Plagiarism Checker X Originality Report

Plagiarism Quantity: 24% Duplicate

Date	Saturday, September 26, 2020
Words	1263 Plagiarized Words / Total 5198 Words
Sources	More than 87 Sources Identified.
Remarks	Medium Plagiarism Detected - Your Document needs Selective Improvement.

Procedia Computer Science 124 (2017) 470�477 1877-0509 � 2018 The Authors. Published by Elsevier B.V. Peer-review under responsibility of the scientific committee of the 4th Information Systems International Conference 2017 10.1016/j.procs.2017.12.179 10.1016/j.procs.2017.12.179 1877-0509 Available online at www.sciencedirect.com Procedia Computer Science 00 (2018) 000 �000 www.elsevier.com/locate/procedia 1877-0509 � 2018 The Authors. Published by Elsevier B.V. Peer-review under responsibility of the scientific committee of the 4th Information Systems International Conference 2017.

4th Information Systems International Conference 2017, ISICO 2017, 6-8 November 2017, Bali, Indonesia A Comparative Study of Software Development Size Estimation Method: UCPabc vs Function Points Sholiqa,*, Renny Sari Dewib, Apol Pribadi Subriadia aInformation Systems Department, Institut Teknologi Sepuluh Nopember, Kampus ITS Sukolilo -Surabaya 60111, Indo nesia bInformation Systems Department, Universitas Internasional Semen Indonesia, Jalan Raya Veteran, Gresik 61122, Indonesia Abstract One of the stages in planning software development projects is to estimate the effort and cost.

The good news is, there are already studies related to the estimated cost of software development projects whose results are close to the real cost. In this stud y, we compared two cost estimation methods which have quite small deviations, such Use Case Points - Activity Based Costing models (then namely UCPabc) and Function Points (FP). Some aspects that were compared are process and parameters, c omplexity factors, and deviation. The results of this research are, first, the difference of process algorithm and parameters.

Secondly, there were differences in complexity factors, 21 factors on UCPabc model and 14 factors on FP

Sources found:

Click on the highlighted sentence to see sources.

Internet Pages

0%	https://www.sciencedirect.com/science/ar
0%	https://www.researchgate.net/publication
0%	https://www.sciencedirect.com/science/ar
0%	https://www.sciencedirect.com/science/ar
1%	https://www.sciencedirect.com/journal/pr
0%	Empty
0%	https://www.bing.com/aclick?ld=e8dZSYwgE
0%	https://pmstudycircle.com/2012/05/planne
1%	https://www.researchgate.net/publication
0%	http://www.worldcomp-proceedings.com/pro
2%	https://www.researchgate.net/publication
1%	http://www.download-paper.com/wp-content
0%	https://www.bing.com/aclick?ld=e8Un9tnfb
0%	https://www.bing.com/aclick?ld=e8VI-tDgP
0%	https://www.sciencedirect.com/science/ar
0%	https://www.sciencedirect.com/science/ar
0%	https://www.researchgate.net/profile/Nor
0%	https://www.researchgate.net/publication
3%	https://www.sciencedirect.com/science/ar
1%	https://www.researchgate.net/publication

method. So, the deviation between two methods of effort estimation toward actual effort was 11.9 percent using UCPabc and 27.8 percent using FP. Therefore, the U CPabc method is the closest method of effort estimation toward actual effort. � 2018 The Authors. Published by Elsevier B.V. Peer-review under responsibility of the scientific committee of the 4th Information Systems International Conference 2017.

Keywords: Software Cost Estimation; Cost Estimation; Function Points; Use Case Points, Activitity Based Costing 1. Introduction Currently, the activity stage of project planning on software development is increasingly diverse. One way is to estimate the cost of software development projects. Along with the development of science, currently many studies have discussed about the accuracy of some cost estimation methods.

Therefore, we compared the results of previous research which is now booming. * Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000 . E-mail address: sholiq@is.its.ac.id Sholiq et al. / Procedia Computer Science 124 (2017) 470�477 471 Available online at www.sciencedirect.com Procedia Computer Science 00 (2018) 000 �000 www.elsevier.com/locate/procedia 1877-0509 � 2018 The Authors. Published by Elsevier B.V. Peer-review under responsibility of the scientific committee of the 4th Information Systems International Conference 2017.

