Journal of Theoretical and Applied Information Technology
31st January 2014. Vol. 59 No.3
© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

CRITICAL REVIEW OF THE EFFORT RATE VALUE IN USE CASE POINT METHOD FOR ESTIMATING SOFTWARE DEVELOPMENT EFFORT

[1]APOL PRIBADI SUBRIADI, [2]SHOLIQ, [3]PUJI AGUSTIN NINGRUM
[1,2,3]Department of Information System, Sepuluh Nopember Institute of Technology (ITS), Indonesia
E-mail: [1]apol@is.its.ac.id, [1]apolpribadi@gmail.com

ABSTRACT

Effort estimation is an activity to estimate the number of business activities of workers as well as how long it takes to accomplish a software development project. This estimation is very important to be able to know how much the relevant value of software generated. One common method used to calculate the estimated effort is the Use Case Point (UCP).
This research aim to review the UCP method, proposed by Karner in 1993, which is only based on three software development project data. However, until now, most researchers still refer to the value of the proposed Estimate Rate (ER) by Karner without questioning its relevance. In the UCP method, the estimated effort obtained from multiplying the UCP Value by the ER Value. ER Value is the ratio of staff-hours required to accomplish each UCP. Karner proposed ER Value was 20 staff-hours.
The final result of this research, the ER Value is equal to 8.2. This value is much smaller than the ER Value proposed by Karner. This is possible due to several reasons: 1) the existence of software engineering methods, 2) the more advanced software engineering technologies, 3) the software by component, 4) the source availability in the internet.

Keywords: *Effort Rate, Use Case Point, Software Development Effort*

1. INTRODUCTION

The failure rate of software development projects is very high. In the range of 2002 to 2010, there has been only 37% successful information technology development projects (Standish, 2011). The largest cause percentage of the software development projects failure is the lack of good planning of the project, which is about 39% (Bull, 1997; KPMG, 1998). This led to the need for an improvement project planning to perform calculations that better reflect the real situation.

Estimation is a predictive quantitative measurement, where the accuracy is expressed with numbers (Tokey, 2010). Furthermore, software estimation is an activity to predict or forecast the output of a project to review the schedule, cost, risk and also the effort in the project (Galorath, 2008).

Use Case Point (UCP) is a method that has the ability to provide effort estimation (staff-hours) required to make a software (Karner, 1993). UCP method is very useful in measuring the effort estimation, since using use cases as input. Use Case is a popular form of representing the functional requirements in software creation. Neill and Laplante (2003), and Ochodek et al. (2011) revealed that 50% of software development had functional requirements which were presented as scenarios or Use Cases in the early stages of software development.

Khatibi and Jawawi (2010), show conclusively that the UCP method can be used to estimate software development effort and prove that the UCP method is good enough. Comparison of effort estimation based on UCP method to an actual effort has a deviation of 19%, while general estimations have a deviation of 20% (You, 2002), 6% (Nageswaran, 2001) or 9% (Carroll, 2005). However, smaller deviation values based on the UCP method would be possible if the estimated value of effort is also getting smaller.

At UCP method, the effort estimation obtained from multiplying the UCP Value by the Effort Rate (ER) Value. Effort estimation will provide the number of employees (staff or staff-hours-days) required in software development projects (Muhardin, 2011). ER is the ratio of the staff-hours required that used by each Use Case Points and variables to calculate Effort Value (Clemmons, 2006).

Karner (1993) gave the ER Value is 20 staff-hours. The data was derived from three software development projects (later, this value becomes a common reference). Schneider and Winters (1998) proposed the ER Values are 20, 28 and 36 staff-hours based on the complexity of the project with reference to the Technical Complexity Factor (TCF). Clemmons (2006) proposed the ER value is 18 staff-hours based on the quality of team personnel and historical data. Ochodek et al. (2011) provide ER Values ranged from 4 to 35 staff-hours were calculated from 14 software projects. Next the researchers use the most recent value of the proposed ER by Karner (1993), Schneider and Winter (1998), or a combination of both. According to Clemmons (2006) and Ochodek, et al. (2011), if there is no previous software development data, then it was suggested to use the default values proposed by Karner ie 20 staff-hours.

2. LITERATURE REVIEW

2.1 Software Effort Estimation

Important aspect in the planning stages of the project implementation software, is the activity estimation in terms of cost, time and resources. Tokey (2010) defines estimation is a quantitative measurement process that the results accuracy can be stated by the number. Therefore, software estimation defined by Galorath (2008) was an activity to forecast the output of a software development project by reviewing the schedule, effort, cost and risk embedded to the project.

