

# VAF FACTOR INFLUENCE ON THE ACCURACY OF THE EFFORT ESTIMATION PROVIDED BY MODIFIED FUNCTION POINTS METHODS

Zdenka Prokopova, Petr Silhavy & Radek Silhavy



**This Publication has to be referred as:** Prokopova, Z[denka]; Silhavy, P[etr] & Silhavy, R[adek] (2018). VAF Factor Influence on the Accuracy of the Effort Estimation Provided by Modified Function Points Methods, Proceedings of the 29th DAAAM International Symposium, pp.0076-0084, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-20-4, ISSN 1726-9679, Vienna, Austria  
DOI: 10.2507/29th.daaam.proceedings.011

## Abstract

The paper presents the Function Points (FP) method, which can be used for a preliminary effort estimation using limited information. Despite the potential for early use of the effort estimation, FP provides meaningful and relatively accurate results. The authors aimed to design Modified Function Points (MFP) methods based on regression model and analyze the influence of Value Adjustment Factor (VAF) on the estimation accuracy of the development effort. For research purposes was selected the ISBSG dataset. Subsequently, the original dataset was reduced according to data requirements and divided into two parts the training and the testing section (in ratio 2:1). The presented analysis was processed as a preparatory phase for further research in this area. Matlab toolboxes were used for the design and verification of discussed algorithms.

**Keywords:** accuracy; function points; software estimation; VAF; development effort

## 1. Introduction

Software development projects planning involves estimation of development time. In many cases, the development time estimation is based on development effort estimation. The effort is measured in person-hours and is based on software size estimation. The software size estimation can be one of the known methods [1], [2], [3], [4]. In this paper, an FP method is involved. The effort estimation in FP is based on the proper setting of the productive factor [5]. The estimation process of the product costs is discussed in [6], [7]. The FP estimation process is a time-consuming, as it requires detailed elicitation of the project requirements and maps them into predefined components. Several simplified methods have been proposed to minimize the effort required to calculate FP [8]. Of course, also new advanced methods have been developed to estimate the size of software projects [9]. Impact of functional and non-functional requirements was studied in [10]; authors discuss a functional size and relationship to complexity. In [11] authors discuss a complexity matrix for FP' variables. Variable influence on estimation accuracy was studied in [12], in which the most influential variables were evaluated. In [13] authors present a case study of using FP in the industry. They confirm a practical impact on the software industry.

In [14] the study of FP application to web-based projects. In [15] authors perform analysis of general system characteristics (GSC), which a baseline for VAF factor is calculating. They denoted that VAF improves accuracy in 50% of cases. This finding was confirmed in [16]. The VAF is one of the variables which relevance is studied in the software effort estimation research. Correlation between effort and FP (including unique FP variables) provided in [16], [17], [18]. The factors influence was studied in [12]. The VAF is usually reported as a factor which brings less influence to accuracy than other factors.

This paper is divided into six sections. In Section 2 a problem statement is outlined. In Section 3, used methods are described. In Section 4 an experiment setup, VAF analysis, and tested method description are presented. In Section 5 research results are discussed. Finally, Section 6 is a conclusion.

## 2. Problem statement

In the presented paper, the authors focused on a comparison of the accuracy of development effort estimation, when three approaches are evaluated. The models will be evaluated on observations, which are described with and without VAF factor. The effect of VAF on estimation accuracy will be studied and described.

The tested methods will be as follows:

- FP method, when mean of PDR is applied
- Modified FP method, in which a linear regression model (LRM) is applied
- Modified FP method, in which stepwise regression (SRM) was tested

The research question answered by this study are as follows:

- RQ1: Does VAF factor improve estimation accuracy?
- RQ2: Are all tested methods equivalent in the sense of estimation accuracy?

## 3. Method used

### 3.1. Function Points Method

The Function Points Method is standardized and maintained by the IFPUG - International User Group of Function Points. The IFPUG was founded in Toronto in 1986. The IFPUG has established that the methodology will distinguish five function types of components (measurement parameters) [19], [20], [21]:

Transaction function types:

- External inputs (EI) – these are external input functions with their logical data groups and elements. External inputs are elementary processes, which include, for example, screens, forms, dialogs, or control signals that allow the user or another program to perform data operations on the system.
- External Outputs (EO) – these are external output functions with their logical data groups and elements. External outputs are elementary processes, which include, for example, screens, report charts, or control signals generated by the system for the end user.
- External queries (EQ) – these are external query functions with their logical data groups and elements. External queries are elementary processes. Nowadays, thanks to modern applications, it is difficult to distinguish the difference between EO and EQ.

