Design a FSM procedure
for the purpose of automatic measurement from source code
with respect to their functional size
from the point of view of the researcher
in the context of MVC applications.

Additionally, our objective to evaluate the quality of this measurement procedure can be stated in GQM form as follows:

Analyze FSM procedure
for the purpose of evaluate
with respect to performance and adoption
from the point of view of the researcher
in the context of MVC applications.

To achieve these objectives, a measurement procedure for MVC-FPA will be systematically defined and empirically validated and a prototype tool that implements this procedure will be developed for this purpose. The FSM procedure design is composed of:

1) The definition of a set of mapping rules that allows the base functional components (BFC) of the source code to be identified, for this purpose, the FSM procedure must define the transformation rules from the technical view of the software (source code) to the functional view of the software (base functional components).

2) The definition of a set of measurement rules for obtaining the functional size of MVC applications.

3) The application of both rules sets to several specific case studies.

4) The design validation by means of conformity evaluation with ISO/IEC 20926.

5) The empirical evaluation of the results in terms of accuracy, reproducibility and repeatability.

6) finally, the empirical evaluation of the perceptions of users employing MVC-FPA to measure functional size in terms of perceived ease of use, perceived usefulness and intention to use.

5. RESEARCH METHODS AND EMPIRICAL STUDIES

This research use an empirical software engineering mixed methods approach [33] [34] that combines five main research approaches: design science [35], systematic literature review [36] [37], surveys [38] [39] [40], case studies [33] [34], and experiments [33] [34]. Our Ph.D. project is designed to be carried out in three stages:

- Stage 1: State the Problem.
- Stage 2: Design the Automated FSM Procedure.
- Stage 3: Validate the FSM Procedure.

These stages are explained in depth in the following subsections.

5.1 State the Problem

The objective is to determine the state of the art regarding this problem. To do this, we will carry out the following activities:

- Define the Problem based on the interests of a group of researchers collaborating on a particular family of SE experiments [41], as well as an initial literature review and mapping study [42].
- Review the relevant Literature based on the problem definition.
- Elicit Knowledge by interviewing researchers to learn about the state of practice.
- Conduct a Systematic Literature Review (SLR) on the subject [37] [38]. We analyze FSM procedures for the purpose of characterize with respect to FSM method and release, input artifact, abstract model, measurement process, mapping and measurement rules, functional domains, automation level, tool support, validation procedure, empirical data, and context from the point of view of the researcher in the context of software applications.
- Conduct an industry survey with practitioners to learn about the state of practice in the software industry [38] [39] [40]. We analyze the software size estimation process for the purpose of characterize with respect to performance and adoption from the point of view of project managers in the context of software organizations in Costa Rica. In order to answer research questions, we will execute an electronic questionnaire, face-to-face interviews and statistical analysis of historical project data.

The product of this stage will be an understanding of the current state of the art and the state of practice in FSM.

5.2 Design the Automated FSM Procedure

The objective is to design and automate the functional size measurement procedure. To do this, we will carry out the following activities:

- Design the functional size measurement procedure: the design and the application of the proposed measurement procedure follows the steps of a process model for software measurement proposed in [15] [16] [17].
- Build a software tool application to automate the FSM procedure: we will develop, implement, and validate the tool that automates the process for the FSM procedure [35].

The products of this stage will be the definition of the FSM procedure and a software tool that automates the process defined in the FSM procedure.

5.3 Validate the FSM Procedure

The objective of this stage is to conduct an empirical evaluation for the functional size measurement procedure in terms of accuracy, reproducibility and repeatability and perceived ease of use, perceived usefulness and intention to use. To do this, we will carry out the following activities:

- Conduct a Case Study: we will conduct a case study with practitioners (professional master degree students) to determine the FSM procedure accuracy, reproducibility, and repeatability [14] [33] [34].

- Conduct a Survey: we will conduct a survey with practitioners (professional master degree students) to determine the perceived ease of use, usefulness, and feasibility of use for the FSM procedure [14] [38] [39] [40].

The products of this stage will be the results for the validation of the FSM procedure.

6. THREATS TO VALIDITY

6.1 Empirical Study I (SLR)

The main threats to validity of the study are the following:

- Publication bias: we are conducting a systematic literature review (SLR) where each step has been carried out in pairs, conflicts in the selection process have been solved either by a third party or in consensus meetings, and inclusion and exclusion of studies at each stage has been recorded in order to decrease the bias for the selection process and data extraction.
- Vested interests of the authors: both authors of this research are active researchers on software estimation. We should be aware of the possible impact of our own interests on the analyses. In particular, it is possible that the recommendations we make are affected by our interests and opinions.
- Limited Scope: we will select only formal FSM procedures. The results of our analysis and observations may not be generalized to all FSM methods and automated counting proposals.

