Journal of Soft Computing Exploration

Homepage: https://shmpublisher.com/index.php/joscex

p-ISSN: 2746-7686
e-ISSN: 2746-0991

.

Activity-based function point complexity of use case diagrams for software effort estimation

Puguh Jayadi[1*], Renny Sari Dewi[2], Kelik Sussolaikah[3]
[1, 3]Department of Informatic, Universitas PGRI Madiun, Indonesia
[2]Department of Digital Business, Universitas Negeri Surabaya, Indonesia

Article Info

Article history:

Received November 25, 2023
Revised December 19, 2023
Accepted January 24, 2024

Keywords:

Function point analysis
Use case diagrams
Software effort estimation
Adjusted function points
Estimation accuracy

ABSTRACT

This study proposes a function point analysis (FPA) based software development effort estimation methodology integrated with use case diagrams. These methods include identifying actor activities, classifying those activities into FPA categories, and calculating unadjusted function points (UFP). Followed by the calculation of technical complexity factors (TCF) and Adjusted Function Points (AFP), this study aims to produce more accurate estimates of man-hours. The results show a UFP of 162 TCF of 11, AFP of 123.12, and an estimated effort of 1846.8 hours worked, while the actual effort is 1228 hours. Evaluation of estimates using the metrics Mean Magnitude of Relative Error (MMER) 0.34, Mean Magnitude of Relative Error (MMRE) 0.50, Mean Absolute Error (MAE) 618.80, Mean Balanced Relative Error (MBRE) 0.50, Mean Inverse Balanced Relative Error (MIBRE) 0.34, and Root Mean Squared Error (RMSE) 618.80, showed sufficient precision despite the overestimation. The study suggests the need to adjust the TCF calculations and considering development environment in more detail to improve the accuracy of the estimate. These findings are essential in improving the precision of effort estimation methodologies in software development, particularly in projects that use use case diagrams as the primary framework.

Corresponding Author:

Puguh Jayadi,
Department of Informatics,
Universitas PGRI Madiun,
Jl. Auri No.14-16, Kanigoro, Kartoharjo, Kota Madiun, Jawa Timur 63117, Indonesia
Email: puguh.jayadi@unipma.ac.id
https://doi.org/10.52465/joscex.v5i1.252

1. INTRODUCTION

The development of information technology provides many conveniences and benefits to various aspects of human life [1]. Software development is becoming increasingly complex in this digital transformation era, making software business estimation a critical component in project management. The business process is a series of interrelated activities to achieve a goal, which is carried out by the system in parallel or sequentially [2]. Effort estimation is a process by which one can predict the development time and cost to develop a software process or product [3]. Traditional effort estimation techniques are frequently required to be revised to cope with the increasing complexity of software projects. As done by [4] which combines use case point (UCP) with Artificial Neural Network. In this context, Function Point Analysis (FPA) is emerging as an effective method that offers objective measurement of user functionality [5]. However, adjusting to contemporary complexity, especially regarding use case diagrams, remains a significant challenge.

Previous research, as described in [6], has explored the use of genetic algorithms in FPA, while [7] incorporates machine learning techniques to predict the estimated effort. Both approaches offer valuable insights but need more attention to the specific dynamics of Use Case Diagrams in estimation. The use case diagram is used to determine what requirements are needed from the system [8]. Within [9] they took an essential step by incorporating use case diagrams into FPAs. However, their methods still need to fully accommodate the diverse complexity of the activities depicted in those diagrams. Further [10] proposes the use of complexity graphs to measure FPA, but the approach must integrate the elements effectively and thoroughly.

A significant limitation in the literature lies in the lack of a methodology that specifically addresses the components of use case diagrams in the context of FPAs [11]. Many existing approaches need to generalize or pay more attention to the specific nuances of the described activity, which can lead to less precise estimates. This research proposes a new method that integrates in-depth Use Case Diagrams with Function Points analysis to produce more accurate work estimates. This unique approach explicitly targets the complexity derived from the interactions between various elements in Use Case Diagrams, which should have been considered in previous research. By considering each activity and its exchanges, this study aims to develop a more holistic and precise framework in software business estimation.

