UCPabc as an Integration Model for Software Cost Estimation

Renny Sari Dewi, Grandys Frieska Prassida
Department of Information Systems
Universitas Internasional Semen Indonesia
Gresik, Indonesia
renny.dewi@uisi.ac.id, grandys.prassida@uisi.ac.id

Sholiq, Apol Pribadi Subriadi
Department of Information Systems
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
sholiq@is.its.ac.id, apol@is.its.ac.id

Abstract—**This study aims to construct a scientific model in estimating effort and cost of software development projects. Use Case Points (UCP) is very important method to estimate the total effort in software development projects. While the technique of Activity-based Costing (ABC) serves as the calculation of costs in each of the activities, especially the allocation of project resources. ABC technique consists of five stages are estimates based on the allocation of its resources, by having the total of UCP first. The result of this research is a proposed model of integration between UCP and ABC method or also called UCPabc. UCPabc model testing performed on five software development project small-medium scales. The results of test evaluations UCPabc estimation model with the company's actual cost turned out to be quite impressive, which the deviation only 2.16 percent. Meanwhile, the company forecast profit margin of 30.4 percent between total project contract value with cost estimation by using UCPabc.**

Keywords—use case points; activity based costing; integration model;UCPabc; software cost estimation; software effort estimation

I. Introduction

A surprising fact that every year the number of spending on information technology continues to increase. Through research institute, International Data Center (IDC) Indonesia, shopping budget for information and communication technology (ICT) in 2016 is predicted to grow 8.3% which is around IDR 197.4 trillion [1]. The increase in ICT spending was in line with the government's commitment in terms of financial transparency and developing the digital economy.

McKinsey researchers [2] state that the cause of the failure of large-scale information technology projects, particularly software, by 66% because of a budget overrun, 33% for overtime, and the remaining 17% because of less value / benefit. The same research from Standish Group [3], until 2015 there was a total of 71% said information technology projects fail. It is caused by several factors, including the lack of support from the executive, organizational maturity, user involvement, to ambiguous objectives of the company. One form of executive support is the ability to estimate the resources that will be used during the project. In Indonesia, effort estimation methods of software development projects are not too special attention. Based on the observation, only the principle of fairness is used during the determination of the cost of software development.

There are many methods of calculation of estimated effort and cost of software development. The method has been commonly known are Function Points Analysis (FPA) [4], Constructive Cost Models (COCOMO) [5], Use Case Points (UCP) [6], etc. Each method certainly has its advantages and disadvantages, which it depends on the conditions of the project developer software.

Most software developers in Indonesia always start the project by formulating the needs of its users. Then the user needs analysis set forth in narrative form use case. Seeing the condition that way, of course, UCP most appropriate method to be applied in the calculation of the estimated effort and cost of software.

UCP method is used for the first time in 1993 by Gustav Karner [6], but over time, UCP become increasingly popular for the calculation of the estimated effort and cost of software as evidenced by the many studies that UCP response was quite good. The study include the following: (i) UCP better than expected by experts, UCP deviation is 19% whereas the experts deviation of 20% [7]; (ii) UCP deviation of about 9% when tested for the estimated 200 software development projects [8]; and (iii) UCP deviation "only" 6.89% on software development projects of small-medium scale [9].

While other studies of UCP variants include: (i) UCPm is UCP modifications to software development in maintaining context [10]; (ii) e-UCP is UCP modification by adding a use case narrative [11]; (iii) the fuzzy-UCP is UCP modification by collaborating with fuzzy theory [12] [13]; and (iv) iUCP is modified UCP by combining software development with Human Computer Interaction (HCI) [14].

In this study, the resource sharing in the enterprise software developers is being considered. The resource in question is not only focused on the effort of software development, but also considers the resource activity for software development projects underway.

In fact, one of the weaknesses of UCP only calculates the estimated effort is intact, so it cannot be distributed based on the activity of its resources. In addition, the UCP another deficiency is the value of the effort is not associated with the optimization of its resources.

