Function Points Method in Game Casual Context

Renny Sari Dewi
Department of Information Systems
Universitas Internasional Semen Indonesia
Gresik, Indonesia
renny.rsd@gmail.com

Trias Widha Andari
Dept. of Visual Communication Design
Universitas Internasional Semen Indonesia
Gresik, Indonesia
trias.andari@uisi.ac.id

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

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

***Abstract*— Function Points (FP) method has been trusty and proven to be effective to estimate the effort of thousands of software development projects. Nowadays, the mobile game development—especially in a casual genre—is growing. Therefore, the researcher is excited to test the FP method whether it can also be applied to estimating the game development effort. FP method usually consists of 3 main stages: weighing 5 parameters of Unadjusted FP, calculating 14 complexity factors, and calculating Adjusted FP. Survey to 8 casual games was conducted and the value of effort for each one of them has been estimated. So, we conclude that FP is not suitable to forecast the effort of mobile casual games. There are 376 percent deviation between effort estimations and actual effort. The authors suspect, for mobile game development, the game's point of view needs to be re-adjusted to 5 input parameters, complexity factors, conversion of programming languages, and productivity rate.**

Keywords—function points, effort estimation, mobile game, game effort

I. Introduction

According to the Newzoo's report in 2017, game market revenues reached US$ 108.9 billion. Forty-two percent (42%) of it was lead by mobile games [1]. In Indonesia, the number of gamers was as big as 43.7 million (ranked 16th worldwide in terms of game revenues).

ESA classified games into 4 groups [2]: casual/ social, multi-player universe, strategy/ role-playing, and puzzle/ trivia. Since 2014, it is known that most users are playing casual/ social games [3]. That is the reason why we are interested in using casual games as our case study of development effort estimation.

Despite the glory of a digital game market in this country, business actors and researchers are still facing difficulties in finding the ideal method to estimate the effort of a game development project. The cost of casual games, as the one dominating game market, needs to be estimated cautiously. Psychologically, the real challenge in a casual game is to postpone the players' boredom until the end of the game.

As a researcher, there needs to be scientific proof of a method that has proven effective in estimating software development efforts. One of them is the oldest effort estimation method, introduced by Alan Albrecht in 1979, Function Points [4] [5] [6] [7]. Use Case Points (UCP) method, introduced by Gustav Karner [8] [9], could possibly be used, too. Other than FP and UCP, there are also other methods, such as Cocomo, Cocomo II, Analogy, and many more. The question is, are those effort estimation method feasible for mobile casual games?

Digital mobile games are based on 3 main factors: dynamics, aesthetics, and mechanics [10]. In this research, we try to apply the Function Points (FP) method to calculate mobile casual game development effort. Here are our considerations when choosing that method:

- FP method has been improved significantly, to the point that there is even the International Function Points User Group (IFPUG) established in 1987 [11],
- FP rule is widely used to estimate the effort of small, intermediate, and big software development projects [12] [13],
- FP is considered better than Use Case Points (UCP), Cost Constructive Models (Cocomo) I and II [14] [15].

II. Related Research

Function Points (FP) method was popularized by Allan Albrecht in 1979. Back then, it was used to estimate the effort of software development in his corporation [6].

This method has been practiced in previous researches of software development project's effort estimation. Here are a few reasons why.

- FP can be applied in various scales—from small to big—of software development project [5] [7].
- The accuracy level of FP can reach 11% in 5 software development job [18].
- FP can be used in projects based on geographic information system (GIS) [19].
- FP scores can be improved by modifying technical complexity level. In a case study, they can go down from 27.5% to 10.45% (approximately 17%) compared to the actual effort of four public service applications [13] [12].

As the science developing, fulfillment of FP method by hundreds of researchers around the world has shown impressive results. That is why we are conducting another research to predict effort in a different case, mobile casual game, by using the FP approach. Fig. 1 present the steps needed to get the estimated score of effort.