Data function types:

- External Interface Files (EIF) - these are external interface files with their records and elements. These are data, which are maintained outside of the system boundary.
- Internal Logical Files (ILF) - these are internal logical files with their records and elements. These are data, which are maintained in the system boundary.

Component	Complexity weights		
	Low	Medium	Large
EI	3	4	6
EO	4	5	7
EQ	3	4	6
EIF	5	7	10
ILF	7	10	15

Table 1. Classification of the complexity of components

After system distributing into the component groups by assigning the number of unique functions (NUF), it is necessary to classify the FP according to a complexity weights into three groups (low, medium and large).

Table 1 shows weights for classification the complexity of individual component groups. Calculation of Unadjusted Function Points (UFP) is performed as the sum of multiplications of NUF and appropriate complexity weights.

$$UFP = \sum_{i=1}^m \sum_{j=1}^k (NUF_{i,j} \times weight_{ij}) \quad (1)$$

where  $m$  is a number of components and  $k$  is a number of complexity group.

The next step is to include the influence of 14 factors called General Systems Characteristics (GSC), which are described in Table 2.

GSC factor	Description
F <sub>1</sub>	Does the system require backup and recovery?
F <sub>2</sub>	Are Data Required for Communication?
F <sub>3</sub>	Does the system include a distributed processing function?
F <sub>4</sub>	Is critical performance required?
F <sub>5</sub>	Will the system work during heavy loads?
F <sub>6</sub>	Does the system require direct data input?
F <sub>7</sub>	Do data inputs require multiple screens or operations?
F <sub>8</sub>	Are the main files up-to-date?
F <sub>9</sub>	Are inputs, outputs, files and queries intricate?
F <sub>10</sub>	Is internal processing complicated and complex?
F <sub>11</sub>	Is the code designed for reuse?
F <sub>12</sub>	Are Conversions and Installation Included in Design?
F <sub>13</sub>	Is the system designed for multiple installations in different locations?
F <sub>14</sub>	Is the application designed to make it easy for users to make changes?

Table 2. GSC - factors affecting the system

Each factor is rated by degree of effect on the application on the 0-5 scale. The following Table 3. represents the significance of the influence factors rating.

Influence	Rating
None	0
Not relevant	1
Mild	2
Average	3
Significant	4
Very strong	5

Table 3. Factors weights

Using GSC factors and their rating is possible to express formula for Value Adjustment Factor (VAF) calculation:

$$VAF = 0.65 + 0.01x \sum_{i=1}^{14} (F_i \times rating) \quad (2)$$

Finally, the number of Adjusted Function Points (AFP) is calculated according to the formula:

$$AFP = UFP \times VAF \quad (3)$$

### 3.2. Regression models

A linear regression model is an approach to modeling a linear relationship between independent (explanatory) variables (one or many) and the dependent variable (response). If the goal of the model is an estimation of the response, then the sum of squared residuals (SSR) is used for model evaluation. The model is represented by linear curve (line), for which SSR is minimal [4], [22], [23].

The least square method is the most common method used for fitting. The linear curve is called a regression line. The case of one independent variable is called a simple linear regression (SLR). For more than one independent variables, the method is called multiple linear regression (4). The multiple linear regression model (MLR) could be defined by the formula:

$$y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \dots + \beta_p X_{ip} + \varepsilon_i, \quad i = 1, \dots, n \quad (4)$$

where  $y_i$  is the dependent variable and  $X_{i1} \dots X_{ip}$  are independent variables (predictors),  $\beta_0$  is an intercept, and  $\beta_1 \dots \beta_n$  are regression coefficients. The value of  $\varepsilon_i$  represent the estimation error. The model is designed as a matrix, where each row represents an observation.