There are several approaches taken by one of the previous studies such as combining the Constructive Cost Model II (COCOMO II) and the Artificial Neural Network (ANN) The results provide values that are close to the actual effort data, but there are shortcomings due to the complexity of the ANN with many parameters [12]. This paper aims to fill in existing gaps by providing new insights into how Use Case Diagrams can be effectively integrated into FPAs. This approach is expected to improve the accuracy of the effort estimates, which directly contributes to more efficient and effective project management. This is critical given the importance of accurate estimation in resource allocation, scheduling, and budget management in software development projects.

2. METHOD

This study takes a methodical approach to incorporating use case diagrams into Function Point measurements in order to produce a more accurate evaluation of software efforts. Figure 1 depicts the research method, which entails finding, categorizing, and calculating numerous components linked to Function Points and their use in the context of Use Case Diagrams. Begin with a description of the use case diagram and then count components using function point analysis and Effort Evaluation.

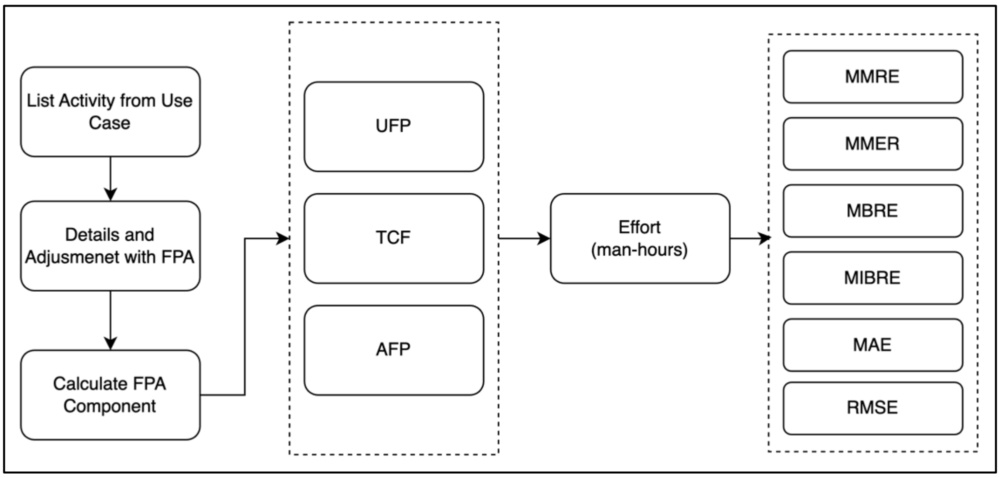

Figure 1. Research method

From Figure 1 these methodological steps are designed to obtain a more holistic and detailed picture of the effort required in a software development project. The following steps are followed:

1) Activity Registration of Each Actor on the Use Case Diagram: First, all activities performed by each actor in the Use Case Diagram are registered. This includes the actions they perform or receive in the system. This step is essential to understand all the interactions that occur, and the elements involved in the system.
2) Activity Details and Adjustments to FPA Components: Each activity is then detailed and categorized according to the five components of the Function Point: External Inputs (EI), External Outputs (EO), External Inquiries (EQ), Internal Logical Files (ILF), and External Interface Files (EIF). This process involves an in-depth analysis of how each activity relates to these components [13].
3) Complexity Calculation for FPA Components: Each component of FPA (EI, EO, EQ, ILF, EIF) is then calculated based on its complexity: Simple, Average, or Complex [14]. Table 1 involves evaluating the level of difficulty, volume of data, and interaction with other components of the system.

Table 1. Component classification

Complexity	Simple	Average	Complex
External Inputs (EI)	1	1 - 5	> 5
External Outputs (EO)	1 - 2	3 - 4	> 5
External Inquiries (EQ):	1 - 3	4 - 5	> 5
Internal Logical Files (ILF):	1 - 2	3 - 4	> 4
External Interface Files (EIF):	1 - 2	3 - 4	> 4

4) Unadjusted Function Point (UFP) Calculation: Based on the previous calculation, the Unadjusted Function Point is calculated [15]. UFP is the total of all component functions of the function point, calculated based on their respective values adjusted for complexity. UFP is calculated by summing the weights of each FPA component (EI, EO, EQ, ILF, EIF) based on its complexity [16].

$$\mathrm{UFP} = \sum(n\mathrm{EI}\times w\mathrm{EI}) + \sum(n\mathrm{EO}\times w\mathrm{EO}) + \sum(n\mathrm{EQ}\times w\mathrm{EQ}) + \sum(n\mathrm{ILF}\times w\mathrm{ILF}) + \sum(n\mathrm{EIF}\times w\mathrm{EIF}) \quad (1)$$