2.2 Use Case Point (UCP) Method

Karner (1993) developed the Use Case Point (UCP) Method as an adaptation of the Function Point Analysis (FPA) Method. The goal is to provide a simple estimation processes adapted to the object orientation in software projects.

The Karner's steps for doing effort estimation with Use Case Point Method (UCP) are as follows:

a. Calculating Unadjusted Use Case Weight (UUCW).
b. Calculating Unadjusted Actor Weight (UAW)
c. Calculating the Technical Complexity Factor (TCF)
d. Calculating Environment Complexity Factor (EF)
e. Calculating Unadjusted Use Case Points (UUCP), where UUCP = UAW + UUCW.
f. Calculating Complexity Factor, where:
   TCF = 0.6 + (0:01*TF)
   ECF = 1.4 + (0:03 * EF).
g. Calculating the Use Case Point (UCP), where:
   UCP = UUCP * TCF * ECF

2.2.1 Calculating Unadjusted Use Case Points (UUCP)

Unadjusted Use Case Points (UUCP) is obtained from the sum of the Unadjusted Use Case Weights (UUCW) with Unadjusted Actor Weights (UAW).

$$UUCP = UUCW + UAW \qquad (1)$$

1) Unadjusted Use Case Weights (UUCW)

The UUCW is one of the factors that contribute to the size of the software being developed. It is calculated based on the number and complexity of the use cases for the system. To find the UUCW for a system, each of the use cases must be identified and classified as Simple, Average or Complex based on the number of transactions the use case contains. Each classification has a predefined weight assigned. Once all use cases have been classified as simple, average or complex, the total weight (UUCW) is determine by summing the corresponding weights for each use case

The total number of Unadjusted Use Case Weights (UUCW) obtained from counting how many use cases of each type (level of complexity) multiplied by the weight of each type, see Table 1.

Table 1. Use Case Type, Weight dan Carry Out

Use Case Type	Weight	Use Case Carry Out
Simple	5	Using ≤ 3 transactions
Medium	10	Using 4 - 7 transaction
Complex	15	Using > 7 transaction

The Formula for UUCW is as below:

UUCW = (Total of Simple Use Cases x 5) +
(Total Average Use Case x 10) +
(Total Complex Use Cases x 15) (2)

2) Unadjusted Actor Weights (UAW)

The UAW is another factor that contributes to the size of the software being developed. It is calculated based on the number and complexity of the actors for the system. Similar to finding the UUCW, each of the actors must be identified and classified as Simple, Average or Complex based the type of actor. Each classification also has a predefined weight assigned. The UAW is the total of the weights for each of the actors.

The next step to get value of UAW is to determine whether the actor as simple, medium or complex.

Total Unadjusted Actor Weights (UAW) obtained from counting, how many actors of each type (level of complexity) multiplied by the weight of each type, see Table 2.

Table 2. Actor Type, Weight and Description

Actor Type	Weight	Actor Description
Simple	1	Interacts through API, as Command Prompt
Medium	2	Interacts through Protocol, as TCP/IP
Complex	3	Interacts through GUI or Web Page

The formula for UAW is as follow:
UAW = (Total No. of Simple actors x 1) +
(Total No. Average actors x 2) +
(Total No. Complex actors x 3 (3)

2.2.2 Calculating Technical Complexity Factor (TCF).

The TCF is one of the factors applied to the estimated size of the software in order to account for technical considerations of the system. It is determined by assigning a score between 0 (factor is irrelevant) and 5 (factor is essential) to each of the 13 technical factors listed in the Table 3 below. This score is then multiplied by the defined weighted value for each factor. The total of all calculated values is the technical factor (TF).

Table 3. Technical Factor Weight

Technical Factor		Weight
1	Distributed System Required	2
2	Response Time Is Important	1
3	End User Efficiency	1
4	Complex Internal Processing Required	1
5	Reusable Code Must Be A Focus	1
6	Installation Easy	0.5
7	Usability	0.5
8	Cross-Platform Support	2
9	Easy To Change	1
10	Highly Concurrent	1
11	Custom Security	1
12	Dependence On Third-Part Code	1
13	User Training	1

TF subsequently used to obtain the value of the Technical Complexity Factor (TCF). The formula given is as follow:

TCF = 0.6 + (0.01 * TF) (4)

2.2.3 Calculating Environmental Complexity Factor (ECF)

The ECF is another factor applied to the estimated size of the software in order to account for environmental considerations of the system. It is determined by assigning a score between 0 (no experience) and 5 (expert) to each of the 8 environmental factors listed in the table below. This score is then multiplied by the defined weighted value for each factor. The total of all calculated values is the environment factor (EF).

