Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 124 (2017) 415–422

Procedia Computer Science

www.elsevier.com/locate/procedia

4th Information Systems International Conference 2017, ISICO 2017, 6-8 November 2017, Bali, Indonesia

A Modification Complexity Factor in Function Points Method for Software Cost Estimation Towards Public Service Application

Renny Sari Dewi[a], Apol Pribadi Subriadi[b,*], Sholiq[b]

[a]*Information Systems Department, Universitas Internasional Semen Indonesia, Jalan Raya Veteran, Gresik 61122, Indonesia*
[b]*Information Systems Department, Institut Teknologi Sepuluh Nopember, Jalan Raya Sukolilo, Surabaya 60111, Indonesia*

Abstract

Lately, the stages of planning software development projects have begun to consider the scientific side. As an intangible outcome, the budgeting must also be done in a transparent and accountable manner. The authors use Function Points (FP) method approach on the basis of 4 main reasons for estimating the effort and cost of software development (4 public service applications as research object). In this study, there are two core phases, first, elaborating complexity factors based on other method references (i.e Use Case Points) and mapping of non-functional requirements on Term of Reference. Furthermore, the second phase is to calculate and compare the estimated effort and cost if the original FP method before and after modified on the complexity factor. We conclude there is a difference of 7.19 percent (equivalent to IDR 13,567,631) between FP method calculations using and not using modified complexity adjustment factors.

Peer-review under responsibility of the scientific committee of the 4th Information Systems International Conference 2017.

Keywords: Software Estimation; Software Size; Software Cost; Function Points; Public Service Application

1. Introduction

Good project planning nowadays has not become the main awareness by the business of information and communication technology in government institution in Indonesia. This can be seen from the lack of news about the business through Google search engine with the keyword "software cost estimation". Yet it becomes a paradox when the project planning frenzy has been missed in forum of academia and researchers.

* Corresponding author. Tel.: +00 00-0000000; fax: +00 00-0000000.
E-mail address: apolpribadi@gmail.com

1877-0509
Peer-review under responsibility of the scientific committee of the 4th Information Systems International Conference 2017
10.1016/j.procs.2017.12.172

Since 1993, Karner has initiated a scientific method to calculate the estimated software development effort called Use Case Points (UCP). But in the last 3 years, the results of research on software cost estimation showed a positive trend. According to Dewi et al [1], deviation between estimated costs and actual costs incurred by the project manager is only about 2.16 percent using the Use Case Points - Activity Based Costing (UCP_{ABC}) integration method in the public service application. While other research results suggest that cost estimation using UCP method in small and medium scale applications has only about 6.89 percent of deviation [2]. It proves that the estimated cost of using UCP scientific method is applicable and sufficiently precise, so it is feasible to be implemented by business actors in software engineering.

In the midst of the issue of electronic government (e-Gov), the government, which is one of the non-profit business in software engineering, should be aware of the estimated cost of software procurement projects, which inevitably leads to budget allocations. Eliminating or erroneously planning a software procurement project may result in less than maximum results. According to survey results conducted by McKinsey [3], faulty software project planning will impact failure by a percentage of 66% as it exceeds budget allocation, 33% due to backward from the specified schedule. McKinsey also claims that 17% of his projects actually suffered losses. In line with the results of the Standish Group survey [4], until 2015 about 71% of software development projects outline failure.

There are several methods of estimation of software development efforts that are well known, including the Use Case Points (UCP) by Karner [5]. The Function Points (FP) method was first echoed by A.J. Albrecht in 1979 which was formally declared by the International Function Points User Group (IFPUG). In addition, software cost estimates can also use the Cost Constructive Model (COCOMO) that has been published up to version 2 by B. Boehm.

