Journal of Theoretical and Applied Information Technology
31st December 2015. Vol.82. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

INVESTIGATING THE RELATIONSHIP BETWEEN SOFTWARE DEFECT DENSITY AND COST ESTIMATION DRIVERS: AN EMPIRICAL STUDY

[1]FADI WEDYAN, [1]HANI BANI-SALAMEH, [2]WAJEEHA AL-AJLOUNI, [3]SHIRIN AL-MANAI
Department of Software Engineering, Hashemite University, Zarqa, 13315 Jordan
E-mail: [1]{fadi.wedyan, hani}@hu.edu.jo, [2]wajeehaajlouni@itc.hu.edu.jo, [3]Sherine_almnay@yahoo.com

ABSTRACT

This paper reports the results of an empirical study, where some hypotheses about the relationship between software quality, expressed in terms of defect density, and cost estimation drivers (cost factors) are assessed. The study is performed using three cost factors, these are: work effort, project size, measured in terms of function points, and average assigned work per team member. The study is performed using the ISBSG repository data set release 12. ISBSG dataset which contains 6006 completed project. For statistical analyses, Regression Analysis and stepwise Analysis of Variance (ANOVA) are used. Our results show that there is a significant negative impact of the project size and work effort on the defect density. On the other hand, the average assigned work per team member has a significant positive impact on defect density. These results suggest that while assigning resources to a project, these quality factors should be considered in order to decrease the defect density, and thus, the maintenance cost of the product.

Keywords: *Cost Estimation drivers, Defect Density, Empirical Study, ISBSG Repository, Software Quality*

1. INTRODUCTION

Recently, research has been directed to the estimation of software development attribute as effort estimation and cost estimation using the data collected from completed projects. Since the International Software Benchmarking Standards Group [1] (ISBSG) began at 1997, it has provided researcher and the information technology industry with the largest multi-organization software engineering repository free access. The ISBSG has established and maintained a database of software projects data that can be used by software project managers, IT managers, and IT customer business managers to make decision and be more aware of critical issues on software development like Estimate software project size, effort, cost, duration, benchmark project performance and plan software development infrastructure. The ISBSG database became an important source of data for empirical software engineering research. As a result, the ISBSG data set has been respected in research area and cited by extensive research**.**

Maintaining quality for software developed for big industry is a challenging task and gets more complicated when adding the economical and technical dimensions. The main aim of industry management is to have a successfully completed project within the stated requirement of best quality, stated duration, and at minimum cost possible to allocate resources in the best economize way. In last few years, under high competition between software development companies, the organizations management became more demanding to get not only a high quality project but with minimum defects in the long run and less costly. A report published in 2002 for the National Institute of Standards and Technology (NIST), Department of Commerce in USA, found that software defects are widespread and harmful to software that they Consequent costs on the economy of the united states is measured in tens of billions annually [2].

Understanding the cost of quality is extremely important in establishing a quality management strategy. The cost of the quality is related to the prevention, finding, and correcting defective software [3]. the Prevention-Appraisal-Failure (PAF) model is a widely used model for estimating quality cost [4][5][6]. The model was first derived from the industry. The PAF model provides a conceptual view of the cost of quality and quality perspectives and it confirms that the quality enhances along with the failure cost decrease as a result of increasing in appraisal cost plus prevention cost. The model provides suggestions for

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

investments in appraisal and prevention activities for material, machine and labor to minimize total cost of quality and to accomplish a satisfactory level of quality [7].

In this paper, we present the results of an empirical study that aims to investigate the relationship between three quality factors and defect density in a software product. We choose the following quality factors: work effort, project size (in terms of function points), and average work assigned to each work team member. The study is performed using data provided by the ISBSG. We choose quality factors that directly affect the development cost, and therefore, can have an influence on the policy of assigning resources to a project and the expected quality. Specifically, the study reported in this paper aims to answer the following research questions:

Q1: Is there a relationship between the defect density and work effort?

Q2: Is there a relationship between the defect density and the project size?

Q3: Is there a relationship between the defect density and the average assigned work per team member?

Our results show that work effort and project size have a significant negative effect on the defect density. On the other hand, the average assigned work per team member has a significant positive impact. These relations should be taken into account when assigning resources to a project in order to reduce the maintenance cost.