Table 3. Environmental Factor and Weight

Environmental Factor		Bobot
1.	Familiarity with the Project	1.5
2.	Application Experience	0.5
3.	OO Programming Experience	1
4.	Lead Analyst Capability	0.5
5.	Motivation	1
6.	Stable Requirements	2
7.	Part Time Staff	-1
8.	Difficult Programming Language	-1

The values on the environmental factor multiplied by the weight value of each, and then summed to obtain Total Environmental Factor (EF). Furthermore EF value will be used to obtain the Environmental Complexity Factor (ECF). The formula given for ECF calculation is as follow:

ECF = 1.4 + (-0.03 * EF) (5)

2.2.4 Calculating the Effort Rate (ER)

Effort rate is the ratio of the number of staff-hours per use case point by the projects in the past. If the project is new and there are no historical data that has been collected, the values should beranging from 15 to 30. However, the value that most often used is 20 (Clemmon, 2006). The formula for Effort Estimation calculation using the UCP method is as follows:

Effort Estimation = UCP x ER (6)

If the ER Value calculated from one project only, the value of ER was given by dividing the actual value of effort with UCP value. The formula is as follows:

Effort Rate (ER) = Actual Effort / UCP (7)

2.3 Regression and Correlation Analysis

2.3.1 Regression

Regression is a dependence analysis to determine the relationship between the dependent variable (Y) with one or more independent variables (X). The goal is to determine the average value of the dependent variable, if the independent variables are known (Usman, 2006). Regression analysis was used to perform prediction from historical data to determine the future effort rate value of software development project. The regression formula is as follows:

$$Y = a + bX \qquad (8)$$

2.3.2 Correlation

Correlation analysis is an analysis to express the strength relationship between the independent variable X with the dependent variable Y. Correlation analysis, commonly used in conjunction with regression analysis, to measure the accuracy of the regression line in explaining the variation in the value of the dependent variable. Linear correlation coefficient is a number of the linear relationship between the variable X with the variable Y, and denoted by "r" (Usman, 2006). Here's the formula for measuring the correlation coefficient:

$$r = \frac{n\sum x_i y_i - (\sum x_i)(\sum y_i)}{\sqrt{(n\sum x_i^2 - (\sum x_i)^2 (n\sum y_i^2 - (\sum y_i)^2)}} \qquad (9)$$

3. PROBLEM IDENTIFICATION

Effort Rate Value (ER) which is used by researchers has variation due to differences in calculation and data availability.

Karner (1993) proposed the ER Value is 20 staff-hours based on three software development project data. Schneider and Winters (1998) proposed the ER Values are 20, 28 and 36 staff-hours based on the complexity of the project with reference to the Technical Complexity Factor (TCF). Clemmons (2006) gave the ER Value by 18 staff-hours on the basis of the personnel team quality calculation and historical data of similar previous software development project. Ochodek, et al. (2011), scored ER Values ranging from 4 to 35 staff - hours which are calculated from previous completely project. Later on, Clemmons (2006) and Ochodek, et al. (2011), suggested to use the ER Values proposed by Karner (1993), if there were not available historical data. This suggestion then becomes a common reference among the researcher to date, as well as carrying major drawback, given by Karner (1993) which was only used three software development project data. Considering to the regression accuracy by using only three discrete data are likely to be in-accurate raises a doubt to the proposed ER Value by Karner (1993). Karner (1993) also did not analyze the data to establish a correlation between the regression equation, so that was a question regarding the level of reliability and validity.

The research question in this research is mainly based on the two shortcomings background above, were overlooked by Karner (1993) in determining the Effort Rate Value. This research question will examine and challenge the used of Effort Rate of 20 which commonly used by current researchers to check the level of compliance.

4. RESEARCH METHODS

4.1 Data Collection

Data were collected in three ways: interviews, questionnaires and documents review.

a) Interview

Data obtained from the interviews were data on the number, duration of project and the number of workers required to work on software projects. The data is then used to calculate the actual value of effort.

b) Questionnaire

The results obtained from this questionnaire are the factors that influence the result of project (technical factors and environmental factors) and value (score) of each factor on software project work.

c) Documents Review

The results obtained from the documents review are the list of Use Case that has been made in the execution of previous software projects and also the number of actors present in the software project. Then the results of this documents review will be used to calculate the Unadjusted Use Case Weight (UUCW) and Unadjusted Actor Weight (UAW).

