Software Effort Estimation Using Early COSMIC to Substitute Use Case Weight

Renny Sari Dewi
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
Email: renny.rsd@gmail.com

Riyanarto Sarno
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
Email: riyanarto@if.its.ac.id

***Abstract*—In the size estimation software, there are many methods that have proven their reliability. One of them is Use Case Points (UCP). UCP has a well-known advantage based on the use case scenario which is a reformation of the user story in the software requirements specification (SRS) document. However, UCP also has several weaknesses, including the use case is a summary of the user story. User stories often do not reveal detailed data. Therefore, the potential ambiguity of the use case must be watched by a business/system analyst. On the other hand, there is an international association called COSMIC, which has developed a global standard for calculating the size of the software namely ISO/IEC 19761. The COSMIC model begins with a user story which is then carried out cascade to sequence diagrams to make an engagement between process/method flow and data. The purpose of this study is to substitute the use case weight of the pure UCP method, to become a COSMIC functional size unit (Cfsu). Then, the estimation results of the two are compared with the actual effort. The case study used as a comparison of the COSMIC and UCP methods is the Hair Salon Online Booking Application. From this study the results obtained are the deviation between the results of the original UCP estimate (keep use case weight) of the actual effort is 76.85 percent. As for software effort estimation using early COSMIC is 92.67 percent against the actual effort.**

Keywords—COSMIC, use case points, use case weight, software estimation, software measurement

I. INTRODUCTION

In the 21st century, there are various modern ways of developing software. Generally, there are 3 phases in software development namely planning, implementation, and evaluation. At all stages, many professionals and researchers have published international standards. One of them is the international standard related to software size estimation in the planning phase.

Starting from 1979, A.J. Albrecht published the first time a method for measuring the scale of software called Function Points (FP) [1]. Through the International Function Points User Group (IFPUG), the FP method is upgraded to ISO/IEC 20926:2009 standard [2]. The FP method revolution also made various other global standards such as ISO/IEC 20968:2002 Mk II FPA [3].

In addition to the FP method, Cost Constructive Model (COCOMO) I and II have also been tested by previous researchers. Since it was first published by Barry W. Boehm [4], COCOMO II received a lot of welcome from many researchers to test the accuracy of its estimated value compared to other methods, as Sarno et al [5] [6] [7] [8].

At the end of 2019, the Common Software Measurement International Consortium (COSMI) launched a version version 4.02 to calculate the estimated software development effort. The COSMIC model approaches 4 types of transactions, Entry (E), eXit (X), Read (R), and Write (W). COSMIC is often used by researchers by describing the Unified Modeling Language (UML) sequence diagrams [9].

Use Case Points (UCP) is one method of calculating software size estimates that was discovered by Gustav Karner in 1993 [10]. UCP has applied various professional cost / size estimators for their accuracy in actual effort. There is a different point of view between the COSMIC and UCP methods. The COSMIC model is a hybrid of FP methods.

As is well known, FP has been standardized by international associations with various types of practical tests in thousands of industries and also algorithmically [11] [12]. Therefore, COSMIC as a respectable association, made another standard, COSMIC Full Function Points (FFP) with ISO / IEC 19761: 2003 version 4.02 [13] and simplified to version 5.0 [14].

While UCP also has the advantage of input parameters based on the description of use cases and actors involved in the system. But for the COSMIC model, the advantage is that it can be clearly described the involvement of data in each process / method flow through sequence diagrams in UML approach [9].

II. STATEMENT OF THE PROBLEM

A. Research Problem

The above description becomes the background of the writer to describe the formulation of the problem as follows:

RP1: Can the use case weight in the UCP method be replaced by early COSMIC?

RP2: What percentage of accuracy is generated at the estimated effort based on the early COSMIC method with UCP?

The two problems above aim to find out more whether changing the use case weight to a COSMIC functional size unit in the UCP method affects the level of accuracy of the real project effort.

B. Research Limitation

The limitations of this research are the case studies used by the authors. The case study specified is a Hair Salon Online Booking application development project. This application is done by 1 advanced programmer plus database designer, 1 layout designer also as a project leader. Work on this application for 12 days according to client's request. The software development life cycle method used is agile programming until the application testing period is complete.

