Check 27. Measurement of QuestDone mobile application.pdf

by Arta Sundjaja

Submission date: 17-May-2019 02:56PM (UTC+0700)
Submission ID: 1102514169
File name: 27._Measurement_of_QuestDone_mobile_application.pdf (436.18K)
Word count: 4095
Character count: 21099

Measurement of QuestDone Mobile Application Using 7 Steps Use Case Points Method

Rhio Sutoyo[1,2]
[1]Computer Science Department, BINUS Graduate Program - Doctor of Computer Science
[2]Computer Science Department, School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
rsutoyo@binus.edu

Harco Leslie Hendric Spits Warnars[1], Ford Lumban Gaol[2], Edi Abdurachman[3]
Computer Science Department, BINUS Graduate Program - Doctor of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
spits.hendric@binus.ac.id[1], fgaol@binus.edu[2], edia@binus.edu[3]

Benfano Soewito
Computer Science Department, BINUS Graduate Program - Master of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
bsoewito@binus.edu

***Abstract*—The rise of mobile application is inevitable. Every year, the number of mobile application is increased. It is important for mobile application project owners to calculate the required resources before building a mobile application. In software metric, Use Case Points method is able to count software size of mobile application based on their functionality. This method utilizes use case diagram as their computation factors in the estimation process. Moreover, two other complexity factors are also considered in this method, which are: Technical Complexity Factor and Environment Factor. In this paper, we present software size calculation of QuestDone Mobile Application using 7 steps use case points method. QuestDone has been implemented, but we do not know its software size (i.e. how big the software, how much it cost, how many people is needed). As the result from use case points method, the Use Case Points value of QuestDone is 126.88 with Effort Estimation equal to 889 hours. The software size estimation process of QuestDone Mobile Application detailed in this paper can give an insight to project owners to count software size of other similar projects.**

Keywords—software size; software metrics; use case points; mobile application

I. INTRODUCTION

Mobile application creation is increasing over the years. There are currently more than 2,000,000 of iOS apps that can be found on the market [1], [2]. Moreover, there are more than 3,000,000 of Android mobile apps [3]. Since people and the market start leaving cell phones for smart phones [4], the rise of mobile application is inevitable. Furthermore, mobile application has brought global impact into the world [5]. For individuals, people become easily communicates (e.g. messages, audio calls, video calls, etc.) and can browse the web in the palm of their hand. Moreover, the number of new gamers is also increased because of the rise of smartphone games, which has the biggest number of total apps over all categories [6], [7]. For business and corporates, the rise of mobile market revolutionizes application purchase, music services, movie services, multimedia consumption, ads personalization, and many other fields [5].

QuestDone Mobile Application is one example of Android mobile application [8]. In this application, users can create party with other users and complete quests together. A completed quest gives points for the users. Then, the users will be able to exchange those points for rewards. QuestDone is developed by using Java language and designed with object-oriented concept. QuestDone application has been implemented. Nevertheless, such software size measurement method was never done before to this application. This research chose QuestDone mobile application because it is quite a general mobile application with simple tasks and medium-size use case design. Moreover, it is closely related with our previous research [8].

With the popularity of mobile application creation, it is very important for mobile application project manager or project owner to calculate the required resources of their application before building it [9]. Thus, they could anticipate and have better preparation for the projects. The calculation of software size on QuestDone mobile application using 7 steps use case points method detailed in this paper will gives an insight about software size estimation process [10], [11]. Furthermore, other similar mobile application projects can also use our result for their references [12]. Based on our finding, there is no other research that focus on calculate the cost of mobile application using use case points method.

This paper will be structured into five sections. The first section discusses the rise of smartphone creation and the importance of software size measurement. The second section explains about Use Case Points (UCP) and the seven steps to calculate UCP and Effort Estimation. QuestDone mobile application, UCP calculation, and effort estimation will be presented in section three. Section four details the analysis of software size measurement from the previous section. Lastly, conclusions of this paper are written in section five.

II. USE CASE POINTS SOFTWARE SIZE MEASUREMENT

Use Case Points (UCP) was introduced by Karner [11] in 1993. Use Case Points works similarly with Function Point (FP) by Albrecht [13] where the goal of the method is to estimate software size. Nevertheless, UCP utilities use case diagram as its factors and can be used only on object-oriented system. The Use Case Points is different from Object Points (OP) [14], because OP includes non-UML factors into its effort calculation [15], [16]. There is also other object oriented metrics which utilizes PHP_depend [17].

The calculation of Use Case Points is divided into 7 steps [11], [18]–[20], as shown in the next equations and they are:

1. Unadjusted Use Case Weights (UUCW)

In this step, each use case is labeled as simple, average, or complex based on their transactions. A use case is considered simple if it has three or less transactions including alternative courses. An average uses case is a use case with three to seven transactions including alternative courses. And a complex use case is a use case that have more than seven transactions including alternative courses. Their weights are 5, 10, and 15, respectively [11]. The total of UUCW is the total sum of multiplication result between each use case and its weight. The equation to calculate UUCW value can be seen in equation (1).

$$UUCW = \sum(\#Use\ Cases * Weight\ Factor) \quad (1)$$

2. Unadjusted Actor Weight (UAW)

In this step, each actor is marked as simple, average, or complex based on their interaction with the system. A simple actor is an actor that represent another system with a defined Application Programming Interface (API). Moreover, an actor is considered average if it has an interaction with another system by a protocol (e.g. Hyper Text Transfer Protocol, File Transfer Protocol, etc.). And an actor is considered complex if it interacts with the system with a Graphical User Interface (GUI). Their weights are 1, 2, and 3, respectively [11]. The total of UAW is the total sum of multiplication result between each actor and its weight. Equation (2) shown below is the equation to count UAW value.

