DOI: 10.2507/28th.daaam.proceedings.084

INFERRING PRODUCTIVITY FACTOR FOR USE CASE POINT METHOD

Tomas Urbanek, Alena Kolcavova & Ales Kuncar

This Publication has to be referred as: Urbanek, T[omas]; Kolcavova, A[lena] & Kuncar, A[les] (2017). Inferring Productivity Factor for Use Case Point Method, Proceedings of the 28th DAAAM International Symposium, pp.0597-0601, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-11-2, ISSN 1726-9679, Vienna, Austria
DOI: 10.2507/28th.daaam.proceedings.084

Abstract

This paper aims to investigate and infer new range for productivity factor for Use Case Points method. Productivity factor is a ratio of man hours per use case point. In this paper, we use linear regression as an inference tool for productivity factor. This value is inferred from three different datasets. Then, we compared these inferred values with standard value for productivity factor. This study use MMRE (Mean Magnitude of Relative Error) measure and standard statistical methods for reliability assessment. The MMRE is also used for all comparisons. The experimental results show that the range for productivity factor is much narrow than recommended values with the mean of 15.

Keywords: Effort Estimation; Optimization; Productivity Factor; Software Engineering; Use Case Points

1. Introduction

Effort estimation is defined as the activity of predicting the amount of effort required to complete a development of software project [1]. Despite of a lot of effort of scientists and software engineers, there is still no optimal and effective method for every software project. The common way to improve effort estimation is to enhance the algorithmic methods. The algorithmic methods use mathematical formula for prediction.

It is very common that this group is also depending on the historical data. The most famous example of algorithmic methods are COCOMO (Constructive Cost Model) [2], FP (Function Points) [3] and last but not least UCP (Use Case Points) [4]. However, there are a many other algorithmic methods. It is essential that the calculation of effort estimation should be completed in early stage of software development cycle. The best case is if these calculations are known during the requirement analysis [4].

The accurate and reliable effort estimates are the crucial factor for the proper development cycle. These estimates are used for effective planning, monitoring and controlling the process of the software development. The prediction of effort estimations in software engineering is complex and complicated process. The main reason is that there are a lot of factors which influences the final prediction.

In this article, we investigated the properties of productivity factor for Use Case Points (UCP) method. We utilize linear regression model and t-test for inferring the confidence interval and the statistically significant value for productivity factor. In our best knowledge, no previous study has investigated the comparison of such models especially on different datasets when Use Case Points method and MMRE (Mean Magnitude of Relative Error) was used. Therefore, this study makes a major contribution to research of Use Case Points method.

1.1. Use Case Points Method

This effort estimation method was presented in 1993 by Gustav Karner [4]. It is based on a similar principle to the function point method. Project managers have to estimate the project parameters to four tables. Due to the aims of this paper, the detailed description of well-known Use Case Points method basic principles is insignificant and hence omitted. Please refer to [4], [5] for more detailed description of the Use Case Points method. The most basic equation for Use Case Points method is equation (1). The effort estimation is determined by multiplying the number of Use Case Points by the productivity factor.

$$UCP = (UUCW + UAW) \cdot TCF \cdot ECF \qquad (1)$$

where UUCW is Unadjusted Use Case Weight, UAW is Unadjusted Actor Weight, TCF is Technical Complexity Factor and ECF is Environmental Complexity Factor.

1.2. Productivity factor

The Productivity Factor (PF) is a ratio of the number of man hours per use case point [6]. The setting of productivity factor is one of the most difficult tasks in an accurate estimation. According to industry experts, if no historical data has been collected, an interval between 10 and 30 hours per Use Case point can be used. The typical value for productivity factor proposed by Karner was 20 and this is also a suggested value for brand new development team. Schneider and Winters [7] proposed a method based on counting number of environmental factors. Additionally, the work of Silhavy et al. [8] propose a new algorithm for calibration of productivity factor based on historical data. The best way to estimate this value is through analysis of previous completed projects for each software organization. This value will be more accurate than multi-organizational dataset [9]. For the calculation of estimated effort (EE), we used equation (2).

$$EE = UCP \cdot PF \qquad \textit{[man/hour]} \qquad (2)$$

2. Research Objectives

This section presents the design of the research questions. The research questions of our study could be outlined as follows:

- RQ-1: What is the mean value of productivity factor of software projects?
- RQ-2: Is appropriate to recommend a standard value of productivity factor as 20?
- RQ-3: Is there a better value for setting up a productivity factor?