III. PREVIOUS WORKS

A. Modifying UCP Method

The original UCP method by Karner [10] was illustrated in Fig. 1.

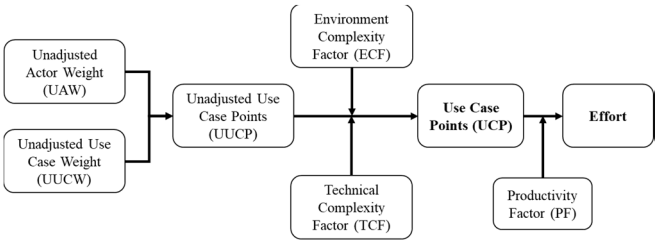

Fig. 1. The Original UCP Method

Since the UCP method was introduced, many researchers have tested the accuracy of the UCP method in various ways. The research in question is:

- Ochodek et al [15] [16] identified the use case and then simplified the UCP method by eliminating unadjusted actor weight (UAW) (see the position in Fig. 1) because it had a low significance to the estimated value of effort. But then Ochodek tried to do automatic use case weight calculations (simple/average/complex) based on the use case names [17].
- Dewi et al [18] integrated the well-known financial accounting model, Activity-Based Costing (ABC) with the UCP method. The results obtained are the accuracy of the UCPabc integration model to the actual effort of about 6.8% [19] [20].
- Nunes et al [21] changed the actor weight classification based on the interface displayed. This was also done by Jha and Malu [22] by modifying parameters in the UCP method.

From the above research, it can be concluded that the modification of the UCP method has been done before.

B. Early COSMIC to Estimate Functional Size Unit

The COSMIC model was first published in 1999. A few years later, the COSMIC model evolved to become a global standard namely ISO/IEC 19761: 2003. Some researchers have tested COSMIC models to get functional software size measurement. Not a few studies that mention the integration of the COSMIC model with other methods, such as Function Points (FP).

- Fehlmann and Kranich [9] performed a cascade of COSMIC models with user stories which were then described as sequence diagrams. This study aims to determine the type of transaction that is Entry (E), eXit (X), Read (R), or Write (W) in the Ticketing Online case study.
- Lavazza et al [23] studied the COSMIC analogy of a simple model through the case of using a rice cooker. This research then continues on the Lavazza real software project [24], [25] based on using 2 methods at the same time.
- The research of Abdullah et al [26] also performed a COSMIC parable in the angry birds (children's game application).

From the literature study above, COSMIC proves it can be used to estimate the size of various types of software, such as business applications and games. This paper also deals with the research of Gencel et al [27] to compare the estimated values obtained from the UCP and Early COSMIC methods with deviations reaching 50%. The difference with this study is that it does not include early COSMIC in the UCP method to replace the unadjusted use case weight (UUCW) component (see Fig. 1).

IV. METHODS

Basically, this research aims to replace the UUCW component into an early COSMIC or COSMIC functional size unit (Cfsu). There are 6 steps as illustrated in the proposed model Fig. 2.

A. Proposed Model

Based on the literature that has been well studied, the author has a track record for modifying existing methods. In this study, the authors intend to make modifications to the Use Case Weight which is one of the weaknesses of the UCP method. The ambiguity that arises in the user story into a number of use cases, will be cascaded into a sequence diagram and then the number of transactions is calculated based on 4 COSMIC criteria [14].

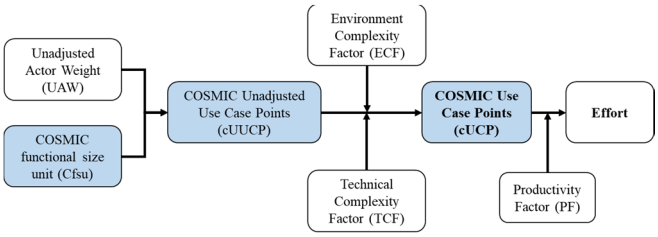

Fig. 2. The Substitution as A Proposed Model

Based on Fig. 2, see the blue-colored component. The explanations as below:

1) Determining Unadjusted Actor Weight

This first step is exactly the same as the original UCP method [10]. Each actor is given a weight based on easy, medium, or advance category. Considerations for weighting can be seen in Table I. To get the UAW value, the authors operate Eq. (1).