On the other hand, financing techniques, Activity-Based Costing (ABC) has been widely applied in the calculation of

978-1-5090-1721-8/16/$31.00 ©2016 IEEE

the estimated cost of application of information technology. The other study: (i) the evaluation of e-commerce using the ABC [15]; (ii) the calculation of the cost of implementation of information technology in companies [16] [17]; and (iii) 2.16% deviation between the calculation using the UCPabc software costs estimation with actual costs [18].

From the above description, a model of integration between UCP and ABC methods is proposed based on the assumption that if these two methods are combined, it will obtain more detailed results of the calculation of estimated effort and the cost of the software development projects based on resource allocation.

II. Literature Review

A. Related Research

Until now, many studies about UCP since it was introduced in 1993 by Karner. Several previous studies about the UCP, such as:

a) UCP estimations with better than expected by experts, the UCP deviation is 19% whereas the experts' deviation of 20% [7].
b) UCP deviation of about 9% when tested for the estimated 200 software development projects [8],
c) UCP deviation of only 6.89% on 4-scale software development projects of small-to-medium [9].
d) whereas for the modification of the UCP there are several studies, namely: (i) maintain UCP (UCPm) ie UCP modifications to software development in the context of care [10]; (ii) e-UCP UCP is modified by adding a narrative use case [11]; (iii) the fuzzy-UCP UCP is a collaboration with fuzzy theory [12] [13]; and (iv) that is modified UCP iUCP by combining software development and interface design calculations (Human Computer Interaction, HCI) [14].

While previous studies about ABC are:

a) evaluation of e-commerce using the ABC [15],
b) calculation of the cost of implementation of IT / SI in companies [16] [17],
c) 2.16% deviation between calculation using the UCPabc cost estimation with actual costs [18].

B. Use Case Points

In the calculation of the estimated effort of software development, Karner [6] states that the UCP has 3 core stages:

1) Stage 1: Count the Unadjusted Use Case Points (UUCP)

To obtain the value UUCP, the first step is calculating the Unadjusted Use Case Weight (UUCW) and Unadjusted Actor Weight (UAW). UAW used to determine the weight of the actors, while UUCW used to calculate the weight and type of use case narratives. Roundtrip method [19] is used to determine the type of use case based on the number of transactions. UAW in equation (1) is based on Table I.

TABLE I. Actor Weight Classification

Actor Type	Description	Weight
Simple	If the actors interact with the Application Programming Interface (API) as the Command Prompt	1
Medium	If the actors interact with protocols (TCP / IP)	2
Advance	If the actors interact through Graphical User Interface (GUI)	3

$$UAW=(\Sigma Simple\ Actor*1)+(\Sigma Medium\ Actor*2)+(\Sigma Advance\ Actor*3) \quad (1)$$

UUCW in equation 2 is based on Table II.

TABLE II. Use Case Weight Classification

Use Case Type	Description	Weight
Simple	Number of transaction(s) < 3	5
Medium	Number of transactions between 4-7	10
Complex	Number of transactions > 7	15

$$UUCW=(\Sigma Simple\ Use\ case*5)+(\Sigma Medium\ Use\ case*10)+(\Sigma Complex\ Use\ Case*15) \quad (2)$$

Thus, UUCP (3) obtained by adding (1) and (2).

$$UUCP=UAW+UUCW \quad (3)$$

2) Stage 2: Count the Environmental and Technical Complexity Factor

The next stage is to calculate the value of the complexity of external factors (environmental and technical). Environmental Factor (EF) has eight aspects (4), while the Technical Factor (TF) has 13 aspects (5).

$$TCF=0,6+(0,01*\Sigma TF) \quad (4)$$

$$ECF=1,4+(-0,03*\Sigma EF) \quad (5)$$

3) Stage 3: Count Total Effort

The final stage is to calculate the total value of UCP (6) before obtaining its effort by multiplying UUCP (3), TCF (4), and ECF (5). The total effort (7) is obtained by multiplying the value of UCP and Effort Rate (ER) that amounted to 8.2 man/hour [20].

$$UCP=UUCP*TCF*ECF \quad (6)$$

$$Total\ Effort=UCP*ER \quad (7)$$

Overall, three stages UCP and five steps ABC presented in Fig. 1 and Fig. 2.