5) Technical Complexity Factors (TCF) Calculation: Then the technical complexity factors are calculated. The TCF describes the factors that affect the technical and software development environment, which can affect the overall complexity of the project, as shown in Table 2. Filled with values between 0 and 5, where 0 is not of influence and 5 is very influential. The formula calculates TCF:

$$TCF = 0.65 + 0.01 \times \sum \mathrm{CF} \quad (2)$$

Table 2. Technical complexity factors

Complexity	Value
Data communications	0-5
Distributed data processing	0-5
Performance	0-5
Heavily used configuration	0-5
Transaction rate	0-5
On-Line data entry	0-5
End-user efficiency	0-5
On-Line update	0-5
Complex processing	0-5
Reusability	0-5
Installation ease	0-5
Operational ease	0-5
Multiple sites	0-5
Facilitate change	0-5

6) Adjusted Function Point (AFP) Calculation: Using UFP and TCF, the adjusted function point is calculated. The AFP provides a more accurate representation of the effort required, considering the technical complexity and development environment (Park et al., 2016). AFP is calculated by multiplying UFP by TCF using the formula:

$$AFP = UFP \times TCF \quad (3)$$

7) Effort (man-hours) Calculation: The next step is to calculate the effort required (in man-hours) based on the AFP. This analysis method involves the use of predefined formulas or models, which may consider factors such as team efficiency, experience, and tools used [17]. The effort required is calculated using AFP and a coefficient that determines the *Productivity Rate (PR)* per Function Point, with the formula:

$$\text{Effort (man-hours)} = AFP \times \textit{Productivity Rate (PR)} \quad (4)$$

8) Evaluate Effort with Statistical Metrics: After the calculation of effort (man-hours) is completed, the next step is to evaluate the results using a series of statistical metrics to validate the accuracy and reliability of the estimates. Using these metrics comprehensively evaluates how close the estimate is to the actual value. Identify areas where the estimation method may need adjustment or improvement. This evaluation is essential to ensure that the methodology developed is theoretical, but also practical and reliable in actual use. These metrics include Estimation Evaluation using the metrics Mean Magnitude of Relative Error (MMER), Mean Magnitude of Relative Error (MMRE), Mean Absolute Error (MAE), Mean Balanced Relative Error (MBRE), Mean Inverse Balanced Relative Error (MIBRE) and Root Mean Squared Error (RMSE) [18], [19].

3. RESULTS AND DISCUSSIONS

The purpose of this study is to calculate the value of effort with function points based on the activity details of the use case diagram. The project data used are from the management system project and the Early Childhood Islamic Education Portal (Pendididkan Islam Anak Usia Dini - PIAUD) with the Use Case Diagram as shown in Figure 2. All data used in this research weighting are based on interviews with all parties involved in developing PIAUD, both programmers and project managers.

The initial step of the study is to identify and record the activities of each actor in use case diagrams. This process involves an analysis of the actor's interaction with the system, including data input, decision making, and output acceptance [20]. These activities are classified by type of interaction to understand the complexity of the functions involved. This analysis reveals that role variations and actor interactions significantly impact system complexity. Logging each interaction in detail provides a better understanding of the workload and complexity of functions in the system, which is essential for determining the number of Function Points [21]. Iterative validation with stakeholders is also carried out to ensure the completeness of the data and a detailed understanding of user needs. This step is an essential foundation for establishing the basis for analyzing software development business estimates.