4.2 Calculation of Actual Effort

After getting the required data from interviews, the next step was the calculation of actual efforts to determine the Actual Value of each project. The formula is as below:

$$\text{Actual Effort} = \sum \text{Workers} \times \sum \text{Times} \qquad (10)$$

4.3 Calculation of Use Case Point (UCP)

Based on questionnaires and documents review that provide data of Use Cases, Actors, Technical and Environmental Factors, then the Use Case

Point (UCP) for each project could be calculated, using the following formula:

$$UCP = UUCP * TCF * ECF \qquad (11)$$

While UUCP, TCF and ECF could be obtained from the steps below:

a) Calculating the Unadjusted Use Case Points (UUCP)

To get the Unadjusted Use Case Points (UUCP) Value, then the weighting and scoring complexity could be implemented with refers to the use cases and actors in terms of the use case diagram. Scoring is calculated based on the parameters that have been determined.

b) Calculating the Technical Complexity Factor (TCF) and the Environmental Complexity Factor (ECF)

To find out Technical and Environmental Complexity Factor, there have been some parameter measurements with accompanying weight. However, objective assessment from project managers or project team is required for giving value to each of these parameters.

4.4 Correlation Analysis and Linear Equations

The output obtained from this phase was to determine the correlation and linear equations between the Actual Value and UCP Value with effort. Next, it will be used to determine the value of the tangent or ER.

Correlation is a statistical technique used to test the presence/ absence of a relationship and the direction of two or more variables. Regression is a continuation of the correlation analysis to examine the extent to which the influence of the independent variable on the dependent variable after the relationship between these variables has been discovered.

Correlation and regression calculations using SPSS was conducted to determine whether the results of the calculations are valid. As well as to search for the free variables which are correlated to each other and determines the form of those relationships.

4.5 Calculating the Value of Effort Rate (ER)

Once the correlation between Actual Effort Value and UCP Value has been known, as well as linear equations, then the next value from the ER of software development project can be determined using the following formula:

$$Tangen\theta(ER) = \frac{\Delta Y}{\Delta X} = \frac{Y_2 - Y_1}{X_2 - X_1} \qquad (12)$$

The other simple way, Effort Rate Value can be associated with "b" at linear regression equation at Formula (8).

5. RESULT

5.1 Population and Sample

This study was directed to the field of software development project in the business purposes. Table 5 shows the project list of software development as research object.

Table 4. Software Development Project List As Research Object

No	Project ID	Project Name/Categories	Technology
1	I	Multy Level Marketing System	JSP, PHP, PostgreSQL
2	II	Sales System	PHP, MySQL
3	III	Education and Training Management	JSP, J-Query, MySQL
4	IV	Electronic Vehicles IDs System	JSP, J-Query, MySQL
5	V	Labor and Workforce System	JSP, J-Query, MySQL
6	VI	On Line Ticketing Management System	JSP, J-Query, MySQL
7	VII	Building Rental System	JSP, J-Query, MySQL
8	VIII	Mall Search Engine System	JSP, J-Query, MySQL
9	IX	Cookies and Food Trading System	JSP, J-Query, MySQL
10	X	Data Dictionary Bank	JSP, J-Query, MySQL

5.2 Actual Effort Value

Actual Value of software project effort is obtained from the results of interviews to the software development project team. The information collected was the number of workers and the amount of time needed to accomplish the software development. Multiplying of workers number by times consumed will result the Actual Effort.

Table 6 shows the actual value for the overall effort of software development in the object.

Table 5. Each Project Actual Effort

No	Project ID	*Actual Effort (staff/hours)*
1	I	3684
2	II	1980
3	III	3950
4	IV	1925
5	V	2175
6	VI	2226
7	VII	2640
8	VIII	2568
9	IX	3042
10	X	1696

5.3 Use Case Point (UCP) Estimation

The following steps are the way how to get UCP Estimation:

5.3.1 Unadjusted Use Case Points (UUCP) Value

To get the Value of Unadjusted Use Case Points (UUCP) should be calculated in advance of Unadjusted Use Case Weight (UUCW) and Unadjusted Actor Weight (UAW).

1) Unadjusted Use Case Weight (UUCW)

Value of Unadjusted Use Case Weight (UUCW) was obtained from a calculation of how many (total) Use Cases of each type (level of complexity) multiplied by the weight of each type.

The UUCW Value calculation of the overall software development projects are presented in Table 7.