The first research question (RQ-1) aims to get an insight on the datasets of this research. We examine the datasets and then the productivity factor of each software project was calculated. This data will be then statistically summarized.

The second research question (RQ-2) aims to produce an evidence that the value for productivity factor (20) is statistically sufficient. Therefore we will perform one sample t-test of this claim.

To address research question (RQ-3), we experimented with simple linear model used for calibrating the productivity factor. To assess the evidences of statistical properties, we used exploratory analysis and hypothesis testing. For all comparisons, the MMRE measure will be used.

3. Experiment

For all models in this study, we used 10-fold cross validation method to assess the reliability of our research. The MMRE is chosen as criteria for all model comparison. Datasets used in this research are described in this section.

There are three datasets for comparison of productivity factor. And we record 5 values for each software project: UUCW, UAW, TCF, ECF and actual effort.

- Dataset from Poznan University of Technology [10] (referred to hereafter as Dataset1)
- Dataset from Subriadi's paper [11] (referred to hereafter as Dataset2)
- Our own dataset collected using document reviews and contributions from software companies (referred to hereafter as Dataset3)

	Dataset1 Value	Dataset2 Value	Dataset3 Value
n	14	10	143
Min.	4.22	11.18	2.18
1st Qu	11.30	12.00	10.39
Median	14.00	15.80	15.33
Mean	15.33	14.29	16.89
3rd Qu.	17.80	15.96	21.29
Max.	35.06	17.30	86.85

Table 1. Statistical comparison of PF in each dataset

Table 1. shows the descriptive statistical comparison of PF in each dataset. The most interesting part of this table is the mean of PF in each dataset. This number is about 15. We can also see some outliers especially for dataset3 (min and max values). We can also note that the median of PF in each dataset is also about 15.

3.1. Simple Linear Regression

In this research, we will utilize a simple linear regression for resolving productivity factor of Use Case Points method. The equation for simple linear regression can be seen on equation (3).

$$\hat{y} = \beta_0 + x \cdot \beta_1 \quad (3)$$

where $\hat{y}$ is prediction (dependent variable), β_0 is intercept, x can be seen as a number of Use Case Points (UCP) and β_1 is productivity factor for Use Case Points method. If we omit the intercept from the equation we get fallowing equation (4)

$$\hat{y} = x \cdot \beta_1 \Rightarrow \hat{y} = UCP \cdot PF \quad (4)$$

where $\hat{y}$ is prediction (dependent variable), x is number of Use Case Points (UCP) and β_1 is productivity factor for Use Case Points method.

4. Results

	CI low	CI high	p-value
Dataset1	10.87	19.78	4.96e-06
Dataset2	12.70	15.88	7.82e-09
Dataset3	15.11	18.67	2.2e-16

Table 2. Summary table for one sample t-test, if the PF is set to 20

In Table 2., the summary for one sample t-test can be seen, if the PF is set to 20. As can be seen in this table, the confidence interval ranges from 10-18. The most important is a fact that on each dataset the PF value of 20 was rejected (p-value). We also noted that the intersection of these entire interval ranges from 15.11 to 15.88.

The confidence interval for dataset1 can be seen in comparison with another as exceptionally wide. This is probably due to wide variety of software projects complexity.

The actual effort for each project in all dataset with linear models can be seen in Fig. 1.. As can be seen on this figure, all three models have very similar slope (the value of this slope can be seen in Table 3.). We also note that the intercept for each model was set to 0.