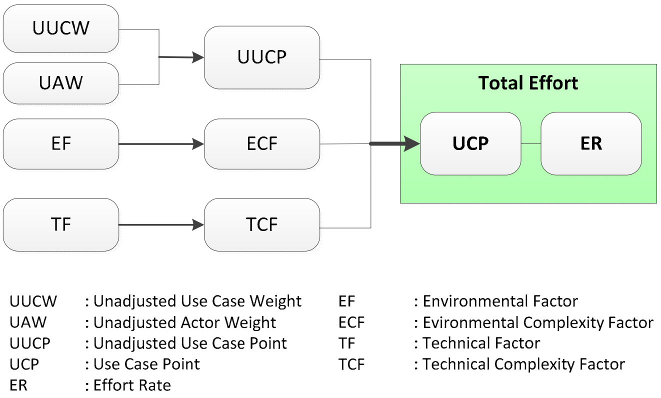
UUCW
UAW
UUCP
EF
ECF
TF
TCF
Total Effort
UCP
ER
UUCW : Unadjusted Use Case Weight
UAW : Unadjusted Actor Weight
UUCP : Unadjusted Use Case Point
UCP : Use Case Point
ER : Effort Rate
EF : Environmental Factor
ECF : Evironmental Complexity Factor
TF : Technical Factor
TCF : Technical Complexity Factor

Fig. 1. Summary Method of UCP

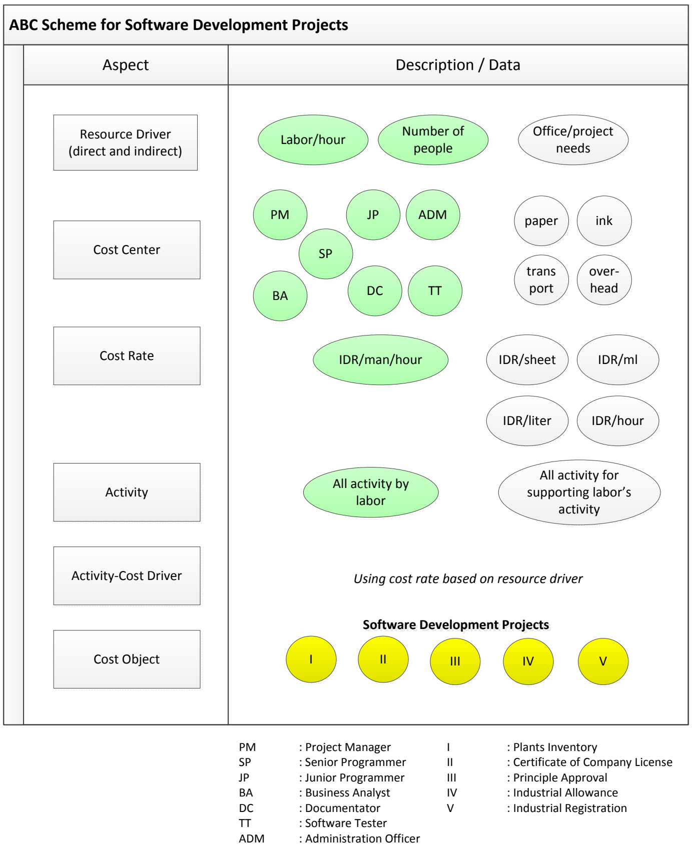

Fig. 2. ABC Scheme for Software Development Project

C. *Activity-Based Costing*

There are five important steps in the Activity-based Costing (ABC) to estimate costs, such as:

1) Step 1: Identify direct and indirect resource drivers

Resource driver on the company is divided into two, direct resource and indirect resource. The direct resource driver is number of employees. While the indirect resource driver such as office supplies, internet, electricity, transportation, sanitation, and sales commissions.

2) Step 2: Assign resources to be cost rate

Imposition of resources towards the financing was conducted to determine the allocation of costs in each of them (direct and indirect). The direct resource cost rate is IDR/person/hour. For indirect resource cost rate to be varied: IDR/hour, IDR/m^3, IDR/month, or IDR/sheet).

3) Step 3: Identify activities

The next step is identifying the activities undertaken by the direct and indirect resources. Some examples of activities undertaken in the direct resources are: survey to the informant, coordination meetings, and others. While activity in the indirect resources are: purchasing office equipment, transportation costs reimbursement, and overhead costs.