$$UAW = 1e+2m+3a \quad (1)$$

where,
e: actor with "easy" category
m: actor with "medium" category
a: actor with "advanced" category

TABLE I. ACTOR WEIGHT CLASSIFICATION

Category	Description	Weight
Easy (*e*)	Actor used API or command prompt	1
Medium (*m*)	Actor interact with protocol layer	2
Advance (*a*)	Actor accessed full of user interface	3

2) Replacing Use Case Weight with Early COSMIC

In the UCP method, we change the Unadjusted Use Case Weight (UUCW), then called Use Case Weight), to COSMIC functional size unit (Cfsu). UUCW is obtained by adding up the multiplication between 3 categories of use cases: simple, average, complex. Weight for simple use case categories is 5, average (10), and complex (15).

Meanwhile, Cfsu is calculated based on the sum of all transactions identified in the sequence diagram. This is in

line with research by Fehlmann et al [9] and Lavazza et al [23]. This stage also addresses the first research problem (*RP1*) (see Chapter II Statement of the Problem).

3) Transforming UAW and Cfsu into cUUCP

This stage is the novelty of this research. In particular, there have been no previous studies that have substituted UUCW to Cfsu (compare Fig. 1 and Fig. 2). The cUUCP calculation is obtained from the number of UAW and Cfsu. Therefore, this stage is also called COSMIC Unadjusted Use Case Points (cUUCP).

4) Calculating the Complexity Factor

In the UCP method, there are 2 types of complexity factors: technical and environmental. The Technical Complexity Factor (TCF) consists of 13 elements that have their respective weights and are multiplied by the 0-5 rating scale (see notes below Table II).

TABLE II. TECHNICAL FACTORS (*TFS*)

No	Description	Weight
1	Mandatory distributed system	2
2	Required highly return time	1
3	User interface efficiency	1
4	Internal processing difficulty	1
5	Reusable program/code	1
6	Ease of installation	0.5
7	Usability system	0.5
8	Support multi-platform	2
9	Ease of changing	1
10	Highly circumstantial	1
11	Security	1
12	Dependence of third-party code	1
13	End-user training	1

Scale: 0 non-exist, 1 exist, 2 necessary, 3 need improve, 4 significant impact, 5 strong influenced

To get the value of Technical Complexity Factor (TCF) we use Eq. (2). is as follows:

$$TCF = 0.01(\sum_{i-1}^{13} TFs_i) + 0.6 \quad (2)$$

where,

TFs_i : sum of the product of *TFs* weight and scale

Whereas the calculation of Environmental Complexity Factor (ECF) is done by adding up the multiplication of 8 aspects of environmental factors (*EFs*). The 8 aspects of *EFs* can be seen in Table III [10] [19]. Eq. (3) is used to calculate ECF values.

$$ECF = -(0.03\sum_{j=1}^{8} EFs_j) + 1.4 \quad (3)$$

where,

EFs_j : sum of the product of *EFs* weight and scale

TABLE III. ENVIRONMENTAL FACTORS (*EFS*)

No	Description	Weight
1	Sameness with another project	1.5
2	Team experience	0.5
3	Coding maturity	1
4	Analyst competence	0.5
5	Leadership encouragement	1
6	Consistency of requirements	2
7	Part-time staff	-1
8	Complexity of programming language	-1

Scale: 0 non-exist, 1 exist, 2 necessary, 3 need improve, 4 significant impact, 5 strong influenced

5) Counting the COSMIC Use Case Points (cUCP)

The cUCP calculation is obtained by multiplying all the components, namely cUUCP, TCF, and ECF. The right formula for cUCP can be seen in Eq. (4).

$$cUCP = (cUUCP)(TCF)(ECF) \quad (4)$$

6) Forcasting and Comparing the Software Effort

This last stage is used to predict the estimated value of software size. The productivity factor (PF) used is 8.2 man/hour [28]. After getting the predicted value, the writer then compares the deviation between the estimated effort of the cUCP method with the original UCP against the actual effort. At this stage, indirectly, has answered the second research problem (*RP2*) (see Chapter II Statement of the Problem).

