Should Function Point Elements be Used to Build Prediction Models?

Kohei Yoshigami, Masateru Tsunoda
Department of Informatics
Kindai University
Osaka, Japan
tsunoda@info.kindai.ac.jp

Yuto Yamada, Shinji Kusumoto
Graduate School of Information Science and Technology
Osaka University
Osaka, Japan
{y-yuto, kusumoto}@ist.osaka-u.ac.jp

Abstract*—Schedule and cost managements are indispensable to develop large software, and predicting of project results such as development effort is the basis of the management. So, high accuracy of project prediction is needed. Many researches about software project prediction use software size as a major explanatory variable. To estimate software size, function point method is often applied. In the method, five fundamental function elements are identified and scores called function point (FP) are given to each element. After that, the sum of FP is calculated. Generally, the FP elements are not used as independent variables directly, and only the sum of them is used as an independent variable. As long as we know, it is not clear to what extent FP elements affect quality, cost, and delivery of software project. Also, it is not clear whether using the elements as independent variables improves prediction accuracy of quality, cost, and delivery or not. So, this study analyzed the influence of the elements to quality, cost and delivery. Also, we evaluated the accuracy of software project prediction when the elements were used as independent variables. The goal of the study is to support data collection and variable selection of the prediction. Experimental results showed that when duration and effort were estimated by analogy based estimation, using FP elements as independent variables was effective to enhance the estimation accuracy.

Keywords—QCD; fault prediction; effort estimation; duration prediction

I. INTRODUCTION

Schedule and cost managements are indispensable to develop large software, and predict results of project such as development effort is the basis of the management. So, high accuracy of project prediction (small difference between predicted and actual value) is needed. When the results are predicted by a mathematical model, the model is built using a dataset collected at past software development projects. To achieve high accurate prediction, many prediction methods have been proposed. Analogy based estimation is one of major estimation models. It selects projects (neighborhood projects) which are similar to the estimated project from past project dataset, and estimates results based on similar projects' results.

One of the advantages of analogy based estimation is that estimation results are comprehensible for estimators such as project managers [16], because they can confirm neighborhood projects used for estimation. Although ordinary estimation models like linear regression model estimate various target projects' effort by one model, analogy based estimation does not make such a model, and estimates results by neighborhood projects' results. So analogy based estimation can reflect individuality of each target project in estimation.

Many researches about software project prediction use software size as a major explanatory variable. To estimate software size, function point method [6] is often applied. Estimating software size is indispensable to make a project plan. In the method, five fundamental function elements are identified from all functions of the estimation target software. After that, scores called function point (FP) are given to each element, based on the complexity of the functions. Five fundamental function elements are the followings [2]:

- Internal Logical Files (ILF): Data files logically related to the estimation target software. They are updated in the software.
- External Interface Files (EIF): Data files referred by the estimation target software, but not updated in the software.
- External Inputs (EI): Processes which update ILF using data input from outside of the estimation target software.
- External Outputs (EO): Processes which output processed data to the outside of the estimation target software. The processed data is made with some conditions and calculations.
- External Inquiries (EQ): Processes which output data to outside of the estimation target software. The data is not processed data, and the processes do not update ILF.

Especially, the sum of FP on estimation target software is called as application FP. Also, in maintenance projects, the method measures FP of added, modified and deleted functions separately.

Generally, above FP elements are not used as independent variables directly, and the sum of them (i.e., application FP) is used as an independent variable. However, it is not clear to what extent FP elements affect quality, cost, and delivery of software project. Also, it is not clear whether using the elements as independent variables improves prediction accuracy of quality, cost, and delivery or not.

This study analyzed the influence of FP elements thoroughly. That is, we analyzed the influence of elements to quality, cost and delivery. After the analysis, we evaluated the accuracy of software project prediction models when the elements were used as independent variables. The goal of the study is to support data collection and variable selection of the prediction models. Concretely speaking, in the analysis, we used ISBSG dataset, and analyzed the relationship of FP elements to fault ratio, productivity (efficiency), and delivery speed (The definitions of them are explained in section II). After that, using analogy based estimation, we predicted the number of faults, development effort and project duration, using FP elements as independent variables, to evaluate the effectiveness of the elements.