4th Information Systems International Conference 2017, ISICO 2017, 6-8 November 2017, Bali, Indonesia A Comparative Study of Software Development Size Estimation Method: UCPabc vs Function Points Sholiqa,*, Renny Sari Dewib, Apol Pribadi Subriadia aInformation Systems Department, Institut Teknologi Sepuluh Nopember, Kampus ITS Sukolilo -Surabaya 60111, Indo nesia bInformation Systems Department, Universitas Internasional Semen Indonesia, Jalan Raya Veteran, Gresik 61122, Indonesia Abstract One of the stages in planning software development projects is to estimate the effort and cost.

The good news is, there are already studies related to the estimated cost of software development projects whose results are close to the real cost. In this stud y, we compared two cost estimation methods which have quite small deviations, such Use Case Points - Activity Based Costing models (then namely UCPabc) and Function Points (FP). Some aspects that were compared are process and parameters, c omplexity factors, and deviation. The results of this research are, first, the difference of process algorithm and parameters.

Secondly, there were differences in complexity factors, 21 factors on UCPabc model and 14 factors on FP method. So, the deviation between two methods of effort estimation toward actual effort was 11.9 percent using UCPabc and 27.8 percent using FP. Therefore, the U CPabc method is the closest method of effort estimation toward actual effort. � 2018 The Authors. Published by Elsevier B.V. Peer-review under responsibility of the scientific committee of the 4th Information Systems International Conference 2017.

0%	http://www.umsl.edu/~sauterv/analysis/fu
0%	https://www.investopedia.com/terms/d/dcf
0%	https://courses.cs.washington.edu/course
0%	https://www.sciencedirect.com/science/ar
0%	https://www.cmu.edu/cee/projects/PMbook/
0%	https://pdfs.semanticscholar.org/e0e6/98
0%	https://www.bing.com/aclick?ld=e8go1l0oN
0%	https://www.sciencedirect.com/science/ar
0%	https://www.geeksforgeeks.org/software-e
0%	https://www.ncbi.nlm.nih.gov/pmc/article
0%	https://www.researchgate.net/publication
0%	https://mafiadoc.com/software-developmen
0%	https://cordis.europa.eu/project/id/2660
0%	http://www.ilo.org/wcmsp5/groups/public/
0%	https://www.researchgate.net/publication
0%	https://www.sciencedirect.com/science/ar
0%	https://www.mdpi.com/1996-1073/12/10/193
0%	https://www.sciencedirect.com/topics/eng
0%	https://www.sciencedirect.com/science/ar
0%	https://blackfalconsoftware.wordpress.co
0%	https://www.bing.com/aclick?ld=e8IjEaRMq
0%	https://www.accountingformanagement.org/
0%	https://www.chems.msu.edu/resources/tuto
0%	http://export.arxiv.org/list/cs/new
0%	https://www.researchgate.net/publication
0%	https://www.tutorialspoint.com/estimatio
0%	https://www.sciencedirect.com/science/ar
1%	https://www.sciencedirect.com/science/ar
0%	https://www.bing.com/aclick?ld=e8gSth_cy
0%	http://digilib.uinsby.ac.id/view/item_ty

Keywords: Software Cost Estimation; Cost Estimation; Function Points; Use Case Points, Activitity Based Costing 1. Introduction Currently, the activity stage of project planning on software development is increasingly diverse. One way is to estimate the cost of software development projects. Along with the development of science, currently many studies have discussed about the accuracy of some cost estimation methods.

Therefore, we compared the results of previous research which is now booming. * Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000 . E-mail address: sholiq@is.its.ac.id 2 Sholiq et al. / Procedia Computer Science 00 (2018) 000 �000 The Use Case Point s method (UCP) since its introduction in 1993 by Karner [1], has been tested by several researchers. The resulting deviation between cost estimates uses UCP and the actual cost of 'only' 6.89 percent in small and medium softwar e [2].

Meanwhile, according to Dewi [3], his research resulted in a smaller deviation of 2.16 percent when UCP method was integrated with Activity Based Costing (ABC) technique (then known as UCPa bc model) to estimate the cost of 5 software development projects. In the Function Points (FP) method, Albrecht introduced this method first in IBM company case study [4]. According to Aguiar [5], the International Function Points User Group (IFPUG) has officially declared that FP methods are suitable for any softwa re genre.