But of the many methods, the authors adjust the conditions that become constraints in calculating the estimated cost of developing public service software. Some of the constraints identified are:

- Often the reference of the government to make the application of public services is only limited to the forms passed through the prevailing laws and regulations.
- The organizational structure of government that handles public services on average has similarities, thereby demonstrating the consistency of business process applications of high public services.
- The time provided for the software project development planning process is relatively short. Therefore, there is only a short time to determine the price of software development.
- The function of systems analyst or business analyst is often ignored, because the role is already included in the budget of third party expenses as software developers. Therefore, the initial reference to the formulation of needs is only the public service form and new business process engineering.

From the 4 reasons above, the authors conclude that the FP method is quite relevant to overcome the problem. One factor why FP is considered faster is because it does not require the results of system analysis in the form of narrative use case scenarios and database concepts.

The selection of 4 public service applications as presented in Table 3 is because the author has never found research on the implementation of FP methods in the realm of government. The author see a fundamental difference between public service applications and enterprise software. Therefore, this study would examine whether FP methods are still eligible for use in government applications. In the calculation of effort and cost estimation using FP method, it is possible that the author would make modification as an effort to adjust to the factors that influence the success of application of public service.

2. Related research

Previous research on the estimated effort and cost of software development has been summarized as presented in Table 1. FP has 5 main measurement parameters that are often listed on the public service form and the flow of its business processes. Five parameters include External Input, External Output, External Inquiry, Internal Logic File, and External Logic File. Of the 5 parameters, then summed up all the scores to get the value of Unadjusted Function Points (UFP). The complexity factors have been modified [5] [6] [9] affect the final weighting value before getting the value of Adjusted Function Points (AFP).

Table 1. Related research

No	Author, Year	Result	Research connectivity	Research gap
1	Albrecht, 1983 [6]	Predicted effort using Function Points (FP) based on software function and number of lines of code in IBM company.	As an initial foundation for calculating software size and justification of complexity factors.	Adjustment of complexity factors are less relevant, so modifications need to be based on the needs of the public service application.
2	Aguiar, 2009 [7]	International Function Points User Group (IFPUG) has recommended FP that has been successfully implemented in government and industry compared to Use Case Points method based on survey conducted.	As the main justification why we chose FP in the government field.	It is not stated whether FP is used to estimate software development costs for the public service sector.
3	Dewi et al, 2014 [1]	Use Case Points (UCP) method integrated with Activity Based Costing (ABC) is able to estimate development cost of 5 government applications.	The use of case study of public service application is interesting enough to be comparative study.	The method used is not FP, so the use of this method shows the novelty factor.
4	Dewi et al, 2016 [8]	The level of cost estimation accuracy using the UCPabc integration model tested by actual cost has a deviation of 2.16% and a profit of 30.4%.	The resulting deviation as a benchmark if the same case study uses the FP estimation method.	Accuracy of cost estimation of profit is the final average result of overall activity in the software development and ongoing activity phases.
5	Sholiq et al, 2016 [2]	Effort software development is distributed on 12 activities tailored to Indonesia salary guide Kelly Services in 2011-2012.	This study as an initial reference activity-based payrate.	-

3. Research method

The research method used to estimate the cost of public service application software is divided into 2 core stages, which is begun with literature study to determine software complexity factor, then continued by getting estimation of development cost.

3.1. Determine value of modification complexity adjustment factor (MCAF)

The difficulty level of a software on FP method has 14 factors [6] covering the complexity of the technical side of the development as well as the complexity of the organization's environment. We consider the complexity factor to be adjusted to the Term of Reference (TOR) of the project. TOR consists of proposed business process engineering, functional requirements and non-functional requirements for software development projects. Factors of complexity are implicitly mentioned in non-functional requirements. Therefore, a comparative study of complexity factors in the well-known cost estimation methods of FP and UCP is then compared with the predetermined project's TOR [9].

3.2. Estimate Software size and cost

The size and cost of developing the public service application using FP method has been modified based on the determination of complexity factor at step 3.1. The steps to get an estimate of the size and cost of public service applications based on Albrecht [6]:

- Count the Unadjusted Function Points (UFP). UFP is obtained by summing up 5 measurement parameters such as External Input (Exi), External Output (Exo), External Inquiry (Exiq), Internal Logic File (Ilof), and External Logic File (Elof) then multiplied by each weighting (see Table 2).