The rest of paper is organized as follows: Section 2 presents the related works on using ISBSG dataset in empirical software quality studies. Section 3 describes the main characteristics of the ISBSG repository, release 12. Section 4 describes the cost estimation drivers and measurements used in the study. Section 5 describes how data is collected and prepared for the empirical study. The results of the empirical study are given in Section 6. Threats to validity are given in Section 7. Finally, conclusions and future work are discussed in Section 8.

2. RELATED WORK

Oligny and abranin [8] compared duration estimates obtained from simple COCOMO equations to an empirical model derived from the ISBSG data set (using Release 3 of the ISBSG dataset). They conclude that COCOMO estimation for duration are "Optimistic" compared with empirical model estimates in ISBSG.

Using release 4 of the ISBSG dataset, Lokan [9] empirically explored the validity of the Function Point measure (FP) analyzing relationships among the five elements of Function Point that basic elements of this measure. The results show strong correlations in new development projects (using 4 GLs).

Jeffery et al. [10] compared modeling techniques, namely Ordinary Least Squares Regression (OLS), stepwise Analysis of Variance (stepwise ANOVA), Regression Trees (CART), and Analogy-based estimation, in terms of their estimation accuracy using Release 5 of the ISBSG dataset as well as another data set from an Australian company called Megatec (within company dataset). Their results show that OLS regression performed as well as Analogy-based estimation using company modeling data. when the multi-company data is used, the OLS regression model provided more accurate results.

Jeffery et al. [11] analyzed Release 6 of ISBSG dataset to investigate how modeling and estimation techniques accuracy differ from each other and assessment of the feasibility of using multi-organizational data compared to the benefits from company-specific data. Their results show that the estimation is more accurate when using the company-specific data.

Buglione and Abran [12] used ISBSG dataset release 10 to illustrate various aspects of the multi-dimensional QEST-LIME model that calculate software performance to gain the business goals about quality economic software. They concluded that not much additional effort needed or costs required to gather more data in order to derive a performance value based on data analyzed in ISBSG repository.

A more recent study was performed using release 10 of the ISBSG dataset in order to carry out experimental result of varies predicting relation between software attributes by Wang et al. [13]. Their results show that the quality level of software can be predicted using MCLP model.

Our contributions of this work is to explore factors that directly affect the development cost (as the recourses, team size, project size, effort consumed, complexity, functionality, etc.) and factors of quality level, and then investigate the relationship between development cost and quality

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

based on the ISBSG repository data set as an experimental study.

3. ISBSG OVERVIEW

ISBSG- The International Software Benchmarking Standards Group- is non-profit organization that aims to "develop the profession of software measurement by establishing a common vocabulary and understanding of terms" [1]. The ISBSG maintains a huge database of completed software projects from different organizations and provided research data on several topics; function point's structure, project duration, and cost estimation for instance. The latest Release (12 February 2013) of ISBSG repository consists of 6006 projects from over 25 countries around the world between 1989 and 2012, with 84% of the projects being 8 years old [1].

The ISBSG dataset has many uses including: project benchmarking, best practice networking , summary analyses, estimation of own future software development projects, assist in evaluating the benefits of changing hardware or software environment, and software development and engineering research for academic research with the objective of improvement in different IT practices [1].

3.1 Dataset Characteristics

Projects in release 12 of the ISBSG dataset have 18 first level attributes and each of them is subdivided into sub attributes with a total of 100 for all attributes. The first level attributes are: Rating, Software Age, Major Grouping, Sizing, Effort, Productivity, Other Metrics, Schedule, Quality Delivered, Grouping Attributes, Architecture, Documents & Techniques, in 1st Platform, 2st Platform, Project Attributes, Product Attributes, Effort Attributes, Size Attributes, and Other Size. The dataset include a mixture of new developments (53%), enhancements (41%), and re-developed projects (6%) that are characterized by many variables collected in conformance to international standards.

The projects cover a wide range of applications, implementation languages, platforms, and development techniques and tools. The projects are cover the following sectors: Communication (15%), Manufacturing (14%), Banking and Financial (14%), Insurance (12%), Government (10%), Electronics & Computers (3%), Service Industry (3%), Wholesale & Retail, Utilities, Professional Services, Construction, Medical & Health Care, Defense, Education, Mining, Logistics (2%) or and less, and (21%) of repository project sector is unknown.