II. DATASET

We used the dataset of software development which was collected from organizations in 20 countries by ISBSG (International Software Benchmarking Standards Group) [7]. The version of the dataset is Release 9. The dataset includes 99 variables and 3026 projects performed between 1989 and 2004. ISBSG dataset is also called as cross-company dataset, since it was collected from various companies. To align status of analyzed projects, we selected projects which satisfy conditions shown in the following. The condition is based on the study of Lokan et al. [10].

- Data quality rating is A or B
- FP measurement rating is A or B
- FP measurement method is IFPUG
- Effort includes working time of development team only (e.g., working time of back-office is not included)

To analyze influence of FP elements to quality, cost, and delivery, we defined fault ratio, productivity, and development speed, and analyzed the relationships (see Table I). Also, we assume that when the some elements are very small or large, it affects quality, cost and delivery. To analyze that, we defied the ratio of the elements (each elements were divided by application FP). Table I shows definitions of each variable such as EI ratio. On the nominal variables (e.g., business sector), when the number of cases on a category was small, the category was treated as missing variables (i.e., we removed the categories when the number of them was small).

Generally, characteristics of new development projects and maintenance projects are very different. Dataset is often stratified by the development type. So, we stratified the dataset before applying statistical analysis (re-development was treated as new development). Note that modified FP and deleted FP are recorded in the maintenance projects only, and added FP is 100% in new development projects.

As preliminary analysis, we analyzed the relationships of FP elements to fault ratio, productivity and development speed. Projects are selected based on the above conditions. In the subset, 421 new development projects and 606 maintenance projects were included. There were missing values in the dataset, and therefore the number of projects used in the analysis was different on each variable.

TABLE I. VARIABLES USED IN THE ANALYSIS

Variables	Detail
Development type	New development or enhancement
Application FP	Total function point of software
Number of faults	Number of faults after software delivery
Effort	Total software development effort
Duration	Development duration of a project
Plan phase ratio	Effort of planning / total effort
Requirement analysis phase ratio	Effort of requirement analysis / total effort
Coding phase ratio	Effort of coding / total effort
Test phase ratio	Effort of testing / total effort
Implementation phase ratio	Effort of implementation / total effort
Business sector	Banking, communications, and so on
Architecture	Stand alone, client/server, and so on
Platform	Mainframe, PC, and so on
Programming language	C/C++/C#, Visual Basic, and so on
EI ratio	EI / application FP
EO ratio	EO / application FP
EQ ratio	EQ / application FP
ILF ratio	ILF / application FP
EIF ratio	EIF / application FP
Add FP ratio	FP added / application FP
Modify FP ratio	FP modified / application FP
Delete FP ratio	FP deleted / application FP
Fault ratio	Number of faults/ application FP
Productivity	Application FP / effort
Development speed	Application FP / project duration

Also, we stratified the dataset before predicting the number of faults, effort, and project duration. We eliminated projects in which a dependent variable or independent variables include missing values (i.e., we applied listwise deletion). So, in the prediction accuracy evaluation, the number of projects used to build the model are the followings:

- Predicting the number of faults (new development): 19
- Predicting the number of faults (enhancement): 43
- Predicting project duration (new development): 251
- Predicting project duration (enhancement): 359
- Predicting effort (new development): 270
- Predicting effort (enhancement): 364

Generally, it is difficult to collet number of faults after software delivery. So, it has many missing values and the number of projects used to predicting the number of faults was small.