B. Case Study

The case study used in this research is a web-based Hair Salon Online Booking application. Actual effort to develop the application is 216 man/hour and 2 workers-with-double-job during 12 workdays.

There is an interactive help desk that can be maximized by application visitors when they want to make a reservation. Starting from choosing the schedule, therapist by gender, and other special service requests. But if there is a special request, the operator is ready to help with solutions beyond the auto help desk.

Starting from the user story, as exemplified by Fehlmann [9], the author also describes the process/function flow and data movement into UML sequence diagrams. Fig. 3 illustrate the user scenarios for normal and alternative/error condition.

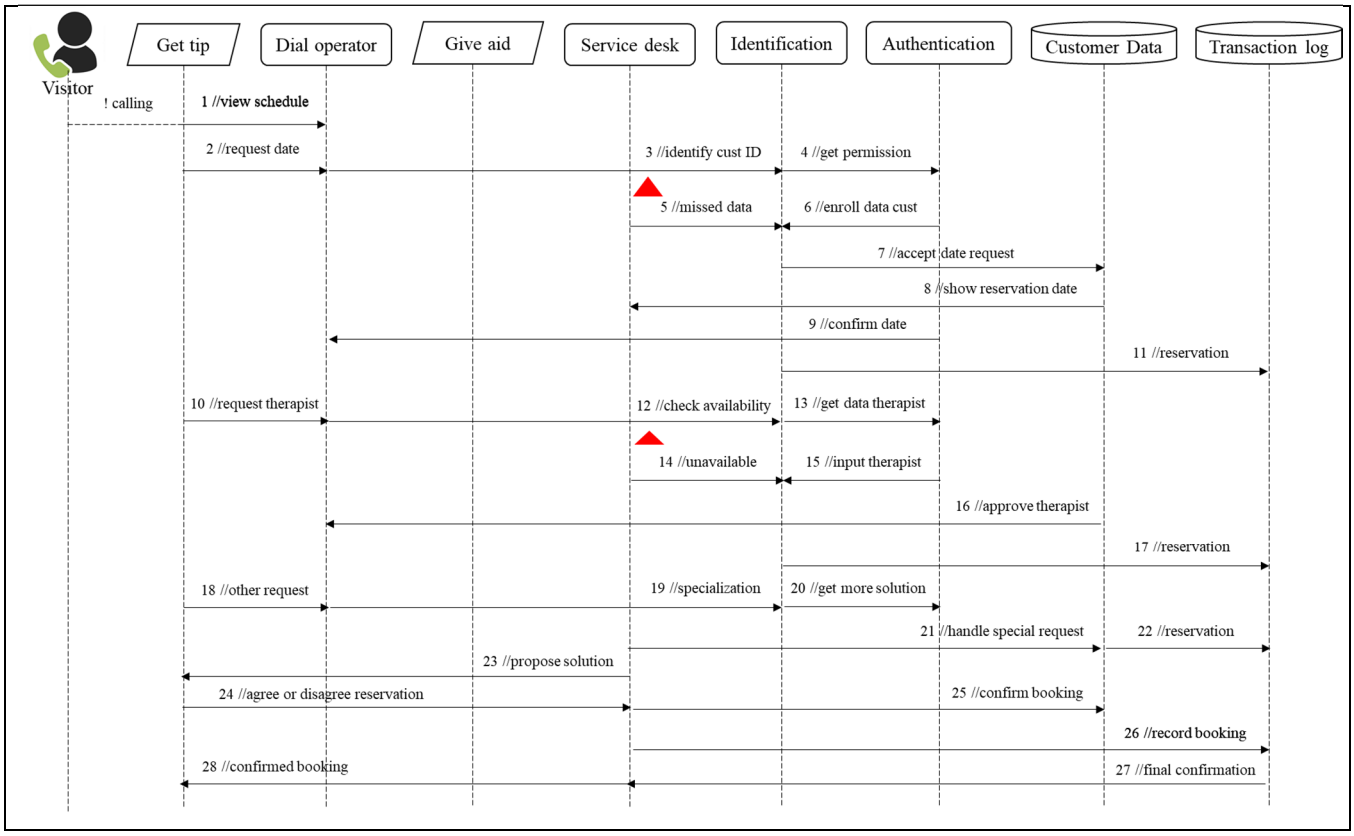

Fig. 3. Hair Salon Online Booking Application Presented in UML Sequence Diagram

V. RESULT

A. *Proposed Model Step-by-Step Result*

Refer to Chapter IV, there are 6 results to obtain each step. Every result would be discussed as detail as well.