Linear regression can be represented as polynomial regression [24], [25] in which the relationship between the dependent variable and the independent variables is modeled as an  $m^{\text{th}}$  degree polynomial (5)

$$y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2}^2 + \dots + \beta_p X_{ip}^m + \varepsilon_i, \quad i = 1, \dots, n. \quad (5)$$

In the case of MLR, the model construction can benefit from the stepwise regression [22], [26], [27]. The goal of the stepwise regression method has identified a set of independent variables and the form of a linear relationship, which brings the minimal value of control criteria. Stepwise process of multiple linear regression is based on the forward - and backward selection that involves an automatic process for the selection of independent variables; and can be briefly described as follows:

1. Setting a starting model, which contains predefined terms,
2. Setting limits for the final model – what type of terms are needed in a model - the linear term is used, squared terms and vice versa,
3. Setting the evaluation threshold – the sum of squared errors (SSE),
4. Adding or removing terms; retesting the model;
5. Stepwise regression is stopped when there is no further improvement in estimation.

Model is rechecked and evaluated after adding (forward) or removing (backward) a term to see if its significance has been reduced below a specified tolerance level. Significance in this context means, that model brings a lower value of SSE.

Forward selection starts without variables in the model and then iterates to add a variable. If a non-significant term is found, it is removed again from the model, and text term is added.

Backward selection works in a similar manner, but terms are removed if they are found to be non-significant. Therefore, stepwise regression requires two significance levels: the first for adding the term and the second for removing the term. Final model can be controlled by setting a type of final model – which terms will be kept – without evaluation their significance to model.

### 3.3. Evaluation criteria

For development effort estimation, there are exist several commonly used criteria, which are accepted as a standard evaluation. In this research, all performed simulations were evaluated according to four of selected metrics. Their equations are given as follows:

Mean Squared Error (MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \widehat{y}_i)^2 \quad (6)$$

The Magnitude of Relative Error (MRE):

$$MRE = \frac{|y_i - \widehat{y}_i|}{y_i} \quad (7)$$

The Mean of Magnitude of Relative Error (MMRE):

$$MMRE = \frac{1}{n} \sum_{i=1}^n MRE_i \quad (8)$$

Prediction level (PRED):

$$PRED(l) = \frac{1}{n} \sum_{i=1}^n \begin{cases} 1 & \text{if } MRE \leq l \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where  $n$  is the number of observations,  $y_i$  is the known real value,  $\hat{y}_i$  is the predicted value and  $l$  is the threshold of percentage error.  $PRED(l)$  describes overall estimation quality within a selected level of percentage errors.

#### 4. Experiment design

##### 4.1. Dataset description

It was used an ISBSG project repository Release 13 as an examined dataset. The ISBSG dataset includes 6760 historical projects. Information according to full dataset is in [28]. Dataset used in our experiments was obtained by selection of projects with high data quality rating (A) and known parameters EI, EO, EQ, EIF, ILF, VAF (see chapter 3.1) needed for UFP computation. Based on these requirements we have got dataset DS of 631 projects. Next experiments will be evaluated by using two sub-datasets (training set DS1 and testing set DS2), which were created using the hold-out approach in ratio 2:1.

Table 4 shows characteristics of used sub-datasets DS1 and DS2. All values in Table 4 are based on UFP and Normalized Work Effort (NWE), which represents development team full effort (real) in person-hours. Parameter  $n$  indicates a number of dataset projects.

	<b>n</b>	<b>Mean UFP</b>	<b>Median UFP</b>	<b>Min UFP</b>	<b>Max UFP</b>	<b>SD UFP</b>	<b>Mean NWE</b>	<b>Median NWE</b>	<b>Min NWE</b>	<b>Max NWE</b>	<b>SD NWE</b>
DS1	423	369.24	198.0	25.0	2983.0	480.94	3791.4	2039.0	31.0	52172.0	5986.6
DS2	208	300.53	204.5	13.0	1810.0	308.91	3347.7	1925.0	107.0	36046.0	4469.9

Table 4. Dataset's Characteristics

In Fig. 1 a boxplot of the training (DS1) and testing (DS2) datasets is presented. As we can see, there are many possible outliers in the datasets DS1 and DS2, but all analyses in this study were carried out on complete datasets DS1 and DS2, including possible outliers.