Interestingly, in research which we have done that the result of cost estimation using FP method gives a small deviation of 3. 26 percent [6]. This means, FP methods are almost close to the actual cost in software development projects. From the above explanation, there are some previous studies on some comparisons of software development project cost estimates by Usharani et al [7]. But from the results obtained, has not yet concluded which method closes to the actual cost of the project.

Therefore, this study aims to compare two methods of estimating the cost of which software development projects are closest to the actual cost. In the future, business software developers are able to independently decide which method is appropriate and close to the actual cost of a software development project. 2. Related research Several previous studies that examine cost estimations in software development projects have been summarized as presented in Table 1. Table 1. Related research No Author, Year Result Research connectivity Research gap 1 Albrecht, 1983 [4] Predicted effort using Function Points (FP) based on software function and number of lines of code in IBM company. As the main literature about feasibility of the FP method to estimate the software development project effort.

FP methods have been tested but have never been compared with other methods. 2 Aguiar, 2009 [5] FP which has been recommended by the International Function Points User Group (IFPUG) as a successful method implemented in government and industry was compared to Use Case Points based on surveys conducted. As the main reason why we chose FP compared to UCP. IFPUG only compares FP with the

0%	https://www.icanrentacar.com/goverment-t
0%	https://link.springer.com/article/10.100
0%	https://www.bing.com/aclick?ld=e8rLEv9p7
0%	https://thedigitalprojectmanager.com/pro
0%	https://www.nature.com/articles/ng1543
0%	http://sourceforge.net/p/functionpoints/
0%	https://en.wikipedia.org/wiki/Activity-b
0%	https://www.digitaldoughnut.com/articles
0%	https://www.chegg.com/homework-help/ques
0%	https://www.chegg.com/homework-help/ques
0%	https://www.bing.com/aclick?ld=e8i2obymQ
0%	https://genomebiology.biomedcentral.com/
0%	https://www.bing.com/aclick?ld=e8YvEEi8p
0%	http://www.jatit.org/volumes/Vol59No3/24
0%	http://home.iitk.ac.in/~shalab/econometr
0%	https://satheespractice.blogspot.com/201
0%	http://saylordotorg.github.io/text_intro
0%	https://www.educba.com/creating-database
0%	https://www.allaboutrequirements.com/201
0%	https://www.researchgate.net/publication
0%	https://www.researchgate.net/publication
0%	https://lbeifits.files.wordpress.com/202
0%	https://www.slideshare.net/nagaraja_gund
0%	https://www.ncbi.nlm.nih.gov/pmc/article
1%	https://www.bing.com/aclick?ld=e8l_J1OuX
0%	https://www.sciencedirect.com/science/ar
0%	https://www.scribd.com/document/40212998
0%	https://www.bing.com/aclick?ld=e8SDyU1sn
1%	https://uisi.ac.id/assets/upload/media/8
0%	https://www.researchgate.net/publication

original UCP method (not yet integrated with other methods).

3 Arnuphaptrairong, 2013 [8] Implementation of FP method with Cocomo method on 15 software development project that produces parameter model based on language programming. FP method has been collaborated with other methods to find optimal effort estimates The modification of FP method with Cocomo resulted in an average deviation of 1,624.31%. 4 Dewi et al, 2014 [3] The Use Case Points method that was integrated with Activity Based Costing (or UCPabc model) was able to estimate the cost of developing 5 public service applications The similarity of case studies taken by the author, namely the application of public services.

- 5 Dewi et al, 2016 [9] The level of accuracy of the estimated cost model using UCP abc compared to actual cost has a low deviation of 2.16 percent . The case studies used have similarities that there are 4 applications of public services. The deviation between the estimated cost of the UCPabc model and the actual cost has never been compared with other estimation methods. 6 Usharani et al, 2016 [7] Critical review on algorithms of some estimation methods such as Analogy Based Estimation and Artificial Neural Network (ANN).

Comparison can be done to find out which one best suits the existing problem. The comparison of the estimation method focuses only on the algorithmic level of detail. 472 Sholiq et al. / Procedia Computer Science 124 (2017) 470�477 Sholiq et al. / Procedia Computer Science 00 (2018) 000 �000 3 Fig. 1 Research method. The UCPabc model consists of two core methods, namely software measurement methods and activity -based costs that are performed on software development projects. Because this UCPabc model uses an activity -based costing technique, the similarity of the cost estimation results to the actual cost of the project becomes higher [3].