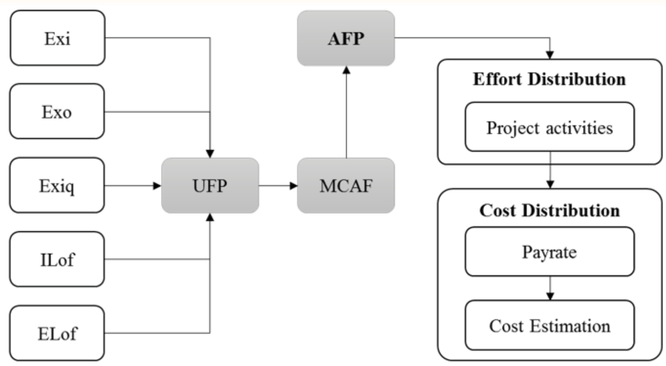
978-1-5386-5721-8/18/$31.00 ©2018 IEEE

According to Dewi [13], the first phase in calculating Unadjusted Function Points (UFP) score is shown in Equation (1).

$$UFP = (Exi*w)+(Exo*w)+(Exiq*w)+(Ilof*w)+(Elof*w) \quad (1)$$

where,
w = weight for each level (simple/average/high)

Next, sum up the weight of each parameter of the Modified Complexity Adjustment Factor (MCAF). There are 16 items of MCAF, varying from 0 to 5 on the interval scale.



Legend:

Exi	: External Input
Exo	: External Output
Exiq	: External Inquiry
Ilof	: Internal Logic File
Elof	: External Logic File
UFP	: Unadjusted Function Points
MCAF	: Modified Complexity Adjustment Factor
AFP	: Adjusted Function Points

Fig. 1. Function Points Method for Software Development [13]

Lastly, multiply UFP and MCAF by the certain constant to become Adjusted Function Points (AFP). Equation (2) below shows the detailed theorem.

$$AFP = UFP*((0.01*TCF)+0.65) \quad (2)$$

A. Game Research

Programmers are collaborating with designers to construct an enjoyable game. That is why, in a game development, designers are asked to add the entertainment value using a comprehensive approach in MDA (Mechanics, Dynamics, and Aesthetics) framework [16]. Hunicke [10] state that LeBlanc detailed each of MDA components as seen in Fig. 2.

MDA is one of the formal approaches often used by researchers to bridge understanding between technical games and game design and development. Shortly, there are 2 perspectives in game development, developer and player itself. Game developers / designers prioritize the technical side, namely mechanics and dynamics in a game. As for players, they prioritize the beauty and its aesthetics in the game. The familiar subjectivity felt by players is certainly very difficult to measure. Because of this, LeBlanc formulated 8 classification to facilitate the emotional measurement of players based on this MDA framework.

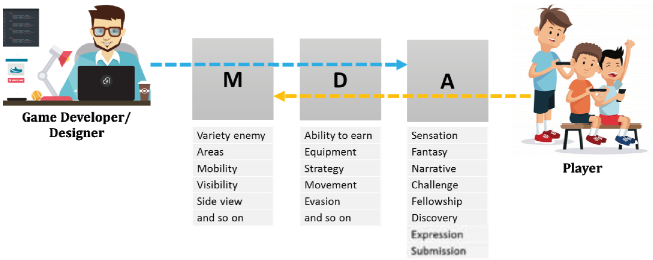


Fig. 2. MDA Framework [10]

The interpretation of Fig. 2:

- Mechanics consist of a variety of enemy unit types; expansive technique and skill; and levels/areas with variable ranges of mobility, environment visibility and side view, and so on. Mechanics in the game are various behaviors, actions, and systematic controls given to players. This mechanic serves to balance and complete the overall game dynamics.
- Dynamics might include the ability to earn or purchase powerful weapons and spy equipment; and also to develop tactics and techniques for stealthy movement, deceptive behavior, evasion, and escape.
- Aesthetics examine role-playing (fantasy element), challenge, and submission. The player will expect coordinated activity on the part of opponents, but probably a lot less emotional expression. If anything, agents should express fear and loathing at the very hint of his presence.