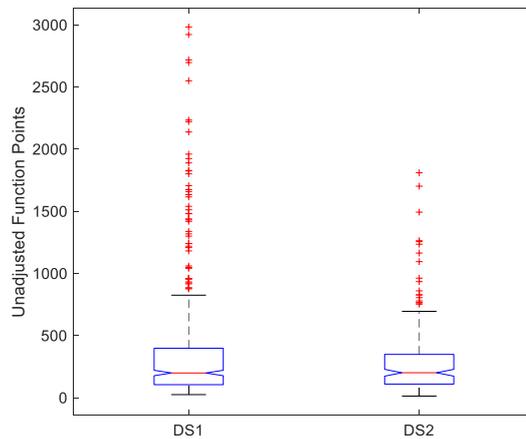


Fig. 1. Boxplot of the training and testing datasets

##### 4.2. VAF factor analysis

As the first step, the correlation between VAF and AFP and correlation between VAF and NWE are studied. The correlation, which is used is a Person correlation coefficient, which can be from interval  $(-1; 1)$ . The value 1 is a strong positive correlation; value -1 represent strong negative correlation. A positive correlation can be interpreted as a linear relationship – if VAF is increasing, then AFP or NWE will be increased too. VAF can influence an AFP in a range of  $\pm 35\%$  [1]. Correlations between VAF, AFP, and NWE for datasets DS1 and DS2 are presented in

Table 5. As can be seen, there is a weak correlation in both datasets. Therefore, further analysis of VAF factor influence of estimation accuracy is needed.

	AFP	NWE
VAF (DS1)	0.16171	0.20874
VAF (DS2)	0.12750	0.12030

Table 5. Correlation coefficients between variables VAF, AFP and NWE

4.3. Tested methods

This paper aimed to determine the effect of the VAF factor on the accuracy of the estimate. Therefore, three methods for effort estimation were tested. First, the basic unmodified FP method, second, the modified FP method using a linear regression model (LRM) and third, the modified FP method using stepwise regression (SRM) was tested. All three methods were tested in the variant without and with the VAF factor as an independent variable. Regression models require historical data for evaluating the effort of a new project, where the dependent variable  $y_i$  equals the NWE vector and the AFP attributes (EI, EO, EQ, EIF, ILF and VAF) are used as independent variables ( $X_i$ ). Models were computed according to equations (1) – (3). Predictors were used as variables, and all of them, were obtained by FP method. Estimations of project development effort obtained by LRM and SRM are reported in person-hours. FP method brings a project size estimation. To convert estimated size to an estimated development effort, a parameter *meanPDR* (10) is needed to be set:

$$meanPDR = \frac{1}{n} \sum_n PDR \tag{10}$$

Where *meanPDR* is mean Product Delivery Rate value. In this study, *meanPDR* = 11.86, which was computed from training dataset DS1;  $n$  is a number of projects (observations), and *PDR* is a product delivery rate for each of the projects in training dataset DS1. Then project development effort (11) is obtained using the following formula:

$$effort = AFP \times meanPDR \tag{11}$$

Regression models were obtained employing a training dataset DS1. The model formulas and other parameters are summarized in Table 6.

	Formula	R2	AdjR2
LRM	$y \sim 1 + AFP$	0.549	0.548
LRM-VAF	$y \sim 1 + AFP \times VAF$	0.572	0.571
SRM	$y \sim 1 + AFP + AFP^2$	0.559	0.557
SRM-VAF	$y \sim 1 + (AFP \times VAF) + (AFP \times VAF)^2$	0.579	0.577

Table 6. Regression models' formulas

As can be seen from Fig. 2., there is some tendency for larger fitted values to have larger residuals. All tested models' errors are proportional to the measured values.