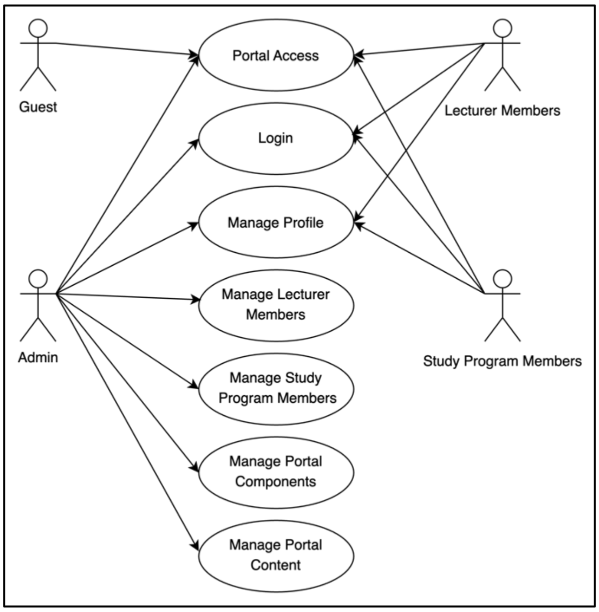

Figure 2. Use case diagram PIAUD portal

After identifying the activities of each actor, the next step is to detail these activities and categorize them according to the elements of the Function Point: External Inputs (EI), External Outputs (EO), External Inquiries (EQ), Internal Logical Files (ILF), and External Interface Files (EIF). This process involves careful analysis of how each activity interacts with the system [22]. For example, activities that send data to a system

.

are classified as EI, while activities that generate data or reports from the system are classified as EO. EQ is identified from activities that request information from the system without significant data changes. ILF and EIF are associated with the management and interaction of the internal and external data [23].

The third step is to calculate the values for each EI, EO, EQ, ILF, and EIF based on their complexity: Simple, Average, and Complex. This calculation requires an assessment of each element based on factors such as the amount of data involved, interactions with other features, and processing needs. The details of the classification are in Table 2. Simple Complexity is usually given to more direct activities requiring less data processing. In contrast, Complex is given to more complicated activities involving various elements of the system or require more complex processing logic. This calculation provides the unadjusted function point (UFP) figure, which forms the basis for estimating further development efforts. This step is essential to ensure that the effort estimate reflects the true complexity of the functional requirements [24].

After determining the level of complexity for each External Input (EI), External Output (EO), External Inquiry (EQ), Internal Logical File (ILF) and External Interface File (EIF), the study proceeded to calculate Unadjusted Function Points (UFP) [25]. This step aggregates points from all function elements based on their complexity. The UFP gives a rough idea of the functional size of the software developed using Formula 1. The results of the UFP assessment are presented in Table 3.

Table 2. Component classification

Actor	Activities	EI	Cpl EI	EO	Cpl EO	EQ	Cpl EQ	ILF	Cpl ILF	EIF	Cpl EIF
Admin	Login	1	S	0		1	S	0		0	
	Manage Profile	1	S	1	S	1	S	1	S	0	
	Manage Lecturer Members	1	S	2	S	1	S	1	S	0	
	Manage Study Program Members	1	S	2	S	1	S	2	S	0	
	Manage Portal Components	1	S	1	S	1	S	2	S	0	
	Manage Portal Content	1	S	1	S	1	S	2	S	0	
Lecturer Members	Login	1	S	0		0		1	S	0	
	Manage Profile	1	S	1	S	1	S	1	S	0	
Study Program Members	Login	1	S	0		0		1	S	0	
	Manage Profile	1	S	1	S	1	S	1	S	0	
Guest, Admin, Lecturer Member, Study Program Member	Portal Access	1	S	1	S	1	S	1	S	0	

S = Simple, A = Average, C = Complex

Table 3. Value unadjusted function point (UFP)

Complexity	*w* Simple	Total	*w* Average	Total	*w* Complex	Total	UFP
External Inputs (EI)	3	11	4	0	6	0	33
External Outputs (EO)	4	8	5	0	7	0	32
External Inquiries (EQ)	3	9	4	0	6	0	27
Internal Logical Files (ILF)	7	10	10	0	15	0	70
External Interface Files (EIF)	5	0	7	0	10	0	0
		Total UFP					162

$TCF = 0.65 + 0.01 \times \sum CF = 0.65 + 0.01 \times 11 = 0.76$

$AFP = UFP \times TCF = 0.76 \text{ x } 162 = 123.12$

The next step is to calculate the effort, expressed in man-hours, required for software development. This calculation is based on AFP and multiplied by Productivity Rate (PR)=15 as in Formula 4 [17].

Effort (man-hours) = $123.12 \times 15 = 1846.8$

Finally, the study evaluated this effort estimation using various metrics such as Mean Magnitude of Relative Error (MMRE), Median Magnitude of Relative Error (MMER), Balanced Relative Error (BRE), Mean Inverted Balanced Relative Error (MIBRE), Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). These metrics are used to measure the accuracy of the effort estimate, providing a view of how close the estimate is to the actual effort known to be 1228-man hours. This evaluation is important for the validation of the estimation methods used and to determine areas that require adjustment or improvement in the research methodology. The evaluation results are as shown in Table 5.