Based on the MDA framework acknowledged in the game design, the Aesthetics component is the beginning of the formation of the 8 Leblanc taxonomies. However, it needs to be understood, this research does not focus on the MDA concept in detail, it's just that the authors pay attention to other aspects of the game that should be considered in making mobile education games that are entertaining and acceptable to game players.

LeBlanc formulated 8 taxonomies—related to aesthetic factor—known as game taxonomy (Table I).

TABLE I. LEBLANC TAXONOMY FOR GAME RESEARCH

Taxonomy	Description
Challenge	things like time pressure and opponent play
Discovery	about rising tension and denouement
Expression	dynamics that encourage individual users to leave their mark
Fantasy	things about making believe
Fellowship	can be encouraged by sharing information across certain members of a session (a team)/ supplying winning conditions that are more difficult to achieve alone

978-1-5386-5721-8/18/$31.00 ©2018 IEEE

Taxonomy	Description
Narrative	flow of game as drama
Sensation	game as sense-pleasure
Submission	milestone to continue the game

On a side note, we are also conducting another research in collaborating LeBlanc's taxonomy to calculate effort of mobile game development [17].

B. *Constraints of Studying Game*

The fact that currently there are still not many researches in reckoning the effort of a mobile game is a challenge. On the other hand, creative business actors are requested to make a quick decision about resources needed to fulfill the needs of customer/client.

We have 2 main assumptions in predicting the effort of mobile casual game phase. Those presumptions are based on researches conducted by IFPUG in their 30 years progress (held in 2017). They are:

- Android platform is built by using Java. Table II shows the average line of code (LOC) index of several programming languages. As seen on the table, LOC index for Java is 53.
- Researches on game productivity are still lacking, that is why we use the IFPUG version of data productivity rate, which is 15.00 (man-hour).

TABLE II. AVERAGE LINES OF CODE (LOC) PER FUNCTION POINTS FOR EACH PROGRAMMING LANGUAGE

No	Programming Language	Average LOC / FP
1	Java	53
2	J2EE	46
3	.NET	57
4	VB.NET	52
5	HTML	34
6	COBOL	61
7	C++	50
8	C#	54
9	ABAP SAP	28
10	ASP	51

Source: http://www.qsm.com/resources/function-point-languages-table

The experimentation of the FP method in case of the casual educational game are rarely found. Meanwhile, the trend of a casual game is growing rapidly. A research in estimating the effort of the digital game was conducted by Sabahat [20], but the complexity factors used in that research are only 8 items, not 14 items. Those factors are number of rules, number of players, animation, 3D visualization, computer opponent, multi-skills, number of type of variants, and miscellaneous game options.

Technical complexity factors of the digital game are totally distinct toward software. For this research, we are using 14 items of difficulty factors with slightly altered definitions to match our case study, namely educational games [17].

III. RESEARCH METHODS

Function Points (FP) method consists of 4 phases. Each phase has detailed iterations. For more details, see the following justification.

1) Identifying parameters

There are 5 main parameters of FP method (see Fig.3)

- External Input (EI), the number of input to the system
- External Output (EO), the number of output from the system interface
- External Inquiries (EQ), the number of demands to a certain input/output
- Internal Logic File (ILF), the number of files needed to make the system work properly
- External Logic File (ELF), the number of files produced by the system (e.g.: files with .pdf, .xls, .doc, or the other extensions)

Each of those parameters will then be classified into 3 categories: simple, average, and complex.

2) Counting Unadjusted Function Points (UFP)

Using Equation (1), we can calculate the UFP score based on 5 parameters multiplied by their own weight (simple/ average/complex).

3) Calculating Technical Complexity Factors (TCF)

As explained in Section 2.B, we will then sum up the score of 14 items of technical complexity factors.

4) Determining Adjusted Function Points (AFP)

In the last phase, we will use Equation (2) to calculate the AFP score. This score will then be converted into effort estimation by multiplying it with productivity rate (15.00) and LOC of Java programming language (53).