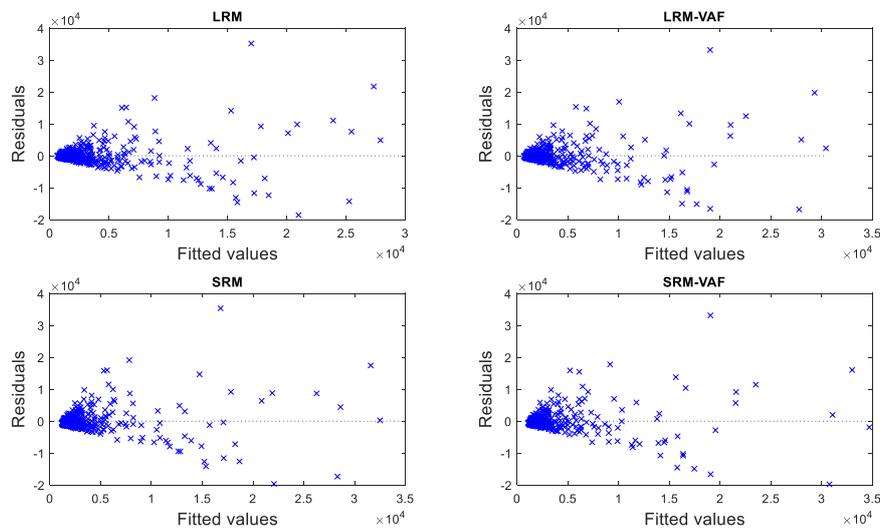


Fig. 2. Comparison of residuals vs. fitted values of tested methods

In Fig 3., estimation errors of tested methods can be shown. As can be seen, means of estimation errors for all tested methods are close to 0, which can be interpreted as errors are normally distributed. The number of outliers is relatively high, which has a negative impact on MSE or MMRE value.

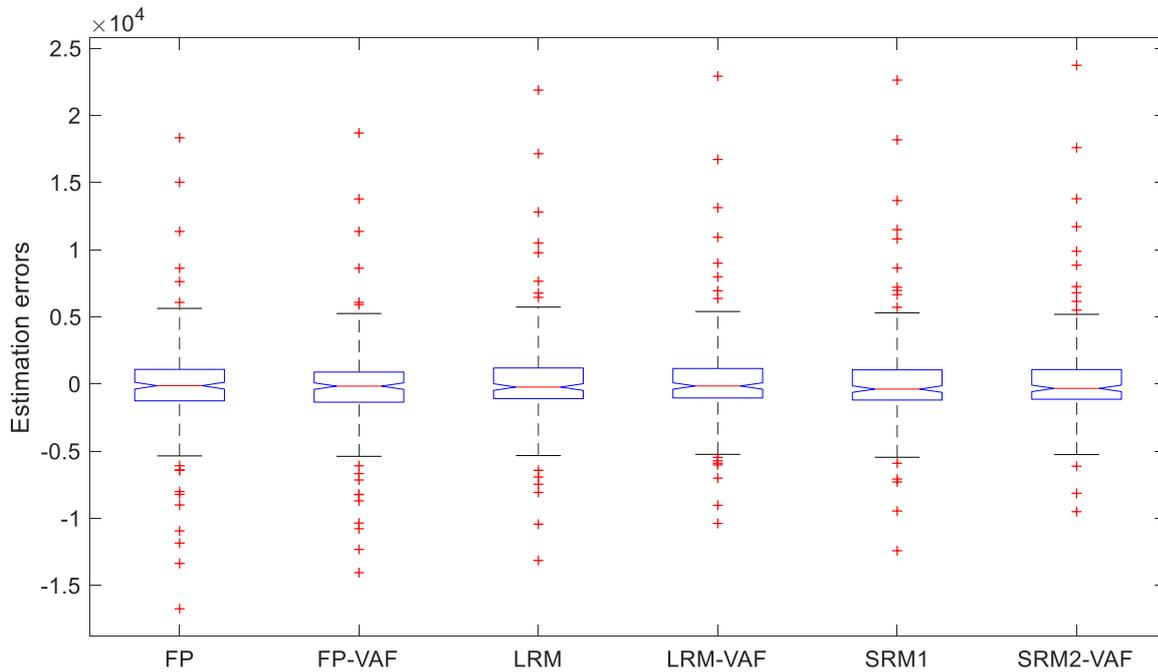


Fig. 3. Comparison of estimation errors of tested methods

The values in Table 7 are given in person-hours. The unit of the measure also results in extremely high values of the MSE parameter, where the sum of squared errors is calculated.