1) *Determining Unadjusted Actor Weight*

In this case study, all actors fall into the "advance" category. Therefore, if the Unadjusted Actor Weight (UAW) value is calculated based on Eq. (1) a value obtained is 6.

2) *Replacing Use Case Weight into Early COSMIC*

Refer to Fig. 3, the process/function details can be summarized as in Table IV.

TABLE IV. TRANSACTION TYPE OF EACH PROCESS

COSMIC		Process/Function Name
No	***Type***	
1	Entry (E)	View schedule
2	Entry (E)	Request date
3	Entry (E)	Identify customer ID
4	Entry (E)	Get permission
5	Entry (E)	Enroll data customer
6	Entry (E)	Accept date request
7	Entry (E)	Request therapist
8	Entry (E)	Get data therapist
9	Entry (E)	Other request
10	Entry (E)	Specialization
11	Entry (E)	Get more solution
12	Entry (E)	Handle special request
13	Entry (E)	Confirm booking
14	Entry (E)	Agree or disagree reservation
15	Exit (X)	Confirm date
16	Exit (X)	Missed data
17	Exit (X)	Unavailable
18	Exit (X)	Input therapist
19	Exit (X)	Propose solution
20	Exit (X)	Confirmed booking
21	Read (R)	Show reservation date
22	Read (R)	Check availability
23	Read (R)	Approve therapist
24	Read (R)	Final confirmation
25	Write (W)	Reservation date
26	Write (W)	Reservation therapist
27	Write (W)	Reservation special request
28	Write (W)	Record booking

Based on Table IV, the total of the Early COSMIC is,

$$14E+6X+4R+4W= 28 \text{ Cfsu}$$

Because this study compares the original UCP with early COSMIC, the calculation of the Unadjusted Use Case Weight (UUCW) based on the use case narrative is presented in Table V.

TABLE V. CALCULATION OF UUCW

No	Description	Weight	Qty
1	Simple	5	1
2	Average	10	0
3	Complex	15	2
Total UUCW (Weight*Qty)			35

Table V means that UUCW from original UCP method have 35 values to the next counting step.

3) Transforming UAW and Cfsu into cUUCP

To get the cUUCP value, the authors add the UAW and Cfsu from the results of the previous stage.

$$cUUCP = 6+28 = 34$$

4) Calculating the Complexity Factor

TCF value obtained after entering Eq. (2) is 0.88. While the ECF value is obtained from Eq. (3) which is 0.95.

5) Counting the COSMIC Use Case Points (cUCP)

Based on Eq. (4), to get cUCP is to multiply all components of cUUCP, TCF, and ECF. The result of software size by cUCP method is 28.42. In the other hand, the original UCP method is 34.28 (see Table VI).

TABLE VI. COMPARATION OF SOFTWARE SIZE

Application Name	UUCW	UUCP	UCP
Hair Salon Online Booking	35	41	34.28
	Cfsu	**cUUCP**	**cUCP**
	28	34	28.42

6) Forcasting and Comparing the Software Effort

To convert software sizes using cUCP into units of effort (man/hour) must refer to certain productivity factors (PF). The PF constant used is referring to the effort rate by Subriadi et al is 8.2 [28].

TABLE VII. SOFTWARE EFFORT ESTIMATION

Application Name	Software Size by UCP	Software Size by cUCP
Hair Salon Online Booking	34.28	28.42
Effort (man/hour)	281.06	233.08
Deviation to Actual Effort (%)	23.15	7.33

Based on Table VII, the authors compare the results of the original UCP and early COSMIC UCP (cUCP) estimation values against the real effort. As explained in Section II-B, the actual effort to develop the application is 216 man/hour.

B. Answering the Statement of The Problem

The answer of each problem of this study presented on Table VIII.