On the other hand, the FP estimation method that has since been introduced has never been compared with any method, IFPUG claims that this method is most suitable to be applied in all types of software development projects [6]. The research stages used to determine which cost estimation method is closest to real cost is by identifying each of the test results of the method. Based on Fig. 1, the novelty of this study is a comparison of two cost estimation methods of software development projects, namely: UCPabc and FP model that has never been done by previous researchers.

The comparison of these two methods refers to the case study on the research of Dewi [3] [9] [6], ie 4 application licensing of companies (Industrial Registration, Principle Approval, Industrial Allowance, and Certificate of Company License). This means, we will compare, which method is closest to real cost if the software development project is a public service application. 3.1. Understanding process and measurement parameter In the UCPabc model, there are many stages to go through.

0% https://www.bing.com/aclick?ld=e8LXegK_O
0% https://www.bing.com/aclick?ld=e87YYzkpw
0% https://www.bing.com/aclick?ld=e8_8H0dQi
0% https://teknosi.fti.unand.ac.id/index.ph
0% https://www.ijcaonline.org/archives/volu
0% https://mt735blog.wordpress.com/9-refere
0% https://www.bing.com/aclick?ld=e8AyWyaNW

The process starts from the use case narrative and the weighting of the actor, then a complex calculation is done so that it can generate effort (man-hour). While the FP method, the parameters to be measured consist of 5 things, including External Input (Exi), External Output (Exo), External Inquiry (Exiq), Internal Logic File (Ilof), and External Logic File (Elof). 3.2. Identifying complexity factors The UCPabc model determines the complexity factor based on the Use Case Points method, which consists of technical factor (TF) and environment factor (EF). While the FP method, the complexity factor is determined by the value of complexity adjustment factor. 3.3.

Comparing the deviation Deviation is the final result of the overall calculation of both cost estimation methods, either UCPabc or FP model. The deviation value is evidenced by comparing the estimated effort to actual effort for each method, i e UCPabc and FP. Sholiq et al. / Procedia Computer Science 124 (2017) 470�477 473 Sholiq et al. / Procedia Computer Science 00 (2018) 000 �000 3 Fig. 1 Research method. The UCPabc model consists of two core methods, namely software measurement methods and activity-based costs that are performed on software development projects.

Because this UCPabc model uses an activity -based costing technique, the similarity of the cost estimation results to the actual cost of the project becomes higher [3]. On the other hand, the FP estimation method that has since been introduced has never been compared with any method, IFPUG claims that this method is most suitable to be applied in all types of software development projects [6].

The research stages used to determine which cost estimation method is closest to real cost is by identifying each of the test results of the method. Based on Fig. 1, the novelty of this study is a comparison of two cost estimation methods of software development projects, namely: UCPabc and FP model that has never been done by previous researchers. The comparison of these two methods refers to the case study on the research of Dewi [3] [9] [6], ie 4 application licensing of companies (Industrial Registration, Principle Approval, Industrial Allowance, and Certificate of Company License).

This means, we will compare, which method is closest to real cost if the software development project is a public service application. 3.1. Understanding process and measurement parameter In the UCPabc model, there are many stages to go through. The process starts from the use case narrative and the weighting of the actor, then a complex calculation is done so that it can generate effort (man-hour). While the FP method, the parameters to be measured consist of 5 things, including External Input (Exi), External Output (Exo), External Inquiry (Exiq), Internal Logic File (Ilof), and External Logic File (Elof). 3.2.

Identifying complexity factors The UCPabc model determines the complexity factor based on the Use Case Points method, which consists of technical factor (TF) and environment factor (EF). While the FP method, the complexity factor is determined by the value of complexity adjustment factor. 3.3. Comparing the deviation Deviation is the final result of the overall calculation of both cost estimation methods, either UCPabc or FP model. The deviation value is evidenced by comparing the estimated effort to actual effort for each method, i e UCPabc and FP. 4 Sholiq et al.