Table 2. Components of unadjusted function points

Measurement parameter	Description	Weight of component (simple–medium–hard)
Exi	Each user input that provides distinct application oriented data to the software	3 – 5 – 6
Exo	Each user output that provides application oriented information to the user	4 – 5 – 7
Exiq	An inquiry is defined as an on-line input that results in the generation of some immediate software response in the form of an online output	3 – 4 – 6
Ilof	Each logical master file	7 – 10 – 15
Elof	All machine-readable interfaces that are used to transmit information to another system	5 – 7 – 10

- Calculate the Value of Modification Complexity Adjustment Factor (MCAF)
  Value of MCAF Is the sum of all predefined factor complexity scores (see formula (1)).

$$MCAF = \left(\sum_{i=1}^{n} x_i .0,01 \right) + 0,65 \quad (1)$$

  in which: x = each complexity adjustment factor; i = count of MCAF
- Count the Adjusted Function Points (AFP)
  After obtaining UFP and MCAF scores, to get AFP value is by inserting each score into formula (2).

$$AFP = UFP * MCAF \quad (2)$$

- Determine the Productivity Factor
  Productivity Factor (PF) is the effort rate in units of function points per day (fp/day). According to Raju & Krishnegowda [10] and Sholiq et al [2], the value of PF is 8.2 PF/hour (we assume that 8 work-hour and 20 work-day in a month).
- Estimate Total Effort and Cost by Multiplying Payrate
  The total effort estimate is obtained from the multiplication of PF that has been converted in man-hours units. While the estimated cost of development using a currency unit IDR. Estimated costs through the FP method, then computed with the net value of each software development project cost. Salary guide by Kelly Services 2011-2012 [2] is required for man-month conversion into man-hours (assuming within 1 month there are 20 working days and 8 hours/day).

4. Result

Based on previous research [8], the gross and net contract value for development projects of 4 (four) public service applications is listed in Table 3.

Table 3. Summary of project software development

ID	Application name	Description	Brutto value (IDR)	Netto value (IDR)[a]
1	Industrial Registration	This application is intended for individuals / business entities in the field of small-scale industry	44,300,000	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 [11]	46,800,000	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 2010 [11]	47,080,000	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	91,500,000	74,115,000

[a] Exclude taxes for company consist of PPn 10%, 1.5% PPh article 22, and 7.5% PPh article 23

4.1 Determine value of modification complexity adjustment factor (MCAF)

Table 4 shows that software complexity factor in FP method is 14 aspects [6], UCP 13 technical aspects [2] [8], and 16 non-functional requirement based on TOR [9].

Table 4. Mapping software complexity factor vs non-functional requirements

No	Function points	Use case points	Non-functional requirements
1	Level of backup and recover reliability	Highly concurrent	System can backup and restore automatically everyday
2	Level of data communications	Required distributed systems	Data progress must be published to applicant interface
3	Level of distributed data processing	Required complex internal processing	Integrated to centralize database
4	Level of performance needs	Response time is important	Software must be optimize when work day
5	Level of environment configuration	Cross-platform support	Configuration guidance must be attached
6	Level of transaction rate	-	Must stable when 500 data input
7	Level of end-user efficiency	End user efficiency	User friendly interface
8	Level of master file update	-	Master file on server must update realtime
9	Level of online real-time update	-	Data update real-time everyday on server
10	Level of reusability	Reusable code to Focus	A few reusable code is better
11	Level of installation ease	Installation easy	Installation guidance must be attached
12	Level of operational ease	Usability	A simple tutorial when start application
13	Level of customer organisation variation	-	Type of application influence data variance
14	Level of change possibility	Easy to change	Possible to change if applicant move the data
15	-	Custom security	Consider the security with encryption code
16	-	User training	After development done, the software must be trial for several applicant and each operator

Based on the results of mapping Table 4, it is concluded that the MCAF value is influenced by 16 factors presented in Table 5.