6.2 Empirical Study II (Survey)

The main threats to validity of the study are the following:

- Population bias: our sample will be biased to organizations above average size and process maturity level of the industry. The sample organizations are consistent with our target audience and conforms to our research goal of exploring the potential improvement of software estimation methods and process.
- Results: when evaluating the generalizability of the results, we must consider that this is a survey of Costa Rica software organizations. There may be cultural issues that reduce the generalizability of the results.

6.3 Empirical Study III (Case Study)

The main threats to validity of the study are the following:

- Differences among subjects: our experimental subjects will be practitioners of the master degree program which is not a representative sample of the population that would normally use a FSM method. We will select a homogeneous group of subjects.
- Materials and tasks used: we will use a representative requirement specification of a real case in the MIS functional domain.
- Measurement time. The starting and ending measurement time will be self-reported by the subjects.

7. CURRENT STATUS AND PLANNED STEPS

7.1 Current Status of the Research

For the first stage, we have already clearly defined the problem, completed the preliminary literature review of FSM methods and procedures, and elicited knowledge. We have also conducted a replicated study about structure and applicability of function points [43] [44]. We are currently conducting the systematic literature review. The SLR protocol is finished and the first run results are in, and we are in the process of executing the snowballing process (backward and forward) in order to identify relevant related papers. For the industrial survey, we have the survey protocol and the questionnaire ready to be executed.

Regarding the second phase, we have reviewed the process model for software measurement [15] [16] [17] and have started to build a software tool to automate the data BFC for the FSM procedure.

As part of the third stage, we are analyzing several empirical studies for FSM procedures to build the design for own empirical validation studies [12] [13] [14].

7.2 Future Planned Steps

We plan to carry out several parallel activities:

- Complete the Systematic Literature Review.
- Complete the Industrial Survey.
- Document the design of the FSM procedure and implement the development of the tool for measurement.
- Start planning the empirical studies for FSM procedure validation.

8. ACKNOWLEDGMENTS

Our research was supported by the Costa Rican Ministry of Science Technology and Telecommunications (MICITT). Our thanks as well to LACCIR, and ESE Group at UCR and COPPE/UFRJ.

9. REFERENCES

[1] Peixoto, C. E. L., Audy, J. L. N., & Prikladnicki, R. (2010, May). The importance of the use of an estimation process. In *Proceedings of the 2010 ICSE Workshop on Software Development Governance* (pp. 13-17). ACM.

[2] Molokken, K., & Jorgensen, M. (2003, October). A review of software surveys on software effort estimation. In *Empirical Software Engineering*, 2003. ISESE 2003. Proceedings. 2003 International Symposium on (pp. 223-230). IEEE.

[3] Boehm, B. W. (1981). *Software engineering economics*.

[4] ISO. (2007). *ISO/IEC 14143-1- Information Technology - Software measurement - Functional Size Measurement. Part 1: Definition of Concepts*.

[5] Kitchenham, B. (1993) Using Function Points for Software Cost Estimation – Some Empirical Results. *10th Annual Conference of Software Metrics and Quality Assurance in Industry, Amsterdam*, Netherlands.

[6] Albrecht, A. J., & Gaffney, J. E. (1983). Software function, source lines of code, and development effort prediction: a software science validation. *Software Engineering, IEEE Transactions on*, (6), 639-648.

[7] Jones, T. C. (1998). *Estimating software costs*. McGraw-Hill, Inc.

[8] Kemerer, C. F. (1987). An empirical validation of software cost estimation models. *Communications of the ACM*, *30*(5), 416-429.

[9] OMG. (2014). *Automated Function Points. Version 1.0.*

[10] Ellafi, R., & Meli, R. A Source Code Analysis-based Function Point Estimation Method integrated with a Logic Driven Estimation Method.

[11] BIPM, I., IFCC, I., ISO, I., & IUPAP, O. (2008). International Vocabulary of Metrology–Basic and General Concepts and Associated Terms, 2008. *JCGM*,*200*, 99-12.

[12] Soubra, H. (2011). *Automatisation de la mesure fonctionnelle COSMIC-ISO 19761 des logiciels temps-réel embarqués, en se basant sur leurs spécifications fonctionnelles* (Doctoral dissertation, École de technologie supérieure).

[13] Condori-Fernández, N., Abrahão, S., & Pastor, O. (2007). On the estimation of the functional size of software from requirements specifications. *Journal of Computer Science and Technology*, *22*(3), 358-370.

[14] Abrahao, S. M., & Director-Lopez, O. P. (2004). On the functional size measurement of object-oriented conceptual schemas: *design and evaluation issues*. Universidad Politecnica de Valencia (Spain).

[15] A Abran, A., & Jacquet, J. P. (1999). A structured analysis of the new ISO standard on functional size measurement-definition of concepts. In Software Engineering Standards, 1999. *Proceedings. Fourth IEEE International Symposium and Forum on* (pp. 230-241). IEEE.

[16] Jacquet, J. P., & Abran, A. (1997, June). From software metrics to software measurement methods: a process model. In *Software Engineering Standards Symposium and Forum, 1997. Emerging International Standards. ISESS 97, Third IEEE International* (pp. 128-135). IEEE.