	PF (slope)	Std. Error	R^2
Dataset1	13.42	1.55	0.85
Dataset2	12.65	0.64	0.97
Dataset3	15.33	0.54	0.85

Table 3. Summary of linear models for each dataset

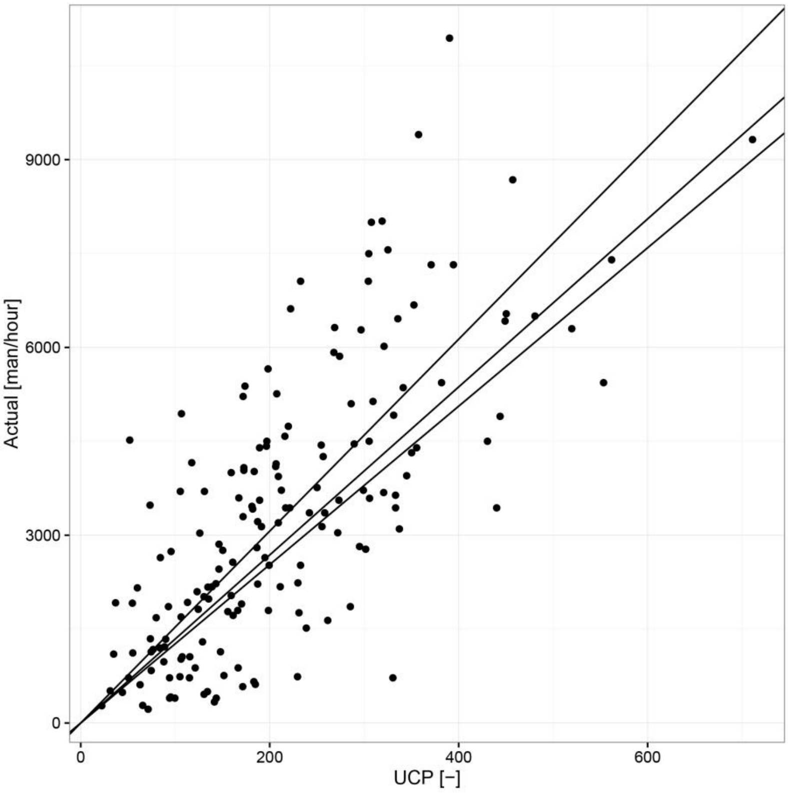

Fig. 1. Linear models on all datasets

In Table 3., the summary of linear models for each dataset is presented. As can be seen, all linear models have an exceptionally good value of R^2. The mean number of slope is about 13.8.

PF	20 MMRE [%]	15.5 MMRE [%]	15 MMRE [%]	Fit MMRE [%]
Dataset1	78	50	48	43
Dataset2	43	15	14	16
Dataset3	96	70	67	69

Table 4. Summary table for calculation of MMRE on chosen PF

As can be seen in Table 4., the summary for calculation of MMRE on chosen PF. We can see from that the standard value 20 for PF yield about 30% MMRE worse result than for value 15. The value of fit means, that we use a value (slope) from linear regression. Also we take note that the fit value is not the best MMRE for all datasets.

5. Discussion

The study started out with goals of answering three research questions outlined in research objective section. These questions are answered in the result section of this paper.

RQ-1: What is the mean value of productivity factor of software projects? This question is answered in result section respectively in Table 1.. The mean value for productivity factor is about 15. What is also important in this table is the values of min and max. These values are probably outliers. Nevertheless, we think that in this case these values are important. Nearly 75% values of PF are below 20. This is very surprising if we take a in account that the 20 is recommended value for productivity factor.

RQ-2: Is appropriate to recommend a standard value of productivity factor as 20? This question is answered in result section. In Table 2., we can see that all PF in our datasets did not exceed the 5% confidence level. Therefore, there is no evidence for supporting PF 20 as a good value. We can see also in Table 4. that the MMRE reached by PF 20 is not as good as for number 15. We can also see in Table 2. that the intersection of confidence interval ranges from 15.11 to 15.88. The number 20 is a relatively distant value from this range.