Table 5. Modification complexity adjustment factor (MCAF)

No	MCAF	Score[b]
1	Level of reliability for recovery	○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
2	Level of data communications	○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
3	Level of distributed data processing	○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
4	Level of performance needs	○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
5	Level of environment configuration	○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
6	Level of transaction rate	○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
7	Level of end-user efficiency	○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
8	Level of master file update	○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
9	Level of online real-time update	○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
10	Level of reusability	○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
11	Level of installation ease	○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
12	Level of operational ease	○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
13	Level of customer organisation variation	○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
14	Level of change possibility	○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5

No	MCAF	Score[b]
15	Level of security	○0 ○1 ○2 ○3 ○4 ○5
16	Level of user training	○0 ○1 ○2 ○3 ○4 ○5

[b] Score 0 is Not present or no influence; 1 is Incidental influence; 2 is Moderate influence; 3 is Average influence; 4 is Significant influence; 5 is Strong impact/essentials

4.2 Estimate software size and cost

The next step is to estimate effort and cost in FP method before and after modified. Table 6 shows that there is a difference in the calculation of FP results and the estimated effort both before and after modification. The first two columns are FP method calculations before modifications are made on the complexity factor which still totals 14 items (see Table 4 in first column). While the next 2 columns, the modification of FP method is done by entering 16 items of Modified Complexity Adjustment Factors (MCAF) as shown in Table 5. Therefore, the authors then detail the results of FP calculation and the estimated effort before modification is 1.454.33 and 12.927.38. As for the calculation after modified is 1.572.77 and 13.980.18 (see Table 6). Then Table 7 shown the total effort estimate is distributed into 12 activities of software development and project management.

Table 6. Amount of FP and effort estimation after modification

ID	Total FP	Total effort before modification[a]	Total modified FP	Total effort after modification[c]
1	210.79	1,873.69	228,52	2,031.29
2	323.18	2,872.71	348,92	3,101.51
3	278.30	2,473.78	301,07	2,676.18
4	642.06	5,707.20	694,26	6,171.20
Total	1,454.33	12,927.38	1,572.77	13,980.18

[c] in man-hour unit

Table 7. Comparison of effort distribution before and after modification

No	Activities	Distributed effort before modification	Distributed effort after modification
1	Requirements	206.84	223,68
2	Specifications	969.55	1.048,51
3	Design	775.64	838,81
4	Implementation	6,722.24	7.269,69
5	Integration testing	904.92	978,61
6	Acceptance & deployment	711.01	768,91
7	Project management	491.24	531,25
8	Configuration management	555.88	601,15
9	Quality assurance	116.35	125,82
10	Documentation	1,085.90	1.174,33
11	Training and support	129.27	139,80
12	Evaluation and testing (warranty)	258.55	279,60
	TOTAL (man-hours)	12,927.38	13.980,18

The result of the estimation of the public service application development effort is then elaborated with the pay rate of each activity that has been converted to a man-hours unit [2] (see Table 8).

Table 8. The comparison of cost distribution before and after modification

No	Activities	Tariff (IDR) (man-hours)	Cost estimation (IDR) before modification	Cost estimation (IDR) after modification
1	Requirements	17.187,50	3,555,029	3,844,549
2	Specifications	17.187,50	16,664,198	18,021,323
3	Design	10.312,50	7,998,815	8,650,235
4	Implementation	10.312,50	69,323,063	74,968,703
5	Integration testing	10.312,50	9,331,951	10,091,941
6	Acceptance & deployment	10.312,50	7,332,247	7,929,382
7	Project management	34.375,00	16,886,387	18,261,607
8	Configuration management	34.375,00	19,108,280	20,664,450
9	Quality assurance	10.312,50	1,199,822	1,297,535
10	Documentation	10.312,50	11,198,341	12,110,329
11	Training and support	10.312,50	1,333,136	1,441,706
12	Evaluation and testing (warranty)	10.312,50	2,666,272	2,883,412
	TOTAL (IDR)		166,597,541	180.165.172

Table 9 shows the comparison of estimated costs using FP method before and after modified toward actual cost. Deviation is obtained by means of absolute difference of each estimation result compared with real cost which then multiplied with 100 percent. FP method without modification resulted in a deviation of 10.45 percent, while the modified FP method with MCAF generated in a smaller deviation of 3.26 percent. This means that the difference between the two method is 7.19 percent better if the FP method modified by its complexity factor using 16 items.