3.2 Variables Definition and Values

The variables relevant to our study of ISBSG dataset are:

- Data Quality Rating: this field shows an ISBSG rating code. The rating is determined by the quality personals after reviewing the projects data. It express data integrity of project from rating (A): high integrity, to (B): has a very limited credibility [1].
- Count Approach: is the description of the technique that used to size the project. Most of the projects in the ISBSG data repository are measured using FSM method based on functional size e.g. IFPUG, NESMA, and COSMIC-FFP.
- Normalized Work Effort: measures the effort that is used in full life cycle of the project.
- Summary Work Effort: measures the total effort recorded against projects.
- Quality (Total Defects Delivered): the number of defects detected in the process in that particular effort breakdown or found within the first month of use of the software after implementation.
- Max. Team Size: estimation of the maximum number of people that have worked as a team on the project. It is estimated by average team size as total hours of effort for the project divided by the project duration in calendar months, divided by the number of hours in a person- month.

4. SOFTWARE COST ESTIMATION AND QUALITY FACTORS

In this section, we introduce the cost estimation drivers and quality factors used in this study.

4.1 Cost Estimation Factors

The cost of software projects is driven by many factors,. As described by Briand et al. [14], these factors have either direct or interaction impact on the project cost. A direct relationship means that the factor directly increases or decreases cost overhead neglect of the values of the other factors. An interaction relationship means that the manner in

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

which a factor affects cost overhead depends on the value(s) of one or more other factors.

In addition, one of most popular parametric cost estimation models COCOMO computes software development effort as function of program size and a set of "cost drivers" that include subjective assessment of product, hardware, personnel and project attributes that confirms the cost estimation in project will depending on size and effort of project [15]. Team member personalities, their experience and knowledge can all impact software development cost.

Effort and cost are closely related. Effort is often measured in person months of the programmers, analysts and project managers. This effort estimate can be converted into a dollar cost by using the following equation [15]:

Cost = Estimated Effort Required * (Average Salary per Staff Person)

4.2 Quality Factors

Based on Cost of Quality (CoQ model) classification that was first time presented by Feigenbaum in 1956, the Prevention-Appraisal-Failure (PAF) model is the one most commonly applied. The scheme of the CoQ was used as a framework metrics to measure the level of quality. Cost of Quality can be categorized into [16]:

- Prevention costs: refers to the cost consumed to reduce future appraisal and failure costs as a protection of failure. Prevention costs include costs for the review and verification of design, cost of quality training and auditing [6].
- Appraisal costs: refers to the cost of checking whether a product meets its quality requirements [6].
- Failure costs; refers to the cost of poor quality. Failure cost is further divided into internal and external failure costs. Internal failure costs arise when a fault is detected in the software or a related work product during verification and validation activities. External cost refers to the financial consequences of failure occurrences during operational software usage [6].

Defect density measures the percentage of faults in a software module and is computed by dividing the total number of defects by the size of the software [17]. The cost of finding defects in during operational stage is not only limited to the cost correcting and maintain the software, but also includes the cost of consequences of the failure which might be catastrophic and consume valuable resources. The additional effort and resources consumed on detection and correction the defect before and after release called rework cost [6].

In the PAF model, it is hypothesized that there is an opposite relationship between the level of quality and failure cost. The model also assumes that the failure costs will reduce as a result of the investment in prevention and appraisal defect to improve the quality level [18][19].

5. EMPIRICAL STUDY

In this section, we describe the steps performed in preparing the empirical study. We start by demonstrating how the projects were selected, then we describe data classification and predication approach. Finally, we describe the results of the study.

5.1 Data Preparation

Having 6006 projects in the ISBSG, filtering is necessary to select projects that better fit our study. In the ISBSG repository. Projects in the ISBSG repository are rated depending on the quality of the metadata given about a project. Rating varies from very good (rate A), to Unreliable (rate D). In addition, some of the projects have missing data (i.e., not all attributes values are given).

In order to select suitable projects for our study, we applied the following steps in the given order:

- Select project with high data Quality rated as A or B (i.e., data is sound and there is no threat to data integrity). Projects having either C or D Data Quality Rating are discarded.
- Select the project effort (Summary work effort as a sub attribute) that have "Count Approach": COSMIC FFP - The Common Software Measurement International Consortium.
- Select projects with available value for project Defect Density “Quality”.
- Select projects with project Size measured in UFP (Unadjusted Function Points).
- Select projects with available value for Average Team Size.