Table 5. Value Evaluation

Matrix	Value
Mean Magnitude of Error Relative (MMER)	0.34
Mean Magnitude of Relative Error (MMRE)	0.50
Mean Absolute Error (MAE)	618.80
Mean Balanced Relative Error (MBRE)	0.50
Mean Inverse Balanced Relative Error (MIBRE)	0.34
Root Mean Squared Error	618.80

Based on the application of the proposed methodology, the estimated interim results of the effort are as follows. Unadjusted Function Point (UFP) of 162 Technical Complexity Factors (TCF) of 11, Adjusted Function Point (AFP) of 123.12, and estimated effort in working hours (Man Hour) of 1846.8. It should be noted that the actual effort, that is, the actual effort required in the project, is 1228 hours [18], [19].

The results of the effort estimation show that the estimates generated from this methodology are significantly different from the actual effort required in the project. MMRE, MdMRE, and other evaluation metrics will be used to evaluate the estimation error rate in greater depth. These results will be analyzed to identify patterns in estimation error and the factors that influence it [26]. Furthermore, these results provide insight into the estimated level of complexity of the project. A TCF of 11 indicates significant technical and environmental factors that affect the project effort. This analysis provides a deeper understanding of what may have been overlooked in previous attempts to estimate [27].

These interim results imply that the proposed estimation method requires further review to improve the accuracy of effort estimates. The significant difference between the estimation and the actual effort indicates the potential to improve the methodology or consider additional factors that influence the estimates. This stage concludes that this research has provided an initial understanding of using use case diagrams in software business estimation, but further improvements are needed to achieve more accurate estimates.

Limitations in this study include assumptions made in using the methodology and limitations in the data used for testing. Acknowledging these limitations is essential to provide context to the reader and avoid misinterpreting the results. As a next step, the study will further analyze the estimation errors, focusing on identifying factors that influence significant differences between estimates and actual effort. The follow-up research plan also includes an exploration of methods of improvement or adjustment in the proposed estimation approach. This aims to improve the accuracy and relevance of software business estimation in developing use case diagram-based development.

Perhaps this result is not better than other studies that use development from other Use Case Points such as Fuzzy [28] or in-depth improvisation in the calculation of the Use Case Point itself [18], [29]. It is critical to keep in mind that creating a more accurate software effort estimating approach begins with these preliminary results. This research can significantly improve software effort estimation techniques by providing more precise and reliable results with further in-depth investigation and review.

4. CONCLUSION

The original goal of this research was to combine Function Point measurements with Use Case Diagrams to create a more precise method for estimating software effort. However, interim findings indicate that more adjustments to effort estimations may be necessary to achieve the desired degree of accuracy. However, these results motivate more research to create more effective techniques. This investigation has revealed several areas where the suggested estimating approach needs to be strengthened. Prospects for development include additional research into the variables influencing estimation mistakes, improved usage of use case diagrams, and more thorough modeling of the technical complexity of the project. Our goal is to make this method better over time so that it may be used for software business estimation.

This research not only develops but also provides avenues for further application in routine software development. Software development organizations can implement this strategy to improve their project planning and management by gaining a deeper understanding of project complexity. Applying this research can help organizations avoid underestimating or overestimation of effort that often leads to problems in software development.

.

CREDIT AUTHORSHIP CONTRIBUTION STATEMEN

Author1: Conceptualization, Methodology, Software, Project administration. **Author 2:** Review & editing. **Author 3:** Review & editing.

REFERENCES

[1] A. A. Nurdin, G. N. Salmi, K. Sentosa, A. R. Wijayanti, and A. Prasetya, "Utilization of Business Intelligence in Sales Information Systems," *J. Inf. Syst. Explor. Res.*, vol. 1, no. 1, pp. 39–48, Dec. 2022, doi: 10.52465/joiser.v1i1.101.

[2] R. Naufalia, S. A. Usman, and C. L. Bambang, "Analysis and development of company business processes using business process model notation (case study of PT Datacomm Diangraha)," *J. Soft Comput. Explor.*, vol. 2, no. 2, Sep. 2021, doi: 10.52465/joscex.v2i2.48.