III. RELATIONSHIP BETWEEN FP ELEMENTS AND OTHER VARIABLES

A. Overview

To clarify the influence of FP elements to quality, cost and delivery, we analyzed the relationship of FP elements to fault ratio, productivity, and development speed. Also, we assumed that when some FP elements are dominant, some development phases need effort (e.g., when EI is high, test phase needs more effort). Based on the assumption, we analyzed the relationship

TABLE II. RELATIONSHIPS BETWEEN FP ELEMENTS AND OTHER VARIABLES (NEW DEVELOPMENT)

		Fault ratio	Productivity	Development speed	Plan phase ratio	Requirement analysis phase ratio	Coding phase ratio	Test phase ratio	Implement phase ratio
EI ratio	ρ	-0.01	0.15	0.07	-0.08	-0.07	0.17	-0.14	**0.30**
	p-value	98%	0%	17%	55%	53%	11%	18%	**2%**
	# of projects	36	421	389	54	93	95	89	**57**
EO ratio	ρ	-0.01	0.00	-0.01	-0.08	0.04	-0.04	0.14	-0.18
	p-value	96%	97%	81%	59%	69%	71%	20%	18%
	# of projects	36	421	389	54	93	95	89	57
EQ ratio	ρ	-0.12	0.08	0.11	0.04	-0.06	0.11	-0.13	0.01
	p-value	48%	9%	3%	76%	58%	29%	22%	97%
	# of projects	36	421	389	54	93	95	89	57
ILF ratio	ρ	-0.09	0.15	0.12	-0.06	-0.09	-0.03	0.01	0.00
	p-value	61%	0%	2%	67%	41%	79%	90%	99%
	# of projects	36	421	389	54	93	95	89	57
EIF ratio	ρ	0.03	**-0.27**	-0.12	-0.01	0.11	-0.08	0.09	-0.03
	p-value	86%	**0%**	2%	96%	29%	42%	41%	85%
	# of projects	36	**421**	389	54	93	95	89	57

TABLE III. RELATIONSHIPS BETWEEN FP ELEMENTS AND OTHER VARIABLES (ENHANCEMENT)

		Fault ratio	Productivity	Development speed	Plan phase ratio	Requirement analysis phase ratio	Coding phase ratio	Test phase ratio	Implement phase ratio
EI ratio	ρ	0.02	-0.04	-0.07	0.34	-0.06	0.01	0.09	-0.38
	p-value	87%	28%	9%	14%	49%	88%	26%	4%
	# of projects	52	606	562	20	147	151	148	29
EO ratio	ρ	**-0.20**	0.02	0.05	-0.39	0.05	-0.03	-0.03	0.49
	p-value	**15%**	63%	21%	9%	57%	68%	72%	1%
	# of projects	**52**	606	562	20	147	151	148	29
EQ ratio	ρ	**0.34**	-0.13	0.17	0.36	-0.19	0.12	0.11	-0.39
	p-value	**1%**	0%	0%	11%	2%	14%	16%	4%
	# of projects	**52**	606	562	20	147	151	148	29
ILF ratio	ρ	-0.16	0.14	0.04	0.43	0.07	-0.10	0.08	0.30
	p-value	26%	0%	37%	6%	39%	22%	31%	12%
	# of projects	52	606	562	20	147	151	148	29
EIF ratio	ρ	-0.01	0.03	0.11	-0.36	-0.08	0.16	-0.16	-0.01
	p-value	92%	53%	1%	12%	36%	5%	5%	96%
	# of projects	52	606	562	20	147	151	148	29
Add FP ratio	ρ	0.07	-0.06	-0.11	-0.58	-0.02	0.15	**-0.33**	0.48
	p-value	62%	12%	1%	1%	77%	6%	**0%**	1%
	# of projects	52	606	562	20	147	151	**148**	29
Modify FP ratio	ρ	-0.10	0.06	0.09	0.56	-0.01	-0.14	**0.34**	-0.40
	p-value	49%	15%	3%	1%	87%	8%	**0%**	3%
	# of projects	52	606	562	20	147	151	**148**	29
Delete FP ratio	ρ	0.10	0.02	0.15	0.36	0.14	-0.01	0.00	-0.25
	p-value	47%	64%	0%	12%	9%	87%	97%	19%
	# of projects	52	606	562	20	147	151	148	29

between FP elements and ratio of each phase (e.g. test phase ratio). When the relationship is strong, using FP elements as independent variables may enhance prediction accuracy of effort of each development phase. Some studies [5] predict effort of each development phase.