Table 6. UUCW Project Value

No	Project ID	UUCW
1	I	355
2	II	145
3	III	325
4	IV	90
5	V	125
6	VI	120
7	VII	200
8	VIII	175
9	IX	245
10	X	140

2) Unadjusted Actor Weight (UAW)

Unadjusted Actor Weight (UAW) Value was obtained from counting how many (total) actors of each type (level of complexity) multiplied by the weight of each type.

The UAW Value calculation of the overall software development projects are presented in Table 8.

Table 7. UAW Project Value

No	Kode Proyek	UAW
1	I	**15**
2	II	**18**
3	III	**12**
4	IV	**6**
5	V	**9**
6	VI	**9**
7	VII	**12**
8	VIII	**9**
9	IX	**12**
10	X	**6**

Once UUCW and UAW Values have been obtained then the Unadjusted Use Case Points (UUCP) Value can be calculated by summing the value of UUCW and UAW. The UUCP Value of the overall software development projects are presented in Table 9.

Table 8. UUCP Project Value

No	Project ID	UUCW	UAW	UUCP (UUCW+UAW)
1	I	355	15	370
2	II	145	18	163
3	III	325	12	337
4	IV	90	6	96
5	V	125	9	134
6	VI	120	9	129
7	VII	200	12	212
8	VIII	175	9	184
9	IX	245	12	257
10	X	140	6	146

5.3.2 Technical Complexity Factor (TCF) Value

Values of these technical factors were obtained from the questionnaire to the software programmer/ developer. Furthermore, the Technical Factors Values are multiplied by the weighting of each factor, and then added together to get the total Technical Factor (TF), which is then used to obtain the value of the Technical Complexity Factor (TCF), see formula (4).

The TCF result base on formula (4), are presented in Table 10.

Table 9. TCF Project Value

No	Project ID	TCF
1	I	1.125
2	II	1.08
3	III	1.095
4	IV	1.02
5	V	1.025
6	VI	1.115
7	VII	1
8	VIII	0.95
9	IX	0.89
10	X	0.965

5.3.3 Technical Complexity Factor (ECF) Value

Environmental Factors Value is obtained from the questionnaire to the developer of the software project. Then the Environmental Factors Values are multiplied by the weighting of each factor, and added together to get the total Environmental Factor (EF). Furthermore this number will be used to obtain the Environmental Complexity Factor (ECF) Value. Using formula (5), ECF Values are presented in Table 11.

Table 10. ECF Project Value

No	Project ID	ECF
1	I	0.77
2	II	0.77
3	III	0.935
4	IV	1.085
5	V	0.98
6	VI	0.995
7	VII	0.92
8	VIII	0.92
9	IX	1.19
10	X	0.755

5.3.4 Use Case Point (UCP)

Referring to the formula (11), the UCP Value can be calculated and presented at Table 12.

Table 12. UCP Project Value

No	Project ID	UUCP	TCF	ECF	UCP
1	I	370	1.125	0.77	320.5
2	II	163	1.08	0.77	135.6
3	III	337	1.095	0.935	354.0
4	IV	96	1.02	1.085	106.2
5	V	134	1.025	0.98	134.6
6	VI	129	1.115	0.995	143.1
7	VII	212	1	0.92	195.0
8	VIII	184	0.95	0.92	160.8
9	IX	257	0.89	1.19	272.2
10	X	146	0.965	0.755	106.4

5.4 Effort Rate (ER) Calculation

To calculate the Effort Rate (ER) in the future practical business environment, based on historical data or empirical data, then the Actual Effort Value and the UCP Value are used to perform correlation analysis and linear regression equation.

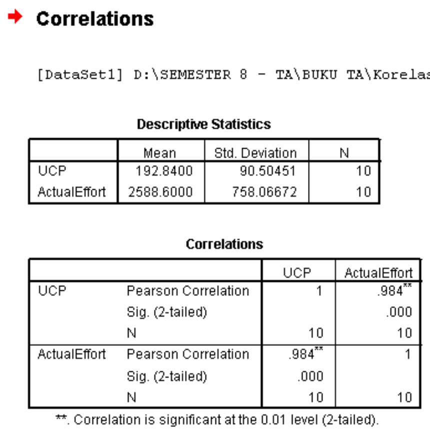
Derived from Table 6 and Table 12, Table 13 presents the tabulation of the Actual Effort Value and the UCP Value that will be used to perform correlation analysis and linear regression equations.