/ Procedia Computer Science 00 (2018) 000 �000 4. Result Based on previous research [9], Table 2 shows the cost budget for software development projects of 4 projects of public service applications. Table 2. Budget for project software development . Project ID Application Name Description Budget (IDR)* 1 Industrial Registration This application is intended for individuals / business entities in the field of small - scale industry 35,883,000 2 Principle Approval This application is intended for individuals / business entities of the middle category industry as a condition of filing Industrian Allowance [12] 37,908,000 3 Industrial Allowance This application is intended for individuals / business entities of the middle category industry as stated in Article 46 Perda Kota Surabaya No.1

Year 2010 [12] 38,134,800 4 Certificate of Company License This application is for all trading business entities ranging from trading business such Usaha Dagang (UD), CV, Ltd., Cooperatives and other business entities 74,115,000 Total 186,040,800 *) The budget is netto, exclude taxes for company: 1.5% PPh article 22, 7.5% PPh article 23, and 10% PPn 4.1. Understanding mindset of UCPabc model dan FP method The estimated cost of the software development project using the UCPabc model is certainly different from the FP method. The several fundamental differences that have been identified in the previous UCPabc and FP model studies [3] [6] are shown in Table 3. Table 3.

Aspect comparison of two method for estimating software effort . No Aspect UCPabc Function points 1 Process Process flow to estimate cost by calculating UCP value, then get total effort value by multiplying UCP with ER or PF. After getting total effort then proceed to find the value of Product Relative Weight from the percentage of UCP calculation results to the total effort of software development project (see Fig 2a).

Process flow to calculate cost estimation using FP method is calculate input, output, inquiry, and logical file then combined with payrate of each determined activity (s ee Fig2b). 2 Parameter The measurement parameters are divided into 2: actors and use case scenarios. Both of these parameters affect activity -based costs in software development.. 5 parameters: external input, external output, external inquiry, internal logical file, and external logical file.

3 Complexity factor Technical and environment complexity factor Only technical complexity factor 4 Advantage In the UCPabc model, Activity Based Costing (ABC) techniques have considered overhead expenses when company uses resource sharing. Therefore, the distribution of the cost is more accurate. The advantage is that the estimation of effort can be done in the absence of system analysis results (use case scenario), but simply consider the input, output, and lo gical files. But on the other hand, the distribution effort against the activity is less accurate.

The cost estimate is based on labor payrate and ignores overhead costs. In general, the difference of the process flow to obtain the estimated effort which is then converted to cost estimates on the UCPabc model and FP method is shown in Fig. 2 (a) and (b). The cost estimation of 4 software development projects for this public service application is influenced by several aspects which are described in Table 4.

Therefore, we examine these aspects to determine the impact of differences between the UCPabc model and the FP method in the form of deviation difference resulting from 4 software development project. 474 Sholiq et al. / Procedia Computer Science 124 (2017) 470�477 Sholiq et al. / Procedia Computer Science 00 (2018) 000 �000 5 Fig. 2 (a) Cost estimation by UCPabc model [9], and (b) Cost estimation by function points method [6]. (a) (b) 4.2. Identifying complexity factors At the stage of identifying complexity factors in the UCPabc and FP models, we refer to the results of previous studies [9] [6].

The UCPabc model, its complexity factor consists of 2 types, namely the technical factor and the team environment. On the technical factor consists of 14 aspects and 7 aspects on the team's environmental factors. While the FP method, there are 14 factors of complexity that must be calculated and given an assessment in the interval score of 0 (no influence) to 5 (strong impact / essentials). The factor of complexity which affects the estimation method of cost in both UCPabc and FP is presented in Table 4. Table 4.

Complexity factor: UCPabc vs FP. No UCPabc (category of complexity) Function points 1 Distributed System Required (TF) Level of backup and recover reliability 2 Response Time is Important (TF) Level of data communications 3 End User Efficiency (TF) Level of distributed data processing 4 Internal Processing Required (TF) Level of performance needs 5 Reusable Code Must Be A Focus (TF) Level of environment configuration 6 Installation Easy (TF) Level of transaction rate 7 Usability (TF) Level of end-user efficiency 8 Cross-Platform Support (TF) Level of master file update 9 Easy To Change (TF) Level of online real-time update 10 Highly Concurrent (TF) Level of reusability 11 Custom Security (TF) Level of installation ease 12 Dependence On Third-Party Code (TF) Level of operational ease 13 User Training (TF) Level of customer organisation variation Sholiq et al. / Procedia Computer Science 124 (2017) 470�477 475 Sholiq et al. / Procedia Computer Science 00 (2018) 000 �000 5 Fig.