We analyzed the relationship between FP elements and business sector, assuming that some business sectors have biases of FP elements (e.g., EI is high, when business sector is banking). Similarly, we analyzed the relationship between FP elements and programming language, assuming that when some FP elements are high, some programing languages are selected (e.g., when EI is high, COBOL is selected). If FP elements have strong relationships to them, FP elements can be used as substitution of them, especially when they are missing values. For example, when FP elements have strong relationship to business sector but it is not recorded (or not disclosed), FP elements can be used as independent variables in prediction models, instead of business sector. Note we easily know programming language if it has missing values but we collect the dataset directly. However, when we analyze the dataset collected by others such as ISBSG dataset, it is difficult to handle missing values of programming language.

When analyzing the relationship between FP elements and ratio scale variables, we used Spearman's rank correlation coefficient, to avoid influence of outliers. To analyze the relationship between FP elements and nominal scale variables, we used the correlation ratio. The ratio indicates strength of the relationship between ratio and nominal scale variables. The range of the value is [0, 1].

TABLE IV. RELATIONSHIPS BETWEEN FP ELEMENTS AND CATEGORICAL VARIABLES (NEW DEVELOPMENT)

	Business sector	Architecture	Platform	Programming Language
EI ratio	**0.37**	**0.41**	0.21	**0.37**
EO ratio	0.22	0.19	0.16	0.22
EQ ratio	0.22	0.13	0.15	0.23
ILF ratio	0.23	0.06	0.16	0.16
EIF ratio	**0.36**	0.24	0.18	**0.35**

TABLE V. RELATIONSHIPS BETWEEN FP ELEMENTS AND CATEGORICAL VARIABLES (ENHANCEMENT)

	Business sector	Architecture	Platform	Programming Language
EI ratio	**0.48**	0.29	**0.35**	**0.36**
EO ratio	0.20	0.16	0.15	0.23
EQ ratio	**0.36**	0.22	0.21	**0.36**
ILF ratio	**0.34**	0.18	0.15	0.22
EIF ratio	**0.40**	0.22	0.29	**0.31**

B. Relationships to Fault Ratio, Productivity, and Development Speed

Table II shows correlation coefficients of FP elements to fault ratio, productivity, and development speed, when development type is new development. Table III shows the coefficients when the type is maintenance. Generally, when absolute value of the coefficients is larger than 0.2, it is regarded that there is weak correlation. So, we focused on such cases in the tables, and they are shown by boldface. When the type is new development, the absolute value of the correlation between EIF ratio and productivity was larger than 0.2. Also, when the type is maintenance, the absolute values between EO ratio and fault ratio, and EQ ratio and fault ratio were larger. Add, modify, and delete FP ratio, which were recorded only in maintenance projects, did not have explicit relationships to fault ratio, productivity and development speed. The analysis result suggests that FP elements did not have strong relationships to quality, cost and delivery. However, some elements have weak relationships to them. So, using FP elements as independent variables may improve the accuracy of prediction in some cases.

C. Relationships to Ratio of Each Development Phase

Table II and III also show the relationship between FP elements and ratio of each development phase. When development type is maintenance, ratio of plan phase and implementation phase had many missing values. So, we did not analyzed them when the type is maintenance.

When the type is new development, the absolute value of the correlation between EI ratio and implementation phase ratio was larger than 0.2. Also, when the type is maintenance, the absolute values between add FP ratio and test phase ratio, and modify FP ratio and test phase ratio were larger. It is reasonable that effort of test phase increases when software is modified greatly. Add FP ratio and test phase ratio were weak correlation. This is because add and modify FP are in a trade-off relationship. From the result, modify FP ratio is expected to be effective to improve estimation accuracy when test phase effort is estimated.