IV. RESULT AND DISCUSSION

We did interview CEOs of two Indonesian game studios and got the result shown in Table III. There are 8 educational mobile games in the casual genre. The effort for each one of them will then be estimated using Function Points (FP).

TABLE III. AN OVERVIEW OF EDUCATIONAL GAME

Game ID	Name	Description		
		Feature	*Size*	*Platform*
1	The Expedition Man	A mobile game in which user pretends to be courier who should adhere to traffic regulations while sending package within the given time.	20 MB	Android (Education -Casual)
2	Jermania	A mobile game in which user can learn the culture and conversations used in Germany.	66 MB	Android (Education -Casual)
3	Abata	A serious mobile game in which the user can learn	41 MB	Android (Education

Game ID	Name	Description		
		Feature	*Size*	*Platform*
		to recite the Quran according to its rules of pronunciation (tajwid).		-Serious)
4	Boci Trace Alphabet	A mobile game in which the user is asked to follow icons of plane, ship, and car to form a letter (alphabet).	26 MB	Android (Education -Casual)
5	Boci Theme Park: Carnival	A mobile game in which the user is asked to play in a carnival such as riding ferris wheel, roller coaster, and many more.	42 MB	Android (Education -Casual)
6	Boci Play Counting	A mobile game in which the user is asked to count particular objects (e.g.: fruits)	16 MB	Android (Education -Casual)
7	Boci Zoo	A mobile game in which the user is asked to imitate animal's sound in the zoo	47 MB	Android (Education -Casual)
8	Boci Play Hide and Seek	A mobile game in which the user is asked to find his friend who's hiding somewhere	13 MB	Android (Education -Casual)

We select educational games created for Android devices considering the high number of mobile game players (42%) [1]. But due to the limited data are given by game studio owners (not mentioning the number of programmers, design artists, and game designers involved in the development process), the testing phase of this research is also limited to deviation of kilo line of code (KLOC) and actual size (see Table III).

A. *Identifying Parameters*

The first thing to do is defining 5 main parameters: External Input (EI), External Output (EO), External Inquiry (EQ), Internal Logic File (ILF) and External Logic File (ELF) as shown in Tabel IV. The value of EQ, ILF, and ELF are all zero since no such parameters found in mobile casual games used as case study.

TABLE IV. INDENTIFIED PARAMETERS TO CALCULATE UNADJUSTED FUNCTION POINTS (UFP)

Game ID	EI	EO	EQ	ILF	ELF
1	29	3	0	0	0
2	7	4	0	0	0
3	4	3	0	0	0
4	8	11	0	0	0
5	4	3	0	0	0
6	7	3	0	0	0
7	9	2	0	0	0
8	10	5	0	0	0

B. *Counting Unadjusted Function Points (UFP)*

According to Equation (1), in order to get UFP score, we need to measure each parameters using 3 levels: simple, average, or complex. The result of the UFP calculation for each criterion is shown in Table VI.

C. *Calculating Technical Complexity Factors (TCF)*

The third step is adding the total score of TCF. There are 14 items that need to be added. The delightful thing is, in casual mobile games (based on Dewi previous research [17]), the score of difficulty level are mostly 0, 1, and 2 (range 0 to 5). That is a proof that game complexity is different from software difficulty (see Table V).

TABLE V. TCF BASED ON EIGHT MOBILE-CASUAL GAME

TCF*	Game ID							
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
A	-	-	-	-	-	1	1	1
B	1	1	1	1	1	1	1	1
C	-	-	-	-	-	-	-	-
D	-	-	-	-	-	-	-	-
E	1	1	1	1	1	1	2	1
F	1	1	1	1	1	1	1	1
G	3	1	1	2	2	2	2	1
H	-	-	-	-	-	-	3	-
I	-	-	-	-	-	-	-	-
J	-	-	-	-	-	-	-	-
K	-	-	-	-	-	-	-	-
L	1	1	1	1	1	1	2	1
M	2	2	2	2	2	1	2	2
N	1	1	1	1	1	3	5	3