Table 9. Deviation between actual cost, fp and modified fp cost estimation

ID	Actual cost (IDR)	FP cost estimation (IDR)	Modified FP cost estimation (IDR)
1	35,883,000	24,206,254	26,177,601
2	37,908,000	36,959,768	39,969,755
3	38,134,800	31,891,199	34,488,405
4	74,115,000	73,540,320	79,529,412
Total	186,040,800	166,597,541	180,165,172
Deviation		10.45%	3.26%

5. Conclusion

This research has produced an important formula in estimating the cost of software development projects, especially in the field of public service applications. Based on the results of the above research, we conclude:

- The Function Points can be used as an estimation method for software development projects cost, in this case for 4 public service applications.
- In the FP method, the factor of complexity that Albrecht has stated [7] is 14 items. While the authors make modifications by matching factors that affect the software cost estimation method Use Case Points and non-functional requirements on the Term of Reference (TOR). From the mapping results obtained Modified Complexity Adjustment Factor (MCAF) amounted to 16 items, such as: 1) level of reliability for recovery, 2) level of data communications, 3) level of distributed data processing, 4) level of performance needs, 5) level of environment configuration, 6) level of transaction rate, 7) level of end-user efficiency, 8) level of master file update, 9) level of realtime online update, 10) level of reusability, 11) level of ease of installation, 12) level of

operational ease, 13) level of customer organization variation, 14) level of change possibility, 15) level of security, and 16) level of user training

- The result of analysis of 4 public service application is the author got the result of software development cost estimation using FP method before and after modification that is 10.45 percent (IDR 19,443,259) and 3.26 percent (IDR 5,875,628) against actual cost of IDR 186,040,800.
- From the above comparison (Table 9), the result of cost estimation using the modified FP method of complexity factor becomes more accurate 7.19 percent (equivalent to IDR 13,567,631) than before modification. Therefore, the budgeting of public service application development projects can adopt from FP method with MCAF (16 items).

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] R. S. Dewi, A. P. Subriadi and 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.

[2] Sholiq, A. P. Widodo, T. Sutanto and 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] M. Bloch, S. Blumberg and J. Laartz, "Digital McKinsey," McKinsey & Company, October 2012. [Online]. Available: http://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/delivering-large-scale-it-projects-on-time-on-budget-and-on-value. [Accessed 4 June 2017].

[4] S. Hastie and S. Wojewoda, "InfoQueue," Infoq, 4 October 2015. [Online]. Available: https://www.infoq.com/articles/standish-chaos-2015. [Accessed 4 June 2017].

[5] G. Karner, "Resource Estimation for Objectory Projects," Objective Systems SF AB, Kista, 1993.

[6] A. e. J. G. Albrecht, "Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation," *IEEE Transactions on Software Engineering,* vol. 9, no. 6, 1983.

[7] M. Aguiar, "Function Points or Use Case Points?," *IFPUG MetricViews Summer,* pp. 14-15, 2009.

[8] 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.

[9] R. S. Dewi, "Software Requirement Specification for Public Permission Applications," Trust Solution, Surabaya, 2013.

[10] H. Raju and Y. Krishnegowda, "Software Sizing and Productivity with Function Points," *Lecture Notes on Software Engineering,* vol. 1, no. 2, pp. 204-208, 2013.

[11] Pemkot Surabaya, "JDIH Surabaya," 2010. [Online]. Available: https://jdih.surabaya.go.id/pdfdoc/perda_165.pdf. [Accessed 29 August 2017].