D. Relationships to Nominal Scale Variables

Table IV and V show correlation ratio between FP elements and nominal scale variables. Table IV includes analysis results of new development, and Table V does those of maintenance. We focused cases where the absolute values were larger than 0.3, and denote them by boldface. When development type was maintenance, the relationships between FP elements and business sector were relatively strong. Except for some cases, other nominal scale variables did not have strong relationships to FP elements, regardless of development type. The results suggest that FP elements has moderate relationships to business sector. Overall, FP elements did not have strong relationships to nominal scale variables, and therefore it is difficult to use FP elements as the substitution of the variables.

Additionally, we analyzed relationships of business sector to add, modify, and delete FP ratio. As a result, correlation ratio of them were 0.35, 0.34 and 0.18. So, it is not probable that modify (or add) FP ratio is high (or low) on some business sectors.

IV. SOFTWARE PROJECT PREDICTION

A. Analogy Based Estimation

The origin of analogy based estimation is CBR (case based reasoning) [14], which is studied in artificial intelligence field. Shepperd et al. [15] applied CBR to software development effort estimation. CBR selects a case similar to current issue from accumulated past cases, and applies solution of the case to the issue. CBR assumes similar issues can be solved by similar solution. Analogy based estimation assumes neighborhood (similar) projects (For example, development size and used programming language is similar) have similar effort, and estimates effort based on neighborhood projects' effort. Although ready-made estimation models such as COCOMO [2] can make estimation without stored software project dataset, analogy based estimation cannot estimate without it. It is a weak point of analogy based estimation, but it can be overcome by using public dataset.

B. Evaluation criteria

To evaluate accuracy of effort estimation, we used average and median of *AE* (Absolute Error), *MRE* (Magnitude of Relative Error) [3], and *BRE* (Balanced Relative Error) [12]. When x denotes actual effort, and $\hat{x}$ denotes estimated effort, each criterion is calculated by the following equations:

$$AE = |x - \hat{x}| \tag{6}$$

$$MRE = \frac{|x - \hat{x}|}{x} \tag{7}$$

TABLE VI. RELATIONSHIPS BETWEEN PREDICTION ACCURACY AND COMBINATIONS OF EXPLANATORY VARIABLES

Dependent variable	Development type	Independent variables	*Average AE*	*Median AE*	*Average MRE*	*Median MRE*	*Average BRE*	*Median BRE*
Number of faults	Maintenance	a	6.8	3.0	148%	74%	**205%**	138%
		b	5.9	2.3	142%	85%	214%	113%
		c	5.4	3.3	141%	73%	280%	153%
		d	7.1	2.6	220%	73%	264%	**85%**
	New	a	7.7	6.4	341%	73%	375%	154%
		b	8.6	5.9	372%	115%	402%	179%
		c	5.7	4.5	170%	81%	**195%**	111%
		d	5.0	3.5	175%	75%	224%	**108%**
Project duration	Maintenance	a	4.3	2.2	113%	61%	141%	70%
		b	5.9	2.3	203%	52%	224%	69%
		c	3.4	2.0	82%	54%	207%	99%
		d	2.9	1.8	87%	45%	**109%**	**62%**
	New	a	6.0	3.0	100%	50%	115%	**66%**
		b	6.8	3.2	111%	51%	125%	71%
		c	4.8	2.8	75%	53%	150%	88%
		d	4.3	2.6	85%	50%	**102%**	71%
Effort	Maintenance	a	2156	941	140%	54%	**174%**	79%
		b	2789	1060	199%	58%	225%	**75%**
		c	2605	1329	174%	66%	230%	109%
		d	2578	1362	175%	64%	220%	103%
	New	a	3182	1521	145%	52%	**186%**	**69%**
		b	3684	1685	173%	53%	198%	70%
		c	3967	2056	157%	68%	261%	134%
		d	3435	2067	189%	69%	231%	118%

$$BRE = \begin{cases} \frac{|x - \hat{x}|}{\hat{x}}, x - \hat{x} \geq 0 \\ \frac{|x - \hat{x}|}{x}, x - \hat{x} < 0 \end{cases} \quad (8)$$