4) Step 4: Assign activities into cost drivers

Once the details of the activities carried out, it is done loading activity based cost driver. Cost drivers can be adapted to the resource drivers that have been formulated previously. Some cost drivers identified from the direct resources are the number of work hours. The indirect resource's cost driver are: office supplies (e.g paper), how many pieces/month; how many total quota/hour of internet data usage/month.

5) Step 5: Calculate and classify into cost objects

The last step is calculating the total of each cost drivers, and then grouped into each cost object, in this case the cost of production for any software development projects.

III. Research Method

Based on the above, there are six main phases to integrate two methods (UCP and ABC) for counting software cost estimation (also called as an integration model UCPabc).
The explanation of each phase are::

A. *Analyze the problem for constructing the proposed models*

The weakness of the Use Case Points method is analyzed whether or not being able to be distributed into the use of resources (see Fig. 1). On the other hand, ABC financing technique is well known in terms of the distribution costs into their respective resources. Both of these methods if integrated will bring a new variant of UCP method, namely UCPabc.

B. *Gather Data from Several Software Development Projects*

Primary data obtained by spreading questionnaire to interview several project managers. While secondary data is from the annual financial report which has been summarized in general classification by the administration officer.

C. *Count Effort (UCP's Result)*

Total effort can be gathered by stage-by-stage Use Case Points (Section 2.2). If UCP has obtained its value, then the next is multiplied it with Effort Rate of 8.2 [20]. Therefore, each project has an estimated software development effort which proportioned to be 100%.

D. *Analyze Resource, Activity, and Cost Rate (ABC's Result)*

Based on the general classification report, the resources, activities and cost rate for software development project can be specified as in Table III.

TABLE III. Identified Resource, Activity, and Cost Rate for Software Development Project

Resource	Activity	Unit Cost Rate
Labor	Survey, meetings, configuring	IDR/man/hour
Overhead	Internet, electricity	IDR/month

Resource	Activity	Unit Cost Rate
Office Equiptment	Paper consumption	IDR/month
	Ink consumption	IDR/month
Sales	Sales commission	IDR/year

E. Distribute UCP Total Effort Estimation Based on Product Relative Weight (PRW)

The total estimated effort that is obtained from the calculation UCP, will be distributed on each activity undertaken by its resources. Distribution of effort based on the PRW produced by the company. Then PRW will be the decider for all project resources' complexity.

F. Determine Cost Object

After the cost rate, total effort, and PRW has been declared, then the next is determining the cost object, which means the cost that load to each software development project. By getting cost object, the company can estimate how much profit margin will be reached.

IV. Result and Discussion

A. Proposed Model

The proposed model to integrate the UCP method and ABC techniques shown in Fig. 3. The total effort estimation which is an output of the UCP method was added to the three main components in the ABC technique, such as: the identification of resources and activities, cost rate for resource and activity, and the product relative weight. In the end, UCPabc generates cost object which can estimate cost of each software development project.

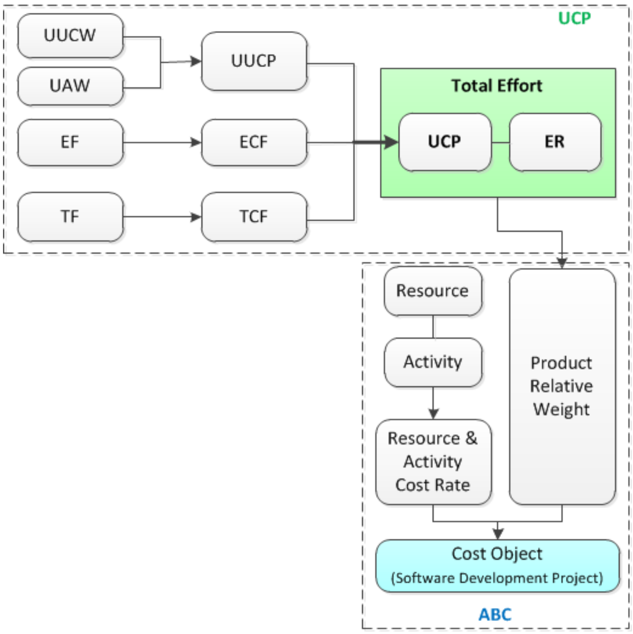

Fig. 3. The Proposed Integration Model of UCP and ABC for Software Development Cost Estimation

As described in the previous chapter, the lack of UCP method is the resulting effort is the whole value. Therefore, the interviews results with some project managers and annual financial report [18] were used as a reference for the activity of each resource in determining the complexity of software development projects.