Table 11. Actual Effort and UCP Project Value

No	Project ID	*Actual Effort*	UCP
1	I	3684	320.5
2	II	1980	135.6
3	III	3950	354.0
4	IV	1925	106.2
5	V	2175	134.6
6	VI	2226	143.1
7	VII	2640	195.0
8	VIII	2568	160.8
9	IX	3042	272.2
10	X	1696	106.4

5.4.1 Correlation Analysis

Based on formula (11) and utilizing SPSS to help calculation, the correlation analysis between the actual effort value and UCP value of the research object is presented at Figure 3.



Correlations

```
[DataSet1] D:\SEMESTER 8 - TA\BUKU TA\Korelasi Regresi.sav
```

Descriptive Statistics

	Mean	Std. Deviation	N
UCP	192.8400	90.50451	10
ActualEffort	2588.6000	758.06672	10

Correlations

		UCP	ActualEffort
UCP	Pearson Correlation	1	.984**
	Sig. (2-tailed)		.000
	N	10	10
ActualEffort	Pearson Correlation	.984**	1
	Sig. (2-tailed)	.000	
	N	10	10

**. Correlation is significant at the 0.01 level (2-tailed).

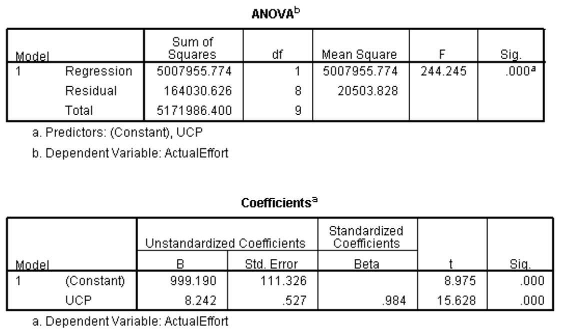
Figure 1: Correlation Analysis between Actual Effort Value and UCP Value

SPSS output shows the Pearson correlation coefficient is 0.984, means that there is a correlation between Actual Effort Value and UCP Value. The correlation between the two variables is very strong as indicated by the correlation value close to +1 and significant at level 0.00 (<0.05). It indicates a significant relationship between the two variables. Positive sign indicates that a correlation exists between the Actual Effort Value with the UCP Value is proportional relationship (Usman, 2006). It can be concluded that the relationship between Actual Effort Value and UCP Value is very strong, significant and direct.

5.4.2 Linear Equations (Regression)

According to formula (8), Figure 4 shows the SPSS output as a basic to form the regression equation between Actual Effort Value and UCP Value for research object.

Regression



ANOVA[b]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	5007955.774	1	5007955.774	244.245	.000[a]
	Residual	164030.626	8	20503.828		
	Total	5171986.400	9			

a. Predictors: (Constant), UCP
b. Dependent Variable: ActualEffort

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	999.190	111.326		8.975	.000
	UCP	8.242	.527	.984	15.628	.000

a. Dependent Variable: ActualEffort

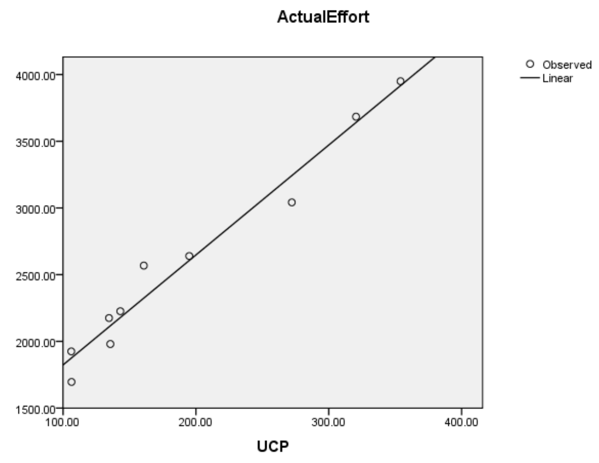


Figure 2: Regression Curve between Actual Effort Value and UCP Value

In this regression analysis, the dependent variable (Y) is Actual Effort Value while independent variable is UCP Value. SPSS output shows that the curve is linear. Based on the ANOVA in Figure 3, the value of F=244.245 and compared with the $F_{(table)}$ (standardize) (df1: 1, DF2: 8) = 5.32, with significant level 0,00 (<0,05). According to Usman (2006), it means that the regression equation is valid and can be used for prediction.