2 (a) Cost estimation by UCPabc model [9], and (b) Cost estimation by function points method [6]. (a) (b) 4.2. Identifying complexity factors At the stage of identifying complexity factors in the UCPabc and FP models, we refer to the results of previous studies [9] [6]. The UCPabc model, its complexity factor consists of 2 types, namely the technical factor and the team environment. On the technical factor consists of 14 aspects and 7 aspects on the team's environmental factors.

While the FP method, there are 14 factors of complexity that must be calculated and given an assessment in the interval score of 0 (no influence) to 5 (strong impact / essentials). The factor of complexity which affects the estimation method of cost in both UCPabc and FP is presented in Table 4. Table 4. Complexity factor:

UCPabc vs FP. No UCPabc (category of complexity) Function points 1 Distributed System Required (TF) Level of backup and recover reliability 2 Response Time is Important (TF) Level of data communications 3 End User Efficiency (TF) Level of distributed data processing 4 Internal Processing Required (TF) Level of performance needs 5 Reusable Code Must Be A Focus (TF) Level of environment configuration 6 Installation Easy (TF) Level of transaction rate 7 Usability (TF) Level of end-user efficiency 8 Cross-Platform Support (TF) Level of master file update 9 Easy To Change (TF) Level of online real-time update 10 Highly Concurrent (TF) Level of reusability 11 Custom Security (TF) Level of installation ease 12 Dependence On Third-Party Code (TF) Level of operational ease 13 User Training (TF) Level of customer organisation variation 6 Sholiq et al.

/ Procedia Computer Science 00 (2018) 000 �000 No UCPabc (category of complexity) Function points 14 Familiarity with the Project (EF) Level of change possibility 15 Application Experience (EF) 16 Object-Oriented Programming Experience (EF) 17 Lead Analyst Capability (EF) 18 Motivation (EF) 19 Stable Requirements (EF) 20 Part Time Staff (EF) 21 Difficult Programming Language (EF) 4.3. Comparing deviation between UCPabc and FP This deviation is the final result of the overall calculation of both cost estimation methods, either UCPabc or FP.

The deviation value of these two methods is evidenced by the proportion of effort distribution and the estimated cost of actual effort (see Table 6 and 7). Based on Table 5, the final value of the calculation using the model of UCPabc integration is 1,917.48 [9], while the calculation uses FP of 1,572.77 [6]. The values of UCPabc and FP are then converted to units of man -hour effort as shown in Table 6. Table 5. Result of estimation: UCPabc vs FP. Project ID Total UCPabc Total FP 1 460.48 228.52 2 472.99 348.92 3 482.45 301.07 4 501.56 694.26 1,917.48 1,572.77

Table 6 shows that the effort required to develop 4 software applications of public services using the UCPabc model. In table 6 there are columns of Unadjusted Use Case Weight (UUCW), Unadjusted Actor Weight (UAW), Technical Complexity Factor (TCF) and Environmental Complexity Factor (ECF) data. UCPabc is obtained with the formula UCPabc = (UUCW + UAW) * TCF * ECF. For example for ProjectID = 1, then UCPabc = (530 + 22) * 0.97 * 0.86 = 460.48. Similarly for ProjectID = 2, 3, and 4, each value is UCPabc = 472.99; 482.45; and 501.56.

Furthermore, using the constant of Productivity Factors (PF) = 8.2 man -hours [10], effort is obtained by multiplying UCPabc with PF, so effort = UCPabc * PF = 460.48 * 8.2 = 3,775.92 man -hours for Projec tID = 1. Whereas ProjectID = 2, 3, and 4 are respectively 3,878.53; 3,956.07; and 4,112.77. Thus, the total effort for the four projects is 15,723.29 man-hours (see Table 6). Table 6. Effort estimation using use case points - activity based costing (UCPabc). Project ID UUCW UAW TCF ECF UCPabc Effort 1 530 22 0.97 0.86 460.48 3,775.92 2 545 22 0.97 0.86 472.99 3,878.53 3 530 22 0.92 0.95 482.45 3,956.07 4 475 24 1.075 0.935 501.56 4,112.77 1,917.48 15,723.29 Source: Dewi et al, 2016 [9] 476 Sholiq et al. / Procedia Computer Science 124 (2017) 470�477 Sholiq et al.