[3] A. Purwanto and L. Parningotan Manik, "Software Effort Estimation Using Logarithmic Fuzzy Preference Programming and Least Squares Support Vector Machines," *Sci. J. Informatics*, vol. 10, no. 1, pp. 1–12, 2023, doi: 10.15294/sji.v10i1.39865.

[4] S. S. Ali, J. Ren, K. Zhang, J. Wu, and C. Liu, "Heterogeneous Ensemble Model to Optimize Software Effort Estimation Accuracy," *IEEE Access*, vol. 11, pp. 27759–27792, 2023, doi: 10.1109/ACCESS.2023.3256533.

[5] A. Y. P. Putri, "Modifikasi Metode Function Point Dengan Menambahkan Kompleksitas Proses Bisnis Pada General System Characteristics Untuk Estimasi Biaya Perangkat Lunak," Institut Teknologi Sepuluh Nopember, 2018.

[6] M. Lefley and M. J. Shepperd, "Using Genetic Programming to Improve Software Effort Estimation Based on General Data Sets BT - Genetic and Evolutionary Computation — GECCO 2003," E. Cantú-Paz, J. A. Foster, K. Deb, L. D. Davis, R. Roy, U.-M. O'Reilly, H.-G. Beyer, R. Standish, G. Kendall, S. Wilson, M. Harman, J. Wegener, D. Dasgupta, M. A. Potter, A. C. Schultz, K. A. Dowsland, N. Jonoska, and J. Miller, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 2477–2487.

[7] Z. Polkowski, J. Vora, S. Tanwar, S. Tyagi, P. K. Singh, and Y. Singh, "Machine Learning-based Software Effort Estimation: An Analysis," in *2019 11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, IEEE, Jun. 2019, pp. 1–6. doi: 10.1109/ECAI46879.2019.9042031.

[8] S. Febriyanti and S. Solehatin, "Application design for web-based car services to increase work time estimates," *J. Student Res. Explor.*, vol. 2, no. 1, pp. 11–21, Jan. 2024, doi: 10.52465/josre.v2i1.231.

[9] A. B. Nassif, M. Azzeh, A. Idri, and A. Abran, "Software Development Effort Estimation Using Regression Fuzzy Models," *Comput. Intell. Neurosci.*, vol. 2019, pp. 1–17, Feb. 2019, doi: 10.1155/2019/8367214.

[10] B. K. Park, S. Y. Moon, and R. Y. C. Kim, "Improving Use Case Point (UCP) Based on Function Point (FP) Mechanism," in *2016 International Conference on Platform Technology and Service (PlatCon)*, IEEE, Feb. 2016, pp. 1–5. doi: 10.1109/PlatCon.2016.7456803.

[11] R. S. Dewi, A. P. Subriadi, and Sholiq, "A Modification Complexity Factor in Function Points Method for Software Cost Estimation Towards Public Service Application," *Procedia Comput. Sci.*, vol. 124, pp. 415–422, 2017, doi: 10.1016/j.procs.2017.12.172.

[12] J. Rashid, S. Kanwal, M. Wasif Nisar, J. Kim, and A. Hussain, "An Artificial Neural Network-Based Model for Effective Software Development Effort Estimation," *Comput. Syst. Sci. Eng.*, vol. 44, no. 2, pp. 1309–1324, 2023, doi: 10.32604/csse.2023.026018.

[13] B. Kumar, "Function point analysis based effort estimation and prediction using Lagrange's interpolation in Agile software development," *Math. Eng. Sci. Aerosp.*, vol. 14, no. 2, pp. 395–416, 2023.

[14] H. T. Hoc, V. Van Hai, H. L. T. K. Nhung, and R. Jasek, "Improving the Performance of Effort Estimation in Terms of Function Point Analysis by Balancing Datasets," 2023, pp. 705–714. doi: 10.1007/978-3-031-21435-6_60.

[15] J. T. M. Dhas and J. Midhunchakravarthy, "Modern Metrics (MM): Software size estimation using function points for artificial intelligence and data analytics applications and finding the effort modifiers of the functional units using indian software industry," *J. Discret. Math. Sci. Cryptogr.*, vol. 26, no. 3, pp. 629–640, 2023, doi: 10.47974/JDMSC-1734.