Therefore, based on values of coefficient at Table column B, Constant = 999.190 and UCP = 8,242, then the linear regression equation can be defined as follow:

$$Y = 999.190 + 8.242X \qquad (13)$$

Where: Y = Effort Value, and
X = UCP Value

5.5 Effort Rate Value Affirmation

Once the linier regression equation was obtained, then it is simple to know the Effort Rate Value. According to the section 4.5, the Effort Rate Value can be derived from formula (13) and conclude that in this research object, the Effort Rate Value is 8.242.

6. DISCUSSION AND CONCLUSION.

6.1 Analysis of Effort Rate (ER) Value

Review of Effort Rate (ER) Value on software development in this study, has shown a value of 8.2. This ER Value was much smaller than ER Value default proposed by Karner which was equal to 20. This is possible due to several reasons, as follows:

a) The Development of Software Engineering Methods

Software engineering methods provide the techniques of how to develop software precisely and easily. These methods facilitate the completion of a series of tasks such as project planning and estimation. Software development method or commonly known as the System Development Life Cycle (SDLC) will allow software developers to accommodate multiple needs. Meanwhile the work can be done in a precise pattern and achieved within a specified time, it also helps to become more cost effective/ efficient.

There are many kinds of models in the software development methods, such as waterfall, prototyping, spiral and others. These kinds of model were very possible not popular, not exist or even probably Karner just did not use in his research and proposed the Effort Rate Value was equal to 20.

b) The Development of Software Engineering Technology

Along with the development of information technology, there is also an improvement in advanced tools to provide ease in working software. Limitations or difficulties experienced by programmer to improvise in typing command lines of programming in the past have no longer existed. There is a lot of ease for current programmer to get interaction with computer. Many kind of appropriate software tools were ready to deliver any function need such as designing the look (interface), compiling it, then make the file executable (. exe) and also database design and connection.

The presences of these software tools will be very helpful for programmer/ developer to accomplish the software. For example: there are MS Project for planning, MS Visio or Process Analyst for Need Analysis; Enterprise Architecture, Rational Rose, Power Designer for Face Design; PHP, ASP, Java, and Eclipse, Netbeans, JDeveloper, Visual Studio for coding purposes; MySQL, Oracle, PostgreSQL, and SQLyog, SQL Developer, DBVisualizer for Databases Purposes. At the end, all of these tools will decrease the effort required.

c) Software by component

Currently, the software developers in creating software have also facilitated by using components that are commonly already available freely. Components are composed of classes that are re-usable and iteratively. For example, Microsoft Visual Basic version 5.0 has been providing technology of Microsoft ActiveX components that are intended for use on the internet, intranet and traditional client environment.

Use of these existing software components could deliver an advantageous in terms of cycle time software development. It can reduce the time by 70% and reduced production costs by up to 84%.

d) Google (Internet) Effect.

Google means internet. By now, everything could be found from internet. Google could find almost any sources need to help software development easier. The literature can be obtained from the internet. In addition, the source code can also be searched on the internet, which could facilitate software developers in terms of doing stage activities of software development. As well as the references and best practice of software projects that are similar or have the same type of software being developed. Ease created by the best available examples of experiences and references by free that can streamline the processing time.

With some reasons that have been outlined above, it can be seen that the current effort required is not as difficult as software developers "effort" is needed in the past. It caused by the availability of appropriate technology and support.

6.2 The Appropriateness of Effort Rate (ER) Value

At research conducted by Ochodek in 2011, the calculation of effort estimation was conducted using Use Case Point (UCP). In the same study, the researchers also calculated the value of ER. Ochodek (2011) perform calculations with two opinions, namely: (i) transactions (T) and, (ii) steps (S). Robiolo (2008) identified a "transaction" with the stimulus approach and activities between the actors and the system. While the "steps", without referring to the other use case (include and extend relations), the number of Use Cases prior specifications review are placed in one group.

Furthermore, Use Case calculation with these opinions generated two different ER Values. ER Steps Values tend to be smaller than the ER Transactions Value. In addition, by using 14 software projects as objects, showed that ER Values varied between a value of 3 to 35 (Ochodek, 2011).

At the end, according to the value of ER which was derived by Ochodeck in 2011, it can be concluded that the ER value of 8.242 which is generated in this study is a reasonable value. This value also shows that smaller than value proposed by Karner.

6.3 Effort Estimation relationships with Project Cost Estimation

Based on the Formula (6), Effort Rate (ER) Value is linier to the estimated effort value, so that the smaller the ER Value, it will result in a smaller estimated Effort Value. Scale unit of effort estimation calculations using the UCP method is Hours of Effort.