The result of the filtering approach is given in Table 1. After applying step 1, 448 projects were excluded and 5558 remain (shown in columns 4 and 5 in the table, respectively). Applying step 2, only 393 projects remain. The defect density was available for 82 projects out of the remaining 393

Journal of Theoretical and Applied Information Technology
31st December 2015. Vol.82. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

projects (third row in the table). Finally, 35 project remain after applying steps 4 and 5.

Effort is usually measured in Person-Month. The summary work effort provides the total effort in hours recorded against the project, and therefore, it can be used to measure human resource required. Using Average Team Size (sub attribute of effort attribute in data set) and effort attribute calculated in measurement method like IFPUG standard for functional size measurement or full function points (COSMIC FFP).

Table 1: Filtering Steps of the ISBSG Dataset

Step	Attributes	Filter	Project Excluded	Remaining Projects
1	Quality Rating	A || B	448	5558
2	Summary Work Effort in "Count Approach"	= COSMIC	5165	393
3	Defect Density	≠ Empty	311	82
4	Project Size	= UFP	0	82
5	Average Team Size	≠ Empty	47	35
Project Remaining				***35***

Because of different functional size measurements can produce different results, it should be taken into account same method of measurement to present the relation in clear way. The COSMIC is Functional Size Measurement Method that is widely used by various type of organization because it provides very good estimates close to the actual effort spent in each project over IFPUG method [20].

5.2 Data Classification and Prediction

The projects selected for the study are analyzed to investigate the relationship between the Cost factors and Quality attribute "Defect Density" using liner regression. The selected projects are categorized into types, these are: new development projects and enhancements project. Both types are analyzed using Linear Regression Analysis and Analysis of Variance (ANOVA) - Single Factor ANOVA. Statistical analyses is performed using Microsoft Office Excel 2011 tools. In the statistical analysis, defect density is considered as a dependent variable and the cost factors as independent.

6. EXPERIMENTAL RESULTS

In this section, we report the results of the empirical study we performed in order to answer the research questions:

Q1: Is there a relationship between the defect density and work effort?

Figure 1 shows the relationship obtained between Defects Density and Normalized Work Effort attributes using Linear Regression Analysis. As shown in the figure, there is a low adjusted R square value (0.05) and inverse confident correlation with value (-0.22). This means the increasing in work effort in project can affects the defect founded negatively.

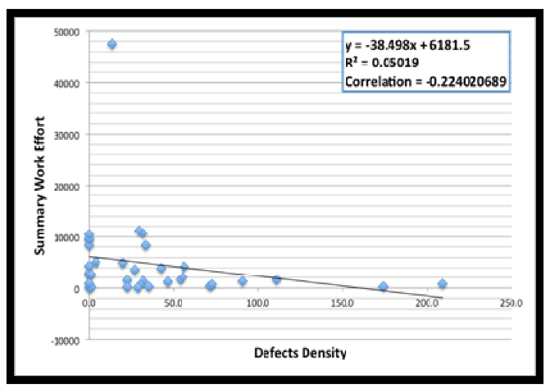

Figure 1: Defects Density Vs. Summary Work Effort

Q2: Is there a relationship between the defect density and the project size?

Figure 2 shows the relationship between Defects Density and Project Size attributes using Linear Regression Analysis. As shown in the figure, there is a low adjusted R square value (0.046) and Inverse Confident Correlation with value (-0.21). This means that increasing the project functional size affects the defect found negatively.

Journal of Theoretical and Applied Information Technology
31st December 2015. Vol.82. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Q3: Is there a relationship between the defect density and the average assigned work per team member?

Figure 3 shows a relationship between Defects Density and Average Assigned worked per Team member (Functional Size/Avg. Team Size) using Linear Regression Analysis. As the figure shows, there is a low adjusted R square value (0.02) and Inverse Confident Correlation with value (0.12). This means that assigning smaller work load to a team member has a positive impact on the defect density.