TABLE VIII. RESEARCH PROBLEM AND ITS ANSWER

No	Details
RP1	Can the use case weight in the original UCP method be replaced by early COSMIC model?
Answer	Yes, it can. Look at the second step in Chapter III Method. Early COSMIC, clearly, can replace the use case weight.
RP2	What percentage of accuracy is generated at the estimated effort based on the early COSMIC method with UCP?
Answer	The percentage of accuracy of cUCP reach 92.67%

Table VIII represented that the replacement of use case weight with early COSMIC more accurate than pure UCP (76.85%) against actual effort.

VI. CONCLUSION

Following the development of functional size measurement, we should not close our eyes by ignoring the constantly updated version of COSMIC. In this study, the authors show a contribution to changing the use case weight in the UCP method to a COSMIC functional size unit (Cfsu). This study proves that the substitution of Cfsu into the modified cUCP method has a better accuracy of estimation value compared to the original UCP against actual effort to 92.67 percent.

VII. FUTURE WORKS

This study talks about the estimated value of effort using the early COSMIC approach in a simple case study. In future research, we intend to test the estimated value of effort on several larger and more complex case objects. Not only focus on changing the use case weight (because many researchers have doubted before), but also modifying the technical and environmental complexity factors. Some of the authors' future studies will bring up inventions of software size estimation methods that are relevant to current software development models.

ACKNOWLEDGMENT

Thanks to the Information Systems Management Laboratory of Institut Teknologi Sepuluh Nopember (ITS) Surabaya and Research Institution (LPPM) of Universitas Internasional Semen Indonesia (UISI).

REFERENCES

[1] A. J. Albrecht, "Measuring application development productivity," in *IBO Conference on Application Development*, 1979, pp. 83–92, doi: 10.1109/WETSoM.2012.622986.

[2] IFPUG, "ISO/IEC 20926:2009 Software and systems engineering — Software measurement — IFPUG functional size measurement method 2009." ISO, Accessed: Apr. 17, 2020. [Online]. Available: https://www.iso.org/standard/51717.html.

[3] ISO, "ISO/IEC 20968:2002 Software engineering — Mk II Function Point Analysis — Counting Practices Manual." ISO, Accessed: Apr. 17, 2020. [Online]. Available: https://www.iso.org/standard/35603.html.

[4] B. Boehm *et al.*, "COCOMO II Model Definition Manual," *Univ. South. Calif.*, vol. 4, no. 1, pp. 6–6, 2000, doi: 10.1525/nr.2000.4.1.6.

[5] K. Langsari and R. Sarno, "Optimizing Effort and Time Parameters of COCOMO II Estimation using Fuzzy Multi-Objective PSO," no. September, pp. 19–21, 2017.

[6] K. Langsari, R. Sarno, and Sholiq, "Optimizing Effort Parameter of COCOMO II Using Particle Swarm Optimization Method," vol. 16, no. 5, pp. 2208–2216, 2018, doi: 10.12928/TELKOMNIKA.v16i5.9703.

[7] R. R. Putri, R. Sarno, D. Siahaan, A. S. Ahmadiyah, and S. Rochimah, "Accuracy improvement of the estimations effort in Constructive Cost Model II based on logic model of fuzzy," *Adv. Sci. Lett.*, vol. 23, no. 3, pp. 2478–2480, 2017.

[8] M. Baiquni, R. Sarno, and others, "Improving the accuracy of COCOMO II using fuzzy logic and local calibration method," in *2017 3rd International Conference on Science in Information Technology (ICSITech)*, 2017, pp. 284–289.

[9] T. M. Fehlmann and E. Kranich, "COSMIC Functional Sizing based on UML Sequence Diagrams," p. 16, 2011.

[10] G. Karner, "Resource Estimation for Objectory Projects," *Object. Syst. SF AB*, 1993.

[11] M. Aguiar, "Function Points or Use Case Points?," p. 2, 2009.

[12] C. Jones, "Function points as a universal software metric," *ACM SIGSOFT Softw. Eng. Notes*, vol. 38, no. 4, pp. 1–27, Jul. 2013, doi: 10.1145/2492248.2492268.

[13] ISO, *ISO/IEC 19761:2011*, 4.02. 2019.