According to Shaleh (2011), the Hours of Effort was used for the calculation of the estimated software development project cost. If the cost (in dollar) per hour for each effort has been known then it is easy to calculate the Estimated Software Development Project Cost (in dollars) by multiplying the Effort Rate Value to the cost of each effort.

At this point, it can be seen that the lower of Effort Rate Value, it will result the lower of Software Development Cost. The final comparison of Estimation Cost between Karner's Value and this Research's Value, as a simplicity, was as 20 (dollars) compares to 8.2 (dollars).

6.4 Conclusion

The results of this empirical and theoretical review study, the Effort Rate (ER) Value is equal to 8,242. This ER Value is much smaller than ER Value proposed by Karner (1993), which was equal to 20. These are possible several reasons those were ignored by Karner, as follows:

a. Methods of software engineering
b. Software engineering technology
c. Software by component
d. Source availability from the internet

This research has provided the correction of Effort Rate provided by Karner and consequently will impact to the value of estimated project cost lower than usually adopted by project estimator.

7. LIMITATIONS AND FUTURE RESEARCH

However, the objects in this study were not separated between a complex or large software development projects with a simple software development projects. More complex the software, it will create the higher value of effort rate. The study also did not take into account the skills quality of the developers or programmers to build the software. The better of individual programmer quality the faster the project can be completed. Thus, the Effort Rate Value that generated will be smaller. Both of the above reasons will make the differences in Effort Rate Value. Therefore, further research which will accommodate these two factors will give a more precise assessment results and more proportional.

REFERENCES:

[1] Anda, Bente, 2002, *Comparing Effort Estimates Based on Use Cases With Expert Estimates.* In Proceedings of Empirical Assessment in Software Engineering (EASE 2002) (Keele, UK, April 8-10, 2002), p. 13.

[2] Bull Survey, 1998, *Failure Causes.* http://www.it-cortex.com/Stat_Failure_Cause.htm#surveys, Retrieved on 1 June 2013

[3] Carroll, Edward R., 2005, *Estimating Software Based on Use Case Points*. Object-Oriented, Programming, Systems, Languages, and Object Oriented Programming Systems Languages and Applications (OOPSLA) Conference, San Diego, CA, pp.257–265.

[4] Clemmons, Roy K., 2006, *Project Estimation With Use Case Point*. Diversified Technical Services, Inc.

[5] Colin J. Neill and Phillip A. Laplante, 2003, *Requirements Engineering: The State of the Practice*, IEEE Software 20(6) pp. 40-45, Nov-Dec 2003.

[6] Galorath, Daniel and Michael W Evans, 2008, *Software Sizing, Estimation and Risk Management*. Auerbach.

[7] Karner, Gustav, 1993, *Resource Estimation for Objectory Projects.* Objective System SF AB.

[8] Khatibi, V., & Jawawi, D. N., 2010, *Software Cost Estimation Methods: A Review*. Emerging Trends in Computing and Information Sciences, p. 21-29, 2010.

[9] KPMG Canada, 1997, *Failure Causes.* http://it-cortex.com/Stat_Failure_Cause.htm#surveys. Retrieved on 02 Juni 2013.

[10] Muhardin, Endy, 2011, *Estimasi Proyek Software*. http://software.endy.muhardin.com/manajemen/estimasi-proyek-software/. Retrieved on 7 March 2013.

[11] Nageswaran, Suresh, 2001, Test Effort Estimation Using Use Case Points. http://cognizant.com/cogcommunity/presentations/Test_Effort_Estimation.pdf. June 2001.

[12] Ochodek, M., Nawrocki, J., Kwarciak, K., 2011 *Simplifying Effort Estimation Based on Use Case Points.* Sciencedirect, Elsevier.

[13] Robiolo, Gabriela., Orosco, Ricardo., 2008, *Employing Use Case to Early Estimate Effort With Simpler Metrics.* Springer-Verlag London Limited.

[14] Saleh, K., 2011, *Effort and Cost Allocation in Medium to Large Software Development Projects*. Intenational Journal of Computers (1), pp.74-79.

[15] Schneider, G. and Winters, J., 1998, *Applying Use Cases – A Practical Guide*. Addison-Wesley.

[16] Standish Group, 2011, *Chaos Reports*. http://standishgroup.com/news. Retrieved on 01 June 2013.

[17] Tokey, Steve., 2010, *Return of Software: Maximizing the Return on Your Software Investment*. Prentice Hall.

[18] Usman, Husaini, 2006, *Introduction to Statistics*, Bumi Aksara.