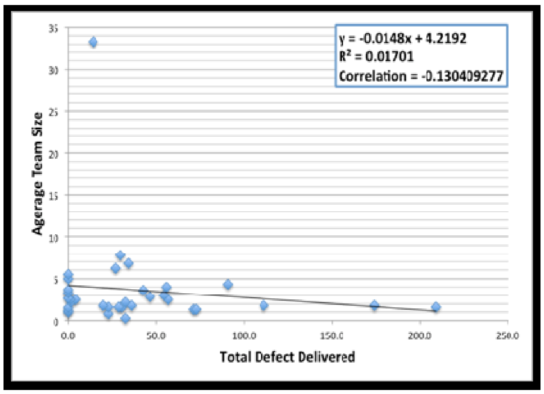

Figure 2: Defects Density Vs. Project Size

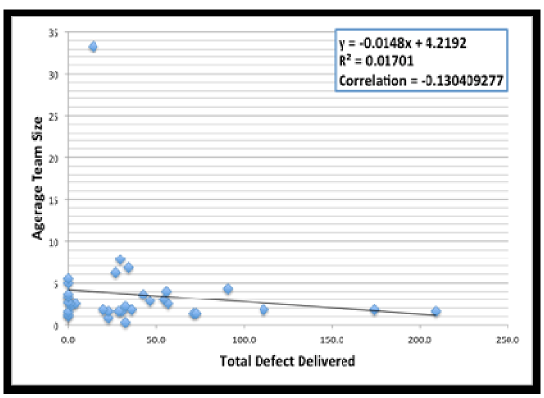

Figure 3: Defects Density Vs. Average Assigned Work Per Team Member

Table 2 summarizes the results of three relationships that performed to observe the effect of different factors on the project quality. All of the tested factors show a weak relationship with defect density attributes with low value for R2, but although of that, still this result are significance and indicate there is a relationship between the cost estimation drivers (Effort, Project Size and work load per team member) and projects defects density. In this empirical study, we performed 3 analyses; a one-way Analysis of Variance (ANOVA) to observe the effects of quality attribute (as a count total of defect density delivered in first month of using the project) on the used quality attributes.

Table 2: Summary of Statistics

Relationship	R2	Correlation
Defects Density Vs. Summary Work Effort	0.05	-0.22
Defects Density Vs. Project Size	0.05	-0.21
Defects Density Vs. Average team size	0.02	0.12

7. THREATS TO VALIDTY

Empirical studies are known to have limitations [21]. We identify three types of threats to the validity of this empirical study: construct validity, internal validity and external validity. Construct validity refers to the meaningfulness of measurements [22]. In our study, the inherent subjectivity of counting functional size measurement methods is a clear construct validity threat [23]. However, since we used high quality projects from the ISBSG repository, the effect of this threat can be minimized.

Internal validity is concerned with cause and effect relationships. That is, to what extent changes in dependent variables can be confirmed to be caused by changes in independent variables [24]. In our study, we identify two threat to the internal validity. First, It might be that the increase or decrease in defect density is affected by other factors, other than the quality factors considered in the study (e.g., the testing approach, the skills of the developers). Second, there are many factors that can affect the defect density relationship with quality factors due to confounding effect [34]. However, to minimize these threat, we used a large number of projects from a large database, therefore, the internal validity is endures to be fairly good.

External validity refers to how well the results of a study can be generalized outside the scope of the study [24]. The external validity of this study is limited to the use of projects from one repository. There is no evidence that the results can be extended or generalized to other projects. The effect of this factor is limited since the projects in the studied sample come from different domains, different ages, with varied number of developers and with variant sizes and different attributes. In

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

future work, we plan to study additional projects from other repositories order to confirm our results.

8. CONCLUSIONS AND FUTURE WORK

Minimizing the defect-density of a software has very important impacts including reducing the maintenance cost and increasing software reliability and robustness. We empirically evaluated the effect of three cost factors on the defect-density, these are: work effort, project size, measured in terms of function points, and average assigned work per team member. Our results suggest that effort project size negatively impact the defect density, while minimizing the work load of a team member results in less defect density. These factors should be taken into consideration while allocating resources for a project.

In future work, we plan to study more factors that might affect the defect-density relationship with the quality factors including: programming languages used to implement the software, the maturity level of the software, and the software domain. We also plan to use projects from different repositories in order to confirm the obtained results.

REFRENCES:

[1] ISBSG, International Software Benchmarking Standards Group, http://www.isbsg.org.

[2] G. Tassey, "The economic impacts of inadequate infrastructure for software testing", *National Institute of Standards and Technology* , RTI Project 7007.011, 2002.

[3] S. Wagner, "Towards a software quality economics for defect-detection techniques", *The 29th Annual IEEE/NASA Software Engineering Workshop*, 2005, pp. 256-274.

[4] S. T. Knox, "Modeling the costs of software quality". *Digital Technical Journal*, Vol. 5, No. 4, 1993, pp 9–16.

[5] H. Krasner, "Using the Cost of Quality Approach for Software", *CrossTalk. The Journal of Defense Software Engineering*, Vol. 11, No. 11, 1998, pp. 6-11.