RQ-3: Is there a better value for setting up a productivity factor? To answer this question, we use 4 values of productivity factor. These numbers are 20, 15.5, 15 and fit value from linear regression. Table 4. shows the comparison of these numbers. The number 20 shows the worst results of MMRE error. Then there are two numbers 15.5 and 15. The

15.5 was chosen because it is an intersection for all confidence intervals. The number 15 was chosen because we would like to know the difference between 15.5 and the nearest natural number. Both values 15.5 and 15 show a difference in MMRE from 20 to about 30%. This can be seen as exceptional improvement from productivity factor set to 20. The fit value shows the slope for linear regression, which is different for each dataset. This value is calculated by least square method and this is also a reason why MMRE does not show the minimal value. We can also see that the number 15 is good choice if we consider only the value of MMRE.

6. Conclusion

Nowadays, 20 is a standard starting value for productivity factor. Also it is recommended use values from 10-30 for productivity factor. The current study found that, we must reconsider these recommendations. In this study nearly 75% of all projects in our dataset have PF value under 20. Moreover, the mean value is about 15. Also our study presented evidences that for all datasets is statistically significant to not set the PF value to 20. In our study, figures prominently range from 10-18. Hence, this is an exceptional difference to recommendation of 10-30. Nevertheless, there is a extensive threat of validity in the use of these datasets especially sample sizes and unknown collection methods. However, the findings of this study have a number of important implications for future research of the productivity factor and for Use Case Point method in general.

7. Acknowledgments

This work was supported by Internal Grant Agency of Tomas Bata University in Zlin under the project No. IGA/FAI/2017/007.

8. References

[1] Keung, J. W. (2008). Theoretical Maximum Prediction Accuracy for Analogy-Based Software Cost Estimation, Software Engineering Conference, 2008. APSEC ’08. 15th Asia-Pacific, pp. 495–502, 2008.

[2] Boehm, W. (1984). Software Engineering Economics, IEEE Transactions on Software Engineering, vol. SE-10, pp. 4–21, jan 1984.

[3] Atkinson, K. & Shepperd, M. (1994). Using Function Points to Find Cost Analogies, 5th European Software Cost Modelling Meeting, Ivrea, Italy, pp. 1–5, 1994.

[4] Karner, G. (1993). Resource estimation for objectory projects, Objective Systems SF AB, pp. 1–9, 1993.

[5] Urbanek, T.; Prokopova, Z.; Silhavy, R. & Vesela, V. (2015). Prediction accuracy measurements as a fitness function for software effort estimation, SpringerPlus, 2015.

[6] Clemmons, R. K. (2006). Project estimation with Use Case Points, CrossTalk, vol. 19, no. 2, pp. 18–22, 2006.

[7] Schneider, G. & Winters, J. (2001). Applying Use Cases: A Practical Guide. Addison-Wesley Professional, 2001.

[8] Silhavy, R.; Silhavy, P. & Prokopova, Z. (2015). Algorithmic Optimisation Method for Improving Use Case Points Estimation, PLOS ONE, vol. 10, nov 2015.

[9] Jeffery, R.; Ruhe, M. & Wieczorek, I. (2000). A comparative study of two software development cost modeling techniques using multi-organizational and company-specific data, Information and Software Technology, vol. 42, pp. 1009–1016, 2000.

[10] Ochodek, M.; Nawrocki, J. & Kwarciak, K. (2011). Simplifying effort estimation based on Use Case Points, Information and Software Technology, vol. 53, pp. 200–213, mar 2011.

[11] Subriadi, A. P. & Ningrum, P. A. (2014). Critical review of the effort rate value in use case point method for estimating software development effort, Journal of Theoretical and Applied Information Technology, vol. 59, no. 3, pp. 735–744, 2014.