Details of the activity obtained from the survey results, regrouped into three main phases [21]: software development phase, quality assurance and testing phase, and ongoing activities. Each phase has its own percentage, such as 42% of software development, 37% of quality assurance and testing phase, and 21% for ongoing activities.

B. Testing and Validate UCPabc Model

Testing and validation aims to determine the percentage of deviation between the actual costs with UCPabc cost estimation model. Actual cost is based on the financial statements of small-medium scale company that handles five software development projects (see Fig. 2 and Table IV).

TABLE IV. Detail Software Development Project

ID	Project Name	Project Value (IDR)	Time (month)
I	Plants Inventory	46.900.000	2
II	Certificate of Company License	91.500.000	3
III	Principle Approval [a]	46.800.000	2[a]
IV	Industrial Allowance [a]	47.080.000	3[a]
V	Industrial Registration	44.300.000	2
Total		276.580.000	10

[a.] Parallel working

The first stage is to summarize the results of calculations of the UCP and the estimated effort for five software development projects (see Section 2.2 and Fig. 1). The UCP value is multiplied by Effort Rate for software development projects 8.2 man/hour [20] (see Table V).

TABLE V. Total Amount for Five Software Development Projects

ID	UCP	Effort	PRW
I	171,08	3.775,92	8.19%
II	501,56	3.878,53	24.01%
III	482,45	3.956,07	23.10%
IV	472,99	4.112,77	22.65%
V	460,48	1.402,86	22.05%
Total	**2.088,56**	**17.126,15**	**100.00%**

According to Fig. 3 and Table V, the Product Relative Weight (PRW) become a multiplying factor in the distribution of the estimated component cost drivers for each software development (resources and activities). Therefore, the resource cost drivers are divided into two, namely labor and overhead, which is each resource has own activity and cost driver.

After obtaining the estimated value of effort on five software development project, the next step is to distribute the effort in each activity (see Table VI). Imposition percentage refers to the previous research [21].

ABC in financing techniques, resulting distribution of effort against five software development projects (see Table VI) and then classified according to the cost rate per hour (resource and activity). Wage determination direct resources (labor) based on the wage index has been frequently used by Kelly Services Indonesia [22]. While the determination of resource cost rate refers to the company's financial statements (see Table VII for summary of activity and resource cost rate).

The last phase in ABC financing techniques which determine the cost object is the cost estimate for any software development project based on its PRW (see Table V). At this stage, the cost estimation of software development projects using UCPabc model were examined and compared with actual costs by cost of its object (shown in Table VIII).

In Table VIII, despite the gap deviation costs vary by its activity, the percentage deviation between overall actual cost and cost estimation of using UCPabc amounted to 2.16%. In Table IX indicate when the deviation between project contract value with a total estimated cost of using UCPabc was 30.4% (IDR 84.104.510), which is a make-sense profit margin for the company in five of software development projects.

TABLE VI. EFFORT DISTRIBUTION FOR FIVE SOFTWARE DEVELOPMENT PROJECTS

Activities/Phases	%	Project ID (Effort)				
		I	II	III	IV	V
Requirement analysis	7.5	105,21	308,46	296,71	290,89	283,19
Specification	7.5	105,21	308,46	296,71	290,89	283,19
Design	10	140,29	411,28	395,61	387,85	377,59
Implementation	10	140,29	411,28	395,61	387,85	377,59
Test Integration	7.5	105,21	308,46	296,71	290,89	283,19
Acceptance and deployment	7.5	105,21	308,46	296,71	290,89	283,19
Evaluation & testing	20.84	292,36	857,10	824,45	808,29	786,90
Project management	8.34	117,00	343,01	329,94	323,47	314,91
Configuration management	4.16	58,36	171,09	164,57	161,35	157,08
Quality assurance	8.34	117,00	343,01	329,94	323,47	314,91
Documentation	4.16	58,36	171,09	164,57	161,35	157,08
Training & technical support	4.16	58,36	171,09	164,57	161,35	157,08
Total of Effort	**100%**	**1.402,86**	**4.112,77**	**3.956,07**	**3.878,53**	**3.775,92**