/ Procedia Computer Science 00 (2018) 000 �000 7 Table 7 shows the estimated effort using the FP method. In the table are given data 4 projects with 5 input parameters, namely: External Input (Exi), External Output (Exo), External Inquiry (Exiq), Internal Logic File (Ilof), and External Logic File (Elof) . Table 7 also contains the Modifi ed Complexity Adjustment Factor (MCAF) [6]. FP with the formula FP = (Exi + Exo + Exiq + IIlof + Elof) * MCAF. For ProjectID = 1, then FP = (39 + 57 + 25 + 42 + 34) * 1.16 = 228.52.

Then fffort is obtained by multiplying FP with PF constant, so that effort for ProjectID=1 equals 1,873.86 man -hours. Likewise for ProjectID= 2, 3, and 4 respectively obtained effort= 2,861.14; 2,468.77; and 5,692.93 man-hours (see Table 7). The total effort for all four projects is 12,896.71 man -hours. Table 7. Effort estimation using function points (FP). Project ID Exi Exo Exiq IloF ELof MCAF FP Effort 1 39 57 25 42 34 1.16 228.52 1,873.86 2 34 48 28 82 94 1.22 348.92 2,861.14 3 34 48 28 68 75 1.19 301.07 2,468.77

4 61 69 81 178 133 1.33 694.26 5,692.93 Total 1,572.77 12,896.71 Source: Dewi et al, 2017 [6] As is the novelty of this study, the comparison of both estimation methods will be compared with actual effort (see Table 8). Based on the results obtained from the estimation effort on each method, when compared with actual effort can be seen in Table 8. Comparison between effort estimation with actual effort is expressed by using deviation defined that: ????????????????????????? = |
(???????? ???????? ????????????????????????? ???????? ???? -
???? ???? ???????????? ????????????????????) / ???? ???? ???????????? ???????????????????? |
The estimated effort using UCPabc model is closer to actual effort with deviation of 11.9%, while FP method of actual effort is 27.8%.

For software project managers, the results of this study can be taken into account when selecting estimation methods to determine the effort (and costs) required for an application development project of public services. Table 8. Summary review of comparison effort estimation toward actual effort . Project ID Real Cost (IDR) Actual Effort UCPabc�s Effort FP�s Effort 1 35,883,000 3,632 3775.92 1,873.86 2 37,908,000 3,728 3878.53 2,861.14 3 38,134,800 3,448 3956.07 2,468.77 4 74,115,000 7,045 4112.77 5,692.93 Total 186,040,800 17,853 15,723.29 12,896.71 Deviation 11.9% 27.8% 5.

Conclusion This study compares between UCPabc and FP. Some aspects of the process are parameters, complexity factors, and deviation. The results of this research were: (1) The difference of process algorithm and parameters. The difference between UCPabc and FP is seen from 4 aspects, namely: process, parameters, c omplexity factor, and profit. The difference between UCPabc and FP of these 4 aspects has been given in Table 3 and Figure 2. (2) Complexity Sholiq et al.

/ Procedia Computer Science 124 (2017) 470�477 477 Sholiq et al. / Procedia Computer Science 00 (2018) 000 �000 7 Table 7 shows the estimated effort using the FP method. In the table are given data 4 projects with 5 input parameters, namely: External Input (Exi), External Output (Exo), External Inquiry (Exiq), Internal Logic File (Ilof), and External Logic File (Elof) . Table 7 also contains the Modifi ed Complexity Adjustment Factor (MCAF) [6].

FP with the formula FP = (Exi + Exo + Exiq + IIlof + Elof) * MCAF. For ProjectID = 1, then FP = (39 + 57 + 25 + 42 + 34) * 1.16 = 228.52. Then fffort is obtained by multiplying FP with PF constant, so that effort for ProjectID=1 equals 1,873.86 man -hours. Likewise for ProjectID= 2, 3, and 4 respectively obtained effort= 2,861.14; 2,468.77; and 5,692.93 man-hours (see Table 7). The total effort for all four projects is 12,896.71 man -hours. Table 7. Effort estimation using function points (FP).

Project ID Exi Exo Exiq IloF ELof MCAF FP Effort 1 39 57 25 42 34 1.16 228.52 1,873.86 2 34 48 28 82 94 1.22 348.92 2,861.14 3 34 48 28 68 75 1.19 301.07 2,468.77 4 61 69 81 178 133 1.33 694.26 5,692.93 Total 1,572.77 12,896.71 Source: Dewi et al, 2017 [6] As is the novelty of this study, the comparison of both estimation methods will be compared with actual effort (see Table 8). Based on the results obtained from the estimation effort on each method, when compared with actual effort can be seen in Table 8.