[14] COSMIC, "ISO/IEC 19761:2011." ISO, Accessed: Apr. 20, 2020. [Online]. Available: https://cosmic-sizing.org/2020/03/announcement-of-version-5-0-of-the-cosmic-measurement-manual-march-31-2020/.

[15] M. Ochodek and J. Nawrocki, "Automatic Transactions Identification in Use Cases," in *Balancing Agility and Formalism in Software Engineering*, vol. 5082, B. Meyer, J. R. Nawrocki, and B. Walter, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 55–68.

[16] M. Ochodek, J. Nawrocki, and K. Kwarciak, "Simplifying effort estimation based on Use Case Points," *Inf. Softw. Technol.*, vol. 53, no. 3, pp. 200–213, 2011, doi: 10.1016/j.infsof.2010.10.005.

[17] M. Ochodek, "Functional size approximation based on use-case names," *Inf. Softw. Technol.*, vol. 80, pp. 73–88, Dec. 2016, doi: 10.1016/j.infsof.2016.08.007.

[18] R. S. Dewi, G. F. Prassida, Sholiq, and A. P. Subriadi, "UCPabc as an integration model for software cost estimation," in *Proceeding - 2016 2nd International Conference on Science in Information Technology, ICSITech 2016: Information Science for Green Society and Environment*, 2017, doi: 10.1109/ICSITech.2016.7852631.

[19] R. S. Dewi, A. P. Subriadi, and Others, "A Comparative Study of Software Development Size Estimation Method: UCPabc vs Function Points," *Procedia Comput. Sci.*, vol. 124, pp. 470–477, 2017.

[20] Sholiq, A. P. Subriadi, F. A. Muqtadiroh, and R. S. Dewi, "A model of owner estimate cost for software development project in Indonesia," *J. Softw. Evol. Process*, 2019, doi: 10.1002/smr.2175.

[21] N. J. Nunes, "iUCP - Estimating interaction design projects with enhanced use case Points," *Lect. Notes Comput. Sci. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinforma.*, vol. 5963 LNCS, pp. 131–145, 2010, doi: 10.1007/978-3-642-11797-8_11.

[22] P. Jha and R. K. Malu, "Cost Estimation Using Parameterized For Use Case Point Model (P-UCP)," *Int. J. Eng. Res. Appl.*, vol. 3, no. 5, pp. 1280–1283, 2013.

[23] L. Lavazza and V. Del Bianco, "A Case Study in COSMIC Functional Size Measurement: The Rice Cooker Revisited," in *Software Process and Product Measurement*, vol. 5891, A. Abran, R. Braungarten, R. R. Dumke, J. J. Cuadrado-Gallego, and J. Brunekreef, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 101–121.

[24] V. del Bianco, L. Lavazza, G. Liu, S. Morasca, and A. Z. Abualkishik, "Model-based early and rapid estimation of COSMIC functional size – An experimental evaluation," *Inf. Softw. Technol.*, vol. 56, no. 10, pp. 1253–1267, Oct. 2014, doi: 10.1016/j.infsof.2014.04.009.

[25] L. Lavazza and S. Morasca, "An Empirical Evaluation of Two COSMIC Early Estimation Methods," in *2016 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA)*, Berlin, Germany, Oct. 2016, pp. 65–74, doi: 10.1109/IWSM-Mensura.2016.020.

[26] N. A. S. Abdullah, N. I. A. Rusli, and M. F. Ibrahim, "A case study in COSMIC functional size measurement: Angry Bird Mobile Application," in *2013 IEEE Conference on Open Systems (ICOS)*, Kuching, Malaysia, Dec. 2013, pp. 139–144, doi: 10.1109/ICOS.2013.6735063.

[27] Ç. Gencel, L. Buglione, O. Demirors, and P. Efe, "A Case Study on the Evaluation of COSMIC-FFP and Use Case Points," p. 21.

[28] A. P. Subriadi, Sholiq, and P. A. Ningrum, "Critical Review of The Effort Rate Value in Use Case Point Method for Estimating Software Development Effort," *J. Theor. Appl. Inf. Technol.*, vol. 59, no. 3, pp. 735–744, 2014.