🟩 Software Development 🟨 Evaluation and Testing ▨ Ongoing Activity

TABLE VII. ACTIVITY AND RESOURCE COST RATE

Activities	Activity Cost Rate / hour (IDR)	Resources	Resource Cost Rate (IDR)
Requirement analysis	22.727	Project Manager (man/hour)	39.773
Specification	22.727	System/Business Analyst (man/hour)	22.727
Design	22.727	Software Tester (man/ hour)	22.727
Implementation[b]	20.454	Senior Programmer (man/hour)	28.409
Test Integration	22.727	Junior Programmer (man/hour)	17.045
Acceptance and deployment	14.204	Documentator (man/hour)	11.364
Project management	39.773	Electricity (per KwH)	1.111
Configuration management	22.727	Internet (per hour)	528
Quality assurance	39.773	Cleaning service (per month)	17.045
Documentation	11.364	Transport (times)	10.000
Training & technical support	22.727	Sales Commission (%)	16,16
Evaluation & testing	39.773	Office Equipment (per month)	40.000
		Paper (per month)	219
		Ink (per month)	200.000
		Meal	20.000
		Add-on (per US$)	12.000
		Other equipment	100.000

TABLE VIII. COST AND DEVIATION PER ACTIVITY

Activities		Cost Estimation with UCPabc (IDR)	Actual Cost (IDR)	Deviation
1.1	Requirement analysis	3.094.020	2.850.000	7.89%
1.2	Specification	10.248.131	8.840.000	13.74%
1.3	Design	10.111.856	8.060.000	20.29%
1.4	Implementation[b]	38.664.760	33.205.000	14.12%
1.5	Test Integration	7.695.139	10.190.000	-32.42%
1.6	Acceptance and deployment	6.575.861	7.410.000	-12.68%
2.1	Evaluation & testing	2.993.497	4.750.300	-58.69%
3.1	Project management	95.496.487	95.456.784	0.04%
3.2	Configuration management	2.875.278	2.270.000	21.05%
3.3	Quality assurance	3.032.727	3.408.400	-12.39%
3.4	Documentation	8.514.040	7.170.000	15.79%
3.5	Training & technical support	3.173.692	4.712.000	-48.47%
Total		**192.475.490**	**188.322.484**	
Average Deviation				**2.16%**

[b] Implementation activity involved 40% effort for junior programmer and 60% for senior programmer

TABLE IX. AMOUNT DEVIATION AND COST OBJECT EACH SOFTWARE DEVELOPMENT PROJECT

ID	Product Relative Weight	UCPabc Cost Estimation (IDR)	Project Value (IDR)	Deviation (IDR)	%
I	8.19%	15.763.743	46.900.000	31.136.257	66.4
II	24.01%	46.213.365	91.500.000	45.286.635	49.5
III	23.10%	44.461.838	46.800.000	2.338.162	5.0
IV	22.65%	43.595.698	47.080.000	3.484.302	7.4
V	22.05%	42.440.846	44.300.000	1.859.154	4.2
Total	**100.00%**	**192.475.490**	**276.580.000**	**84.104.510**	
Average Deviation (%)					**30.4**

V. CONCLUSION

Based on the results of research conducted, it can be concluded that: (i) integration between Use Case Point and Activity-Based Costing (then namely UCPabc model), is able to produce software cost estimation, especially using resource sharing system; (ii) a software development company that handle 5 small-medium scale projects in 2013 is interviewed for testing the model; (iii) the percentage deviation per activity of three phases between actual costs and cost estimation using UCPabc model amounted to 2.16%; (iv) cost object of five software development projects have average deviations (as good profit margin) 30.4%, between total projects contract value and cost estimation using UCPabc model.