Lower value of each criterion indicates higher estimation accuracy. Intuitively, *MRE* means relative error to actual effort. However, *MRE* have biases for evaluating under estimation [9]. Maximum *MRE* is 1 even if terrible underestimate is occurred (For instance, when actual effort is 1000 person-hour, and estimated effort is 0 person-hour, *MRE* is 1). So we gave weight to *BRE* whose evaluation is not biased [13].

C. Results

In the experiment, the dependent variable of each prediction was the number of faults, project duration or effort. To evaluate effect of FP elements to prediction accuracy, we combined independent variables as follows (we set the pattern 'c' to compare 'd'):

a. Application FP, nominal scale variables and FP elements

b. Application FP and nominal scale variables

c. Application FP

d. Application FP and FP elements (used them as the substitution of nominal scale variables)

To calculate evaluation criteria, we applied leave-one-out cross validation and applied analogy based estimation. We set the number of neighborhoods (parameters needed to apply analogy based estimation) as four, when the number of faults was predicted. Also, we set the number as eight, when project duration and effort was predicted. Note that some studies [11] predicts project duration.

Application FP and each FP element have strong relationships, and therefore multicollinearity should be considered when multiple linear regression is applied. In contrast, there is not need to care it when applying analogy based estimation (Using such variables sometimes enhance prediction accuracy and vice versa).

Table VI shows prediction accuracy of each variable. We focused on average and median of *BRE*, and show the minimum of them by boldface.

Predicting the number of faults: When development type was maintenance, and independent variables were application FP and nominal scale variables (b; see subsection C), prediction accuracy was relatively high (average and median *BRE* were the second smallest). On new development, average *BRE* was smallest when an independent variable was application FP (c). Median *BRE* was the smallest when independent variables were application FP and FP elements (d). The result suggests that FP elements do not greatly improve prediction accuracy of the number of faults. Note that the number of projects used for prediction was small (see section II). To enhance the reliability of the result, more projects are needed.

Predicting duration: When development type was maintenance, and independent variables were application FP and FP elements (d), average and median *BRE* were the smallest. Also, when the type was new development, prediction accuracy was relatively high by using the independent variables (d).

Predicting effort: When development type was new development, and independent variables are application FP, nominal scale variables, and FP elements (a), average and median *BRE* were the smallest. Also, when the type was maintenance, prediction accuracy was relatively high by using the independent variables (a).

From the results, when applying analogy based estimation, using FP elements as independent variables is expected to enhance the accuracy of effort and duration estimation. We applied linear size adjustment [1] in the estimation, and using FP elements is considered to improve the adjustment. Since in preliminary analysis, when the adjustment was not used, the accuracy was not improved.

V. RELATED WORK

Some studies use FP elements as independent variables. For example, Buglione et al. [4] used them to estimate software development effort. Also, Lavazza et al. [8] used them to predict application FP. This is because measuring application FP needs effort, and to lessen the effort, some of FP elements are measured, and based on them, application FP is predicted. However, as long as we know, past studies did not analyzed the influence of FP elements to quality, cost, and delivery thoroughly. Especially, the relationship between the elements and the nominal variables, and that of development phase ratio was not analyzed. Additionally, prediction accuracy of analogy based estimation using the FP elements was not evaluated before.