	<b>MSE</b>	<b>MMRE</b>	<b>PRED(0.25)</b>
FP	1.2592e+07	1.4159	0.20192
FP-VAF	1.1881e+07	1.4335	0.25481
LRM	1.1904e+07	1.3394	0.22115
LRM-VAF	1.1376e+07	1.3225	0.20192
SRM	1.2274e+07	1.4623	0.22115
SRM-VAF	1.1675e+07	1.4265	0.23077

Table 7. Results comparison

### 5. Results discussion

According to the presented values of the correlation coefficients in Table 5, it can be seen that correlations between VAF, and AFP, and NWE in both datasets are weak (coefficients are from 0.12030 to 0.20874). Therefore, the authors decided to further verification of VAF's impact on the accuracy of the development effort estimation. Three methods for the effort estimation were tested in the variant without and with the VAF factor; basic unmodified FP method, the modified FP method using a linear regression model (LRM) and the modified FP method using stepwise regression (SRM).

The answer to RQ1: VAF factor improve estimation accuracy negligibly. The simulation results of MSE presented in Table 7 show that the use of VAF brings approximately 5.7 % improvement in case of the FP, 4.4 % improvement in case of the LRM and 4.9 % improvement in case of the SRM.

The answer to RQ2: Simulation results prove that all tested methods are nearly equivalent in the sense of estimation accuracy. Results of MSE presented in Table 7 show that LRM brings approximately 5.5 % improvement compared to FP. SRM brings approximately 2.5 % improvement compared to FP and SRM brings approximately 3.1 % degradation compared to LRM.

## 6. Conclusion

This article aimed to determine whether the VAF factor has an impact on the accuracy of the development effort estimation and how large that effect is. To meet these goals, as a first step, we have analyzed the data regarding the correlation between VAF and AFP and between VAF and NWE.

Subsequently, was performed a simulation verification of the development effort estimation using three methods: FP, LRM and SRM in two variants without and with the VAF. As can be seen from Table 7, none of the tested methods brings a significant improvement in estimation, when VAF is used and all tested methods are nearly equivalent in the sense of estimation accuracy. For the testing purposes, it was used ISBSG dataset, and simulation calculations were performed using the Matlab environment. Future work will explore the possibility of improving development effort estimations by using other enhancements to the original FPM, but also by analyzing data from the perspective of outliers, data clustering, etc.

## 7. Acknowledgments

This work was supported by the Faculty of Applied Informatics, Tomas Bata University in Zlín within the Excellent Programme under the project No. RO30186021025 / 2102.