On the other hand, the UCPabc model still need advance testing if the software development company has limited number of employees and implement resource sharing (one labor doing many projects and become more than one subject/role). As a challenge in the future, UCPabc model can be tested on enterprise-scale software development projects with multi-years funding.

ACKNOWLEDGEMENT

The material contained in this paper has received permission from the related company. Thanks to the research institution (LPPM) UISI which has provided funding and a travel grant to publish this research.

REFERENCES

[1] Indotelko, “Indotelko.com,” 20 Dec 2015. [Online]. Available: http://www.indotelko.com/kanal?c=in&it=belanja-tik-indonesia-tumbuh-8-3-2016. [Accessed June 29, 2016].

[2] S. B., J. L. & Michael Bloch, “Delivering large-scale IT projects on time, on budget, and on value,” McKinsey.com, Tel Aviv, 2012.

[3] S. Group, Standish Group, 04 Oct 2015. [Online]. Available: https://www.infoq.com/articles/standish-chaos-2015. [Accessed June 29, 2016].

[4] A. Albrecht & J. Gaffney, “Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation,” *IEEE Transactions on Software Engineering,* vol. 9, no. 6, pp. 639 - 648, 1983.

[5] B. Boehm et.al, Software Cost Estimation With COCOMO II, New Jersey: Prentice Hall, 2000.

[6] G. Karner, “Resource Estimation for Objectory Projects, Objective Systems SF AB,” 1993.

[7] B. Anda, “Comparing effort estimates based on use cases with expert estimates,” in *Proceeding of Empirical Assessment in Software Engineering (EASE)*, United Kingdom, 2002.

[8] E. Carrol, “Estimating Software Based on Use Case Points,” in *OOPSLA '05 Companion to the 20th annual ACM SIGPLAN conference*, 2005.

[9] Sholiq, A. P. Widodo, T. Sutanto dan A. P. Subriadi, “A Model to Determine Cost Estimation for Software Development Projects of Small and Medium Scales Using Use Case Points,” *Journal of Theoretical and Applied Information Technology,* vol. 85, no. 1, pp. 87-94, 2016.

[10] S. Diev, “Software Estimation in the Maintenance Context,” *ACM,* vol. 31, no. 2, pp. 1-8, 2006.

[11] K. Periyasamy & A. Ghode, “Cost Estimation using extended Use Case Point (e-UCP) Model,” *IEEE,* 2009.

[12] F. Wang, X. Yang, X. Zhu & L. Chen, “Extended Use Case Points Method For Software Cost Estimation,” *IEEE,* pp. 1-5, 2009.

[13] J. Lee, L. Wen-Tin & J. Kuo, “Fuzzy Logic as a Basic for Use Case Point Estimation,” *IEEE,* pp. 2702-2707, 2011.

[14] N. J. Nunes, “iUCP - Estimating Interaction Design Projects with Enhanced Use Case Points,” in *TAMODIA*, Heidelberg, 2010.

[15] N. Roztocki, “Activity-Based Costing for E-Commerce,” in *Proceedings from the Industrial Engineering*, 2001.

[16] R. Neumann et.al, “Cost Management Using ABC for IT Activities and Services,” *Management Accounting Quarterly,* vol. 6, no. 1, pp. 29-40, 2004.

[17] J. Cleary, T. Gerdsen & L. O’Brien, “The Activity Based Costing of Library and Information Technology Services in Higher Education,” 2006.

[18] R. S. Dewi, A. P. Subriadi & Sholiq, “Use Case Point - Activity-Based Costing: New Method for Estimating Software Development Cost,” *Jurnal Sisfo,* vol. 5, no. 1, pp. 142-148, 2015.

[19] I. Jacobson, M. Christorson, P. Jonsson & G. Overgaard, “Object-Oriented Software Engineering. A Use Case Driven Approach,” Addison-Wesley, 1992.

[20] A. P. Subriadi, Sholiq & P. A. Agustiningrum, “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.

[21] K. Saleh, “Effort and Cost Allocation in Medium to Large Software Development Project,” *International Journal of Computer,* vol. 5, no. 1, pp. 74-79, 2011.

[22] KellyServices, “Kelly Services Indonesia Salary Guide,” 2013. [Online]. Available: http://kellyservices.co.id/ID/Resource-Centre/Salary-Guide/.