[16] A. Farhan, "Penggunaan metode use case point activity-based costing dan adjusted function point untuk estimasi biaya pembuatan software," 2021. [Online]. Available: https://repository.uinjkt.ac.id/dspace/handle/123456789/65074%0Ahttps://repository.uinjkt.ac.id/dspace/bitstream/123456789/65074/1/ANDRIA FARHAN-FST.pdf

[17] R. S. Dewi, T. W. Andari, A. P. Subriadi, and Sholiq, "Function Points Method in Game Casual Context," in *2018 International Conference on Electrical Engineering and Computer Science (ICECOS)*, IEEE, Oct. 2018, pp. 367–372. doi: 10.1109/ICECOS.2018.8605188.

[18] P. Jayadi, A. C. Aria Bima, Y. P. Yudha, and Kelik Sussolaikah, "End User Development pada Use Case Point untuk peningkatan Estimasi Perangkat Lunak," *TEMATIK*, vol. 10, no. 1, pp. 74–82, Jun. 2023, doi: 10.38204/tematik.v10i1.1289.

[19] P. Jayadi, Juwari, L. Azis, and K. Sussolaikah, "Estimasi Pengembangan Perangkat Lunak Dengan Use Case Size Point," vol. 3, pp. 332–340, Mar. 2023, doi: 10.47065/bit.v3i1.408.

[20] N. Marcheta, "Effort Estimation Modeling Of E-Government Application Development Using Function Points Based On Tor And Srs Document," *J. Inf. Technol. Its Util.*, vol. 3, no. 1, p. 5, Aug. 2020, doi: 10.30818/jitu.3.1.2839.

[21] M. Baiquni, R. Sarno, Sarwosri, and Sholiq, "Improving the accuracy of COCOMO II using fuzzy logic and local calibration method," in *2017 3rd Int. Conf. Sci. Inf. Technol. (ICSITech)*, IEEE, Oct. 2017, pp. 284–289. doi: 10.1109/ICSITech.2017.8257126.

[22] H. Hamzah, R. Saptono, and R. Anggrainingsih, "Development of Software Size Estimation Application using Function Point Analysis (FPA) Approach with Rapid Application Development (RAD)," *ITSmart J. Teknol. dan Inf.*, vol. 5, pp. 96–103, Dec. 2016, doi: 10.20961/its.v5i2.1988.

[23] S. Sariyanti and A. Ardiansyah, "Pengembangan Kakas Estimasi Perangkat Lunak dengan Function Point dan Use Case Point untuk Praktikum Rekayasa Perangkat Lunak," *J. Sarj. Tek. Inform.*, vol. 6, no. 2, pp. 89–97, 2018, doi: 10.12928/jstie.v6i2.15233.

[24] Sholiq, R. S. Dewi, and A. P. Subriadi, "A Comparative Study of Software Development Size Estimation Method: UCPabc vs Function Points," *Procedia Comput. Sci.*, vol. 124, pp. 470–477, 2017, doi: 10.1016/j.procs.2017.12.179.

[25] F. A. Juyuspan and A. Hidayati, "Estimasi Pengelolaan Suatu Proyek Dalam Pengembangan Perangkat Lunak Menggunakan Analisa Function Point," *J. SimanteC*, vol. 5, no. 2, pp. 85–92, 2016.

[26] S. Shukla and S. Kumar, "Towards ensemble-based use case point prediction," *Softw. Qual. J.*, vol. 31, no. 3, pp. 843–864, Sep. 2023, doi: 10.1007/s11219-022-09612-2.

[27] J. F. Vijay, "Retraction Note: Enrichment of accurate software effort estimation using fuzzy-based function point analysis in

business data analytics," *Neural Comput. Appl.*, vol. 35, no. 6, pp. 4797–4797, Feb. 2023, doi: 10.1007/s00521-022-08159-4.

[28] M. Hariyanto and R. S. Wahono, "Estimasi Proyek Pengembangan Perangkat Lunak Dengan Fuzzy Use Case Points," *IlmuKomputer.com J. Softw. Eng.*, vol. 1, no. 1, pp. 54–63, 2015.

[29] A. Srivastava, S. S.K, and S. Q. Abbas, "Advancement Of UCP With End User Development Factor: AUCP," *Int. J. Softw. Eng. Appl.*, vol. 6, no. 2, pp. 01–10, Mar. 2015, doi: 10.5121/ijsea.2015.6201.

.