VI. CONCLUSIONS

To support data collection and variable selection of software project prediction, we analyzed the relationships of FP (function point) elements to quality, cost, and delivery. Also, we evaluated prediction accuracy when the elements were used as independent variables on analogy based estimation. The number of faults, project duration and effort were used as a dependent variable on the prediction. In the analysis, we observed the followings:

- Overall, FP elements did not have strong relationships to the fault ratio, productivity and development speed, except for some cases.
- Modified FP may be effective to improve prediction accuracy of effort estimation of test phase.
- The strength of the relationship between FP elements and business sector was moderate. Other nominal scale variables such as programing language did not have strong relationship to the elements.
- When duration and effort were estimated by analogy based estimation, using FP elements as independent variables was effective to enhance the estimation accuracy.

Especially, our result is effective when variables are selected before estimating effort and duration. That is, it is better to use FP elements as independent variables to improve estimation accuracy of effort and duration.

ACKNOWLEDGMENT

This research was partially supported by the Japan Ministry of Education, Science, Sports, and Culture [Grant-in-Aid for Scientific Research (C) (No. 16K00113)]

REFERENCES

[1] M. Azzeh, “A replicated assessment and comparison of adaptation techniques for analogy-based effort estimation,” Empirical Software Engineering, vol.17, no.1-2, pp.90-127, 2012.

[2] B. Boehm, Software Engineering Economics, Prentice Hall, 1981.

[3] S. Conte, H. Dunsmore, and V. Shen, Software Engineering, Metrics and Models, Benjamin/Cummings, 1986.

[4] L. Buglione, and C. Gencel, “Impact of Base Functional Component Types on Software Functional Size Based Effort Estimation,” Product-Focused Software Process Improvement, Lecture Notes in Computer Science, Vol.5089, pp.75-89, 2008.

[5] F. Ferrucci, C. Gravino, and F. Sarro, “Exploiting prior-phase effort data to estimate the effort for the subsequent phases: a further assessment,” In Proc. of International Conference on Predictive Models in Software Engineering (PROMISE), pp. 42-51, 2004.

[6] International Function Point Users Group, “IFPUG: Function Point Counting Practices Manual, Release 4.3,” International Function Point Users Group, 2010.

[7] International Software Benchmarking Standards Group (ISBSG): ISBSG Estimating: Benchmarking and research suite, ISBSG, 2004.

[8] L. Lavazza, S. Morasca, and G. Robiolo, “Towards a simplified definition of Function Points,” Information and Software Technology, Vol.55, No.10, pp.1796-1809, 2013.

[9] C. Lokan, “What Should You Optimize When Building an Estimation Model?” In Proc. of International Software Metrics Symposium (METRICS), pp.34, 2005.

[10] C. Lokan, and E. Mendes, “Cross-company and single-company effort models using the ISBSG Database: a further replicated study,” In Proc. of the International Symposium on Empirical Software Engineering (ISESE), pp.75-84, Rio de Janeiro, Brazil, Sep. 2006.

[11] C. López-Martín, and A. Abran, “Neural networks for predicting the duration of new software projects,” *Journal of Systems and Software*, Vol.101, pp. 127-135, 2015.

[12] Y. Miyazaki, M. Terakado, K. Ozaki, and H. Nozaki, “Robust Regression for Developing Software Estimation Models,” Journal of Systems and Software, vol.27, no.1, pp.3-16, 1994.

[13] K. Mølokken-Østvold, and M. Jørgensen, “A Comparison of Software Project Overruns-Flexible versus Sequential Development Models,” IEEE Transactions on Software Engineering, Vol.31, No.9, pp.754-766, Sep. 2005.

[14] E. Rich, and K. Knight, Artificial Intelligence, McGraw-Hill, 1990.

[15] M. Shepperd, and C. Schofield, “Estimating software project effort using analogies,” IEEE Transactions on Software Engineeiring, vol.23, no.12, pp.736-743, 1997.

[16] F. Walkerden, and R. Jeffery, “An Empirical Study of Analogy-based Software Effort Estimation,” Empirical Software Engineering, vol.4, no.2, pp.135-158, 1999.