Comparison between effort estimation with actual effort is expressed by using deviation defined that: ???????????????????????? = | (???????? ???????? ???????????????????????? ???????? ???? - ???? ???? ???????????? ????????????????????) / ???? ???? ???????????? ???????????????????? | The estimated effort using UCPabc model is closer to actual effort with deviation of 11.9%, while FP method of actual effort is 27.8%. For software project managers, the results of this study can be taken into account when selecting estimation methods to determine the effort (and costs) required for an application development project of public services. Table 8. Summary review of comparison effort estimation toward actual effort .

Project ID Real Cost (IDR) Actual Effort UCPabc�s Effort FP�s Effort 1 35,883,000 3,632 3775.92 1,873.86 2 37,908,000 3,728 3878.53 2,861.14 3 38,134,800 3,448 3956.07 2,468.77 4 74,115,000 7,045 4112.77 5,692.93 Total 186,040,800 17,853 15,723.29 12,896.71 Deviation 11.9% 27.8% 5. Conclusion This study compares between UCPabc and FP. Some aspects of the process are parameters, complexity factors, and deviation. The results of this research were: (1) The difference of process algorithm and parameters.

The difference between UCPabc and FP is seen from 4 aspects, namely: process, parameters, c omplexity factor, and profit. The difference between UCPabc and FP of these 4 aspects has been given in Table 3 and Figure 2. (2) Complexity 8 Sholiq et al. / Procedia Computer Science 00 (2018) 000 �000 factors. There are differences in complexity factors, 21 factors on UCPabc model consisting of 14 technical factors and 7 team environmental factors.

Whereas in FP there are 14 complexity factors. (3) Deviation between UCPabc and FP. UCPabc has a better average deviation (i .e 11.9 percent) compared to FP which has a deviation of 27.8 percent for 4 softwaremaking projects from public services as test data. Therefore, the UCPabc method is the closest method of effort estimation toward actual effort. Acknowledgements The material contained in this paper has received permission from its related company.

Thanks to the research institution, LPPM Universitas Internasional Semen Indonesia, and LPPM Institut Teknologi Sepuluh Nopember / DRPM Ristek Dikti, which have provided funding and a travel grant to publish this research at ISICO 2017. References [1] G. Karner, "Resource Estimation for Objectory Projects," Objective Systems SF AB, Kista, 1993. [2] 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. [3] R. S. Dewi, A. P.

Subriadi dan Sholiq, �Use Case Points - Activity Based Costing: A New Method for Software Development Cost Estimation,� Jurnal Sisfo, vol. 5, no. 3, pp. 318 -323, 2015. [4] A. e. J. G. Albrecht, "Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validat ion," IEEE Transactions on Software Engineering, vol. 9, no. 6, 1983. [5] M. Aguia r, �Function Points or Use Case Points?,� IFPUG MetricViews Summer, pp. 14-15, 2009. [6] R. S. Dewi, Sholiq dan A. P.

Subriadi, �A Modification Complexity Factor in Function Points Method for Software Cost Estimati on Towards Public Service Application,� dalam 4th Information Systems International Conference (ISICO) , Bali, 2017. [7] U. K., V. A. V. dan V. D., �A Survey on Software Effort Estimation,� dalam International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) , 2016. [8] T. Arnuphaptrairong, �Early Stage Software Effort Estimation Using Function Point Analysis: Empirical Evidence,� dalam International MultiConference of Engineers and Computer Scientists Vol II , Hongkong, 2013. [9] R. S. Dewi, Sholiq and A. P.

Subriadi, "UCPabc as an Integration Model for Software Cost Estimation," in International Conference on Science in Information Technology (ICSITech) , Balikpapan, 2016. [10] A. P. Subriadi, Sholiq dan P. A. Ningrum, �Critical Review of The Effort Rate Value in Use Case Point Method for Estimating S oftware Development Effort,� Journal of Theoretical and Applied Information Technology, vol. 59, no. 3, pp. 735 -744, 31 January 201 4. [11] R. S.

Dewi, �Software Requirement Specification for Public Permission Applications,� Trust Solution, Surabaya, 2013. [12] H. Raju dan Y. Krishnegowda, �Software Sizing and Productivity with Function Points,� Lecture Notes on Software Engineeri ng, vol. 1, no. 2, pp. 204-208, 2013.