## 8. References

- [1] Azzeh, M., Nassif, A. B., & Banitaan, S. (2018). Comparative analysis of soft computing techniques for predicting software effort based use case points. *IET Software*, 12(1), 19-29. doi:10.1049/iet-sen.2016.0322
- [2] Nawrocki, J., Ochodek, M., Jurkiewicz, J., Kopczynska, S., & Alchimowicz, B. (2014). Agile Requirements Engineering: A Research Perspective. *Sofsem 2014: Theory and Practice of Computer Science*, 8327, 40-51.
- [3] Pow-Sang, J. A. (2017). Experiences using the Jigsaw learning technique to teach IFPUG function points. 2017 Ieee World Engineering Education Conference (Edunine).
- [4] Silhavy, R., Silhavy, P., & Prokopova, Z. (2018). Evaluating subset selection methods for use case points estimation. *Information and Software Technology*, 97, 1-9. doi:https://doi.org/10.1016/j.infsof.2017.12.009
- [5] Urbanek, T., Kolcavova, A., & Kuncar, A. (2017). Inferring productivity factor for use case point method. Paper presented at the 28th DAAAM International Symposium on Intelligent Manufacturing and Automation, DAAAM 2017.
- [6] Chwastyk, P., & Kołosowski, M. (2014). Estimating the cost of the new product in development process. Paper presented at the 2013 24th DAAAM International Symposium on Intelligent Manufacturing and Automation, Zadar.
- [7] Cvetkovic, N., Morača, S., Jovanović, M., Medojević, M., & Lalić, B. (2017). Enhancing the agility and performances of a project with lean manufacturing practices. Paper presented at the 28th DAAAM International Symposium on Intelligent Manufacturing and Automation, DAAAM 2017.
- [8] Bock, D. B., & Klepper, R. (1992). Fp-S - a Simplified Function Point Counting Method. *Journal of Systems and Software*, 18(3), 245-254. doi:Doi 10.1016/0164-1212(92)90101-0
- [9] Tichenor, C. B. (1997). The Internal Revenue Service functional point analysis program: A brief. *Compsac 97 : Twenty-First Annual International Computer Software & Applications Conference*, 591-592. doi:Doi 10.1109/Cmpsac.1997.625077
- [10] Ochodek, M., & Ozgok, B. (2015). Functional and Non- functional Size Measurement with IFPUG FPA and SNAP - Case Study. *Software Engineering in Intelligent Systems (Csoc2015)*, Vol 3, 349, 19-33. doi:10.1007/978-3-319-18473-9\_3
- [11] He, H. J., Xia, L., Luo, L., Yan, H. Z., Zhu, J., & Tang, J. B. (2016). Improvement of Complexity Matrix of IFPUG in Embedded-Software Projects. 2016 2nd Ieee International Conference on Computer and Communications (Iccc), 911-914.
- [12] Prokopova, Z., Silhavy, P., & Silhavy, R. (2018). Influence Analysis of Selected Factors in the Function Point Work Effort Estimation. Paper presented at the Proceedings of the Computational Methods in Systems and Software.
- [13] Celar, S., Mudnic, E., & Kalajdzic, E. (2009). Software size estimating method based on MK II FPA 1.3 unadjusted. Paper presented at the Annals of DAAAM for 2009 and 20th International DAAAM Symposium "Intelligent Manufacturing and Automation: Focus on Theory, Practice and Education", Vienna.
- [14] Ruhe, M., Jeffery, R., & Wiczorek, I. (2003). Using Web objects for estimating software development effort for Web applications. Paper presented at the Proceedings - International Software Metrics Symposium.
- [15] Lokan, C. J. (2000). An empirical analysis of function point adjustment factors. *Information and Software Technology*, 42(9), 649-660.
- [16] Lokan, C. J. (1999). Empirical study of the correlations between function point elements. Paper presented at the International Software Metrics Symposium, Proceedings.
- [17] Jeffery, D. R., Low, G. C., & Barnes, M. (1993). A Comparison of Function Point Counting Techniques. *Ieee Transactions on Software Engineering*, 19(5), 529-532. doi:10.1109/32.232016
- [18] Kitchenha, B. (1997). The Problem with Function Points. *Ieee Software*, 14(2), 29-31. doi:10.1109/MS.1997.582972

- [19] Abran, A., & Robillard, P. N. (1994). Function points: A study of their measurement processes and scale transformations. *The Journal of Systems and Software*, 25(2), 171-184. doi:10.1016/0164-1212(94)90004-3
- [20] Bundschuh, M., & Dekkers, C. (2008). *The IT measurement compendium: estimating and benchmarking success with functional size measurement*: Springer Science & Business Media.
- [21] ISO/IEC. (2007). ISO/IEC 14143-1:2007(en) - Information technology — Software measurement — Functional size measurement. In Part 1: Definition of concepts: ISO/IEC.
- [22] Silhavy, P., Silhavy, R., & Prokopova, Z. (2018). Stepwise Regression Clustering Method in Function Points Estimation. Paper presented at the Proceedings of the Computational Methods in Systems and Software.
- [23] Silhavy, R., Silhavy, P., & Prokopova, Z. (2017) Improving algorithmic optimisation method by spectral clustering. In: Vol. 575. *Advances in Intelligent Systems and Computing* (pp. 1-10).
- [24] Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). *Introduction to linear regression analysis* (5th ed.). Hoboken, NJ: Wiley.
- [25] Idri, A., Amzal, F. A., & Abran, A. (2016). Accuracy Comparison of Analogy-Based Software Development Effort Estimation Techniques. *International Journal of Intelligent Systems*, 31(2), 128-152. doi:10.1002/int.21748
- [26] Silhavy, P., Silhavy, R., & Prokopova, Z. (2017) Evaluation of data clustering for stepwise linear regression on use case points estimation. In: Vol. 575. *Advances in Intelligent Systems and Computing* (pp. 491-496).
- [27] Amasaki, S., & Lokan, C. (2014). The Effects of Gradual Weighting on Duration-Based Moving Windows for Software Effort Estimation. *Product-Focused Software Process Improvement, Profes 2014*, 8892, 63-77.
- [28] ISBSG. (2015). ISBSG Development & Enhancement Repository – Release 13. Retrieved 2.2.2015, from International Software Benchmarking Standards Group (ISBSG) <http://isbsg.org>.