[17] Fetcke, T., Abran, A., Dumke, R. (2001) A Generalized Representation for Selected Functional Size Measurement Methods *In Current Trends in Software Measurement*. In: Dumke, R., Abran, A. (eds.) Shaker, (pp. 1–25).

[18] Meli, R. (2000, October). Functional and technical software measurement: conflict or integration. *In 3rd. European Software Measurement Conference* (FESMA'00), Madrid, Spain.

[19] ISO. (2011). *ISO/IEC 14143-2:2011 Information technology -- Software measurement -- Functional size measurement -- Part 2: Conformity evaluation of software size measurement methods to ISO/IEC 14143-1*.

[20] ISO. (2003). *ISO/IEC TR 14143-3:2003 Information technology -- Software measurement -- Functional size measurement -- Part 3: Verification of functional size measurement methods*.

[21] ISO. (2002). *ISO/IEC TR 14143-4:2002 Information technology -- Software measurement -- Functional size measurement -- Part 4: Reference model*.

[22] ISO. (2004*). ISO/IEC TR 14143-5:2004 Information technology -- Software measurement -- Functional size measurement -- Part 5: Determination of functional domains for use with functional size measurement.*

[23] ISO. (2012). *ISO/IEC 14143-6:2012 Information technology -- Software measurement -- Functional size measurement -- Part 6: Guide for use of ISO/IEC 14143 series and related International Standards.*

[24] Albrecht, A. J. (1979, October). Measuring application development productivity. *In Proceedings of the Joint SHARE/GUIDE/IBM Application Development Symposium* (Vol. 10, pp. 83-92). Monterey, CA: SHARE Inc. and GUIDE International Corp.

[25] ISO. (2009). *ISO/IEC 20926, Software and systems engineering - Software measurement – IFPUG functional size measurement method.*

[26] Reenskaug, T. (1979). MVC. Retrieved 04 01, 2014, from XEROX PARC 1978-79: http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html.

[27] Team, M. A. (2009). MVC Overview. Retrieved 04 01, 2014, from ASP.NET: http://www.asp.net/mvc/tutorials/older-versions/overview/asp-net-mvc-overview.

[28] Eckstein, R. (2007). Java SE Application Design With MVC. Retrieved 04 01, 2014, from ORACLE: http://www.oracle.com/technetwork/articles/javase/index-142890.html.

[29] Ozkan, B., & Demirors, O. (2009). Formalization Studies in Functional Size Measurement: How Do They Help?. *In Software Process and Product Measurement* (pp. 197-211). Springer Berlin Heidelberg.

[30] Lother, M., & Dumke, R. (2001). Points Metrics-Comparison and Analysis. *InIWSM* (Vol. 1, p. 228ņ267).

[31] Marín, B., Giachetti, G., & Pastor, O. (2008). Measurement of functional size in conceptual models: A survey of measurement procedures based on COSMIC. *In Software Process and Product Measurement* (pp. 170-183). Springer Berlin Heidelberg.

[32] Basili, V. R., & Rombach, H. D. (1988). The TAME project: Towards improvement-oriented software environments. *Software Engineering, IEEE Transactions on,* 14(6), 758-773.

[33] Shull, F., Singer, J., & Sjøberg, D. I. (2008). *Guide to advanced empirical software engineering*. Springer.

[34] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in software engineering*. Springer.

[35] Von Alan, R. H., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, 28(1), 75-105.

[36] Biolchini, J., Mian, P. G., Natali, A. C. C., & Travassos, G. H. (2005). Systematic review in software engineering. *System Engineering and Computer Science Department COPPE/UFRJ*, Technical Report ES, 679(05).

[37] Kitchenham, B. A., & Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering*.

[38] Travassos G. H., S. L. Peeger, V. R. Basili. (2004) Experimental Software Engineering: an Introduction. *1st Experimental Software Engineering Latin American Workshop - ESELAW*. 2004.

[39] Kasunic, M. (2005). Designing an effective survey (No. CMU/SEI-2005-HB-004). CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.

[40] Pfleeger, S. L., & Kitchenham, B. A. (2001). Principles of survey research: part 1: turning lemons into lemonade. *ACM SIGSOFT Software Engineering Notes*, 26(6), 16-18.

[41] Juristo, N., & Vegas, S. (2009, October). Using differences among replications of software engineering experiments to gain knowledge. In *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement* (pp. 356-366). IEEE Computer Society.

[42] Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008, June). Systematic mapping studies in software engineering. *In 12th International Conference on Evaluation and Assessment in Software Engineering* (Vol. 17, p. 1).

[43] Jeffery, R., & Stathis, J. (1996). Function point sizing: structure, validity and applicability. *Empirical Software Engineering*, *1*(1), 11-30.

[44] Carver, J. C. (2010, May). Towards reporting guidelines for experimental replications: a proposal. In *1st International Workshop on Replication in Empirical Software Engineering*.