[6] S. A. Slaughter, D. E. Harter, and M. S. Krishnan, "Evaluating the Cost of Software Quality", *Communications of the ACM*, Vo. 41, No. 8, 1998, pp. 67–73.

[7] K. Vincent, S. Sakesun, and G. E. Norman, “The relationship between quality and quality cost for a manufacturing company”, *International Journal of Quality & Reliability Management*, Vol. 21, No. 3, 2004, pp. 277-290.

[8] S. Oligny, P. Bourque, and A. Abran, ‘An empirical assessment of project duration models in software engineering", *Proceedings of the Eighth European Software Control and Metrics conference (ESCOM)*, 1997.

[9] C. J. Lokan, "An Empirical Study of the Correlations between Function Point elements", *Proceedings of the Sixth International Software Metrics Symposium*, Los Alamitos CA, 1999, pp. 200-206.

[10] R. Jeffery, M. Ruhe, and I. Wieczorek, "A Comparative Study of Two Software Development Cost Modeling Techniques using Multi-organizational and Company-specific Data", *Information and Software Technology*, Vol. 42, Issue 14, 2000, pp. 1009-1016.

[11] R. Jeffery, M. Ruhe, and I. Wieczorek, "Using Public Domain Metrics to Estimate Software Development Effort", *Proceedings of the Seventh International Software Metrics Symposium*, 2001, pp. 16–27.

[12] L. Buglione and A. Abran, "Performance Calculation and Benchmarking Using the ISBSG Release 10 Data Repository", *Proceedings of the fifth Software Measurement European Forum*, 2008, pp. 175–192.

[13] X. Wang, L. Zhang, Y. Zhang, and Y. Shi, "A Knowledge Discovery Case Study of Software Quality Prediction: ISBSG Database", *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT),* 2010, pp. 219-222.

[14] L. C. Briand, K. El Emam, and F. Bomarius, "COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking and Risk Assessment", *Proceedings of the 20th International Conference on Software Engineering*, 1998, pp. 390-399.

[15] H. Leung, and Z. Fan, "Software Cost Estimation, Handbook of Software Engineering", *Hong Kong Polytechnic University*, 2002.

[16] L. M. Karg, M. Grottke and A. Beckhaus, "A Systematic Literature Review of Software Quality Cost Research", *Journal of Systems and Software*, Vol. 84, No. 3, 2011, pp. 415–27.

[17] C. Rahmani and D. Khazanchi, "A study on defect density of open source software", *9th IEEE/ACIS International Conference on Computer and Information Science*, 2010, pp. 679-683.

[18] K. Vincent, S. Sakesun, and G. E. Norman, “The relationship between quality and quality cost for a manufacturing company”

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

International Journal of Quality & Reliability Management, Vol. 21 (3), 2004, pp. 277-290.

[19] L. P. Carr, and L. A. Ponoemon, "The Behavior of Quality Costs: Classifying the Confusion", *Journal of Cost Management Practices*, 1994, Vol. 12, pp. 26-34.

[20] C. Symons, "A Comparison of the Key Differences between the IFPUG and COSMIC Functional Size Measurement Methods", *Common Software Measurement International Consortium*, 2011.

[21] B. A. Kitchenham, S. L. Pfleeger, L. M. Pickard P. W. Jones D. C. Hoaglin K. El Emam, and J. Rosenberg. "Preliminary guidelines for empirical research in software engineering", *IEEE Trans. Software Engineering*, Vol. 28, Aug 2002, pp. 721– 734.

[22] F. Kerlinger. "Foundations of Behavioral Research", *Third Edition. Harcourt Brace Jovaonvich College Publishers*, Orlando, Florida, 1986.

[23] L. Lavazza, and R. Meli, "An Evaluation of Simple Function Point as a Replacement of IFPUG Function Point", *the International Conference on Software Process and Product Measurement (IWSM-MENSURA)*, 2014, pp. 196-206.

[24] N. Fenton, S. Pfleeger, "Software Metrics: A Rigorous and Practical Approach", *second edition, PWS Publishing Co.*, Boston, MA, 1999.

[25] K. El Emam, S. Benlarbi, N. Goel, and S.N. Rai, "The Confounding Effect of Class Size on the Validity of Object-Oriented Metrics," *IEEE Trans. Software Eng*ineering, vol. 27, no. 7, July 2001, pp. 630-650.