*TCF scale between 0 until 5

TCF Description

A	: Level of reliability for recovery
B	: Level of data communications
C	: Level of distributed data processing
D	: Level of performance needs
E	: Level of environment configuration
F	: Level of transaction rate (pipe communication)
G	: Level of end-user efficiency
H	: Level of master file update
I	: Level of online real-time update
J	: Level of reusability
K	: Level of installation ease
L	: Level of operational ease
M	: Level of customer variation
N	: Level of change possibility

D. *Determining Adjusted Function Points (AFP)*

The last step is determining the final score or AFP. AFP score will then be converted to Kilo Line of Code (KLOC) and effort (man-hour). Table VI shows the exertion result of Equation (2) explained in Section 2.B.

AFP number (displayed in Table VI) then will be converted to KLOC. We treat Equation (3) to do this.

$$KLOC = AFP*c \tag{3}$$

where,
c = 53, constant for Java programming language based on IFPUG agreement

TABLE VI. CALCULATED UFP, TCF, AND AFP

No	Description	Game ID								Total
		1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	
1	Unadjusted Function Points (UFP)	105	45	30	86	30	39	39	60	434
2	Technical Complexity Factor (TCF)	10	8	8	9	9	11	19	11	85
3	Adjusted Function Points (AFP)	78.75	32.85	21.90	63.64	22.20	29.64	32.76	43.80	325.54

To convert AFP score to the effort (man-hour), we are using Productivity Rate (PR)=15. PR is based on IFPUG agreement in 2017. To get the estimated effort, we follow Equation (4).

$$Effort = AFP*PR \tag{4}$$

where,
PR = 15

TABLE VII. GAME KLOC AND EFFORT ESTIMATION

Game ID	KLOC (Mega Byte)	Effort Estimation (man-hour)
1	4,173.75	1,181.25
2	1,741.05	492.75
3	1,160.70	328.50
4	3,372.92	954.60
5	1,176.60	333.00
6	1,570.92	444.60
7	1,736.28	491.40
8	2,321.40	657.00
Total	**17,253.62**	**4,883.10**

TABLE VIII. THE COMPARISON OF GAME ESTIMATION EFFORT VS. ACTUAL EFFORT

Game ID	Effort Estimation (man-hour)	Actual Effort (man-hour)	Deviation (%)
1	1,181.25	455.85	39
2	492.75	911.70	185
3	328.50	455.85	139
4	954.60	911.70	96
5	333.00	911.70	274
6	444.60	2,279.25	513
7	491.40	7,293.60	1,484
8	657.00	1,823.40	278

Implementation result of Equation (3) and (4) are summarized in Table VII. Effort estimation from that calculation will then be compared to actual effort. Deviation exists as the result of comparison between effort estimation and actual effort is shown in Table VIII. Deviation average for those 8 mobile casual games is as many as 376%.

V. CONTRIBUTION AND LIMITATION

The theoretical contribution of this research is to suggest that business players in the field of mobile games are more careful if they want to use the FP method. This is due to the magnitude of the deviation between the actual effort and the estimation effort which almost reaches 4 times as much as the case in this study.

The limitation in this study is that the sample cases used are still casual game scales and the number is too small, so it is not comprehensive to justify whether the FP method can be applied in the case of mobile games especially on medium to large scale.

VI. CONCLUSION AND FUTURE WORK

From this research, we can conclude that:

- There are 4 phases in FP method: (i) identifying parameters; (ii) counting Unadjusted Function Points (UFP); (iii) calculating Technical Complexity Factors (TCF); and (iv) determining Adjusted Function Points (AFP).
- Deviation average between effort estimated by using FP and actual effort for each mobile-casual game is 376%.

Based on the results above, the application of the FP method is not entirely good if implemented in game cases. So that in the future, business people in the field of mobile games, must consider the Leblanc taxonomy to fit the actual effort calculation, and reduce its deviation to effort estimation using the FP method. Therefore, in the future, this research will be continued with the modification of the FP method by considering the eight LeBlanc taxonomies and re-testing with more samples of mobile casual games.

ACKNOWLEDGMENT

We would like to thank Kementerian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia—Ministry of Research, Technology, and Higher Educations of Indonesia—for facilitating the research funding. We would also like to thank Lembaga Penelitian dan Pengabdian Kepada Masyarakat (LPPM) Universitas Internasional Semen Indonesia (UISI)—Reseach and Community Service Department of UISI—, LPPM Institut Teknologi Sepuluh Nopember (ITS), and CEOs of Nightspade.com and RollingGlory.com for the contributions given on this research.

REFERENCES

[1] E. McDonald, "The global games market will reach $108.9 billion in 2017 with mobile taking 42%," *NewZoo*. pp. 1–5, 2017.

[2] Entertainment Software Association, "2014 Essential Facts About the Computer and Video Game Industry," *Soc. Sci. Comput. Rev.*, 2014.

[3] Entertainment Software Association, "Essential facts 2017," *Soc. Sci. Comput. Rev.*, 2017.

[4] A. J. Albrecht, "Measuring application development productivity," in *IBO Conference on Application Development*, 1979, pp. 83–92.

[5] A. J. Albrecht and J. E. Gaffney Jr, "Software function, source lines of code and envelopment effort prediction: a software science validation," in *Software engineering metrics I*, 1993, pp. 137–154.

[6] C. J. Lokan, "Function Points," in *Advances in Computers*, vol. 65, 2005, pp. 297–347.

[7] M. Aguiar, "Function Points or Use Case Points?," *Crosstalk*, 2006.

[8] G. Karner, "Resource estimation for objectory projects," *Object. Syst. SF AB*, pp. 1–9, 1993.

[9] R. S. Dewi, G. F. Prassida, and A. P. Subriadi, "UCPabc as an integration model for software cost estimation," in *Science in Information Technology (ICSITech), 2016 2nd International Conference on*, 2016, pp. 187–192.

[10] R. Hunicke, M. LeBlanc, and R. Zubek, "MDA: A Formal Approach to Game Design and Game Research," *Work. Challenges Game AI*, pp. 1–4, 2004.

[11] Ifpug, *Function Point Counting Practices Manual*, vol. on06/23/. 2010.

[12] 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.

[13] 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.

[14] B. Boehm, R. Valerdi, J. A. Lane, and W. a. Brown, "COCOMO Suite Methodology and Evolution," *Softw. Eng. Technol.*, vol. 18, no. 4, pp. 20–25, 2005.

[15] B. Boehm *et al.*, "COCOMO II Model Definition Manual," *Univ. South. Calif.*, vol. 4, no. 1, pp. 6–6, 2000.

[16] M. LeBlanc, "Mechanics, Dynamics, Aesthetics: A Formal Approach to Game Design," *Lect. Northwest. Univ.*, 2004.

[17] R. S. Dewi and T. Andari, "Game Complexity Factor: A Collaborative Study of {LeBlanc} Taxonomy and Function Points Method," in *2018 International Conference on Electrical Engineering and Computer Science (ICECOS) (ICECOS 2018)*, 2018.

[18] B. Jeng, D. Yeh, D. Wang, S.-L. Chu, and C.-M. Chen, "A Specific Effort Estimation Method Using Function Point," *J. Inf. Sci. Eng.*, vol. 27, pp. 1363–1376, 2011.

[19] T. Wahyono, B. Soewito, S. W. H. L. Hendric, and F. L. Gaol, "Software complexity measurement of water poverty mapping application with function point method," in *Applied Computer and Communication Technologies (ComCom), 2017 International Conference on*, 2017, pp. 1–5.

[20] N. Sabahat, A. A. Malik, and F. Azam, "A Size Estimation Model for Board-Based Desktop Games," *IEEE Access*, vol. 5, pp. 4980–4990, 2017.

978-1-5386-5721-8/18/$31.00 ©2018 IEEE