$$UAW = \sum(\#Actors * Weight\ Factor) \quad (2)$$

3. Unadjusted Use Case Points (UUCP)

In this step, the Unadjusted Use Case Points (UUCP) can be calculated by adding the result of Unadjusted Use Case Weights (UUCW) and Unadjusted Actor Weight (UAW). UUCP value can be calculated by using equation (3).

$$UUCP = UUCW + UAW \quad (3)$$

4. Technical Complexity Factor (TCF)

In this step, difficulty of system construction also be included to the process. Thus, we need to count Technical Complexity Factor (TCF). It is divided into 13 complexity factors and each factor has its own weight [11]. All factors and its weights are listed in Table I.

TABLE I. TECHNICAL COMPLEXITY FACTOR (TCF)

Factor	Description	Weight
T1	Distributed System	2.0
T2	Response Time/Performance Objectives	1.0
T3	End-user Efficiency	1.0
T4	Internal Processing Complexity	1.0
T5	Code Reusability	1.0
T6	Easy Installation Process	0.5
T7	Easy to Use	0.5
T8	Portability to Other Platforms	2.0
T9	Easy to Change	1.0
T10	Concurrent/Parallel Processing	1.0
T11	Security Features	1.0
T12	Access for Third Parties	1.0
T13	End-user Training	1.0

The weighting value of each technical complexity factor can be filled from 0 to 5 numbers [11]. The scoring guide depends on the importance of those factors to the system. Technical Factor (TF) can be calculated by combining all values from multiplication between each weighting value and its weight. Then, the value will be used to count Technical Complexity Factor (TCF). The equation to calculate TCF value can be seen in equation (4), suggested by Albrecht [11].

$$TCF = 0.6 + (0.01 * Technical\ Factor) \quad (4)$$

5. Environmental Factor (EF)

In this step, another complexity factor that affect the software size also be included. It is quite similar with TCF, but Environmental Factor (EF) has its own properties and weights. The detail can be seen in Table II.

TABLE II. ENVIRONMENTAL FACTOR (EF)

Factor	Description	Weight
E1	Familiarity with Development Process Used	1.5
E2	Application Experience	0.5
E3	Object-Oriented Experience of Team	1.0
E4	Lead Analyst Capability	0.5
E5	Motivation of the Team	1.0
E6	Stability of Requirements	2.0
E7	Part-Time Staff	-1.0
E8	Difficult Programming Language	-1.0

The weighting value of each environmental factor can be filled from 0 to 5 numbers [11]. EFactor can be calculated by combining all values from multiplication between weighting value and its weight. Then, the value will be used to count Environmental Factor (EF). EF value can be computed by using equation (5). The constants was obtained based on interviews with users at Objective Systems [11].

$$EF = 1.4 + (-0.03 * EFactor) \quad (5)$$

6. Use Case Points (UCP)

In this step, the value of Use Case Points (UCP) will be calculated. It is the result of multiplication between Unadjusted Use Case Points (UUCP), Technical Complexity Factor (TCF), and Environmental Factor (EF). The equation (6) presented below is the equation to count UCP value.

$$UCP = UUCP * TCF * EF \quad (6)$$

7. Effort Estimation (E)

In this step, we measure the value of Effort Estimation (E). It is the result of multiplication of Use Case Points (UCP) and the value of Person Hour per UCP (PH per UCP). Effort Estimation value can be calculated by using equation (7).

$$E = UCP * PH\ per\ UCP \quad (7)$$

III. Measurement of QuestDone Mobile Application Using 7 Steps Use Case Points Method

QuestDone is an Android mobile application designed for users to play and socialize together by utilizing Global Positioning System (GPS) inside the smartphone [8]. This application encourages users to see various places, complete task, collect gift / stamp, and socialize with other players. Several screenshots of the application can be seen in Fig 1. The figures shown are main menu, quest list, friend list, and view shop.

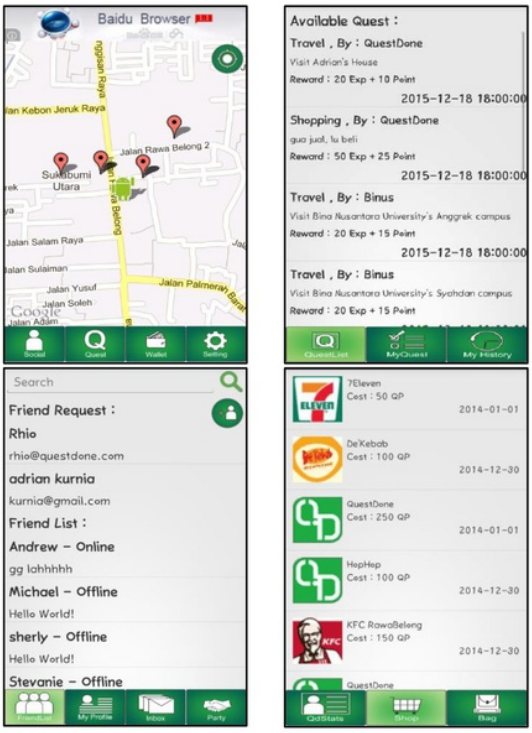

Fig. 1. Screenshots of QuestDone Mobile Application [8]

The use case of QuestDone can be seen in Fig 2 and there are two actors: users and members. Moreover, there are 19 use cases where 13 of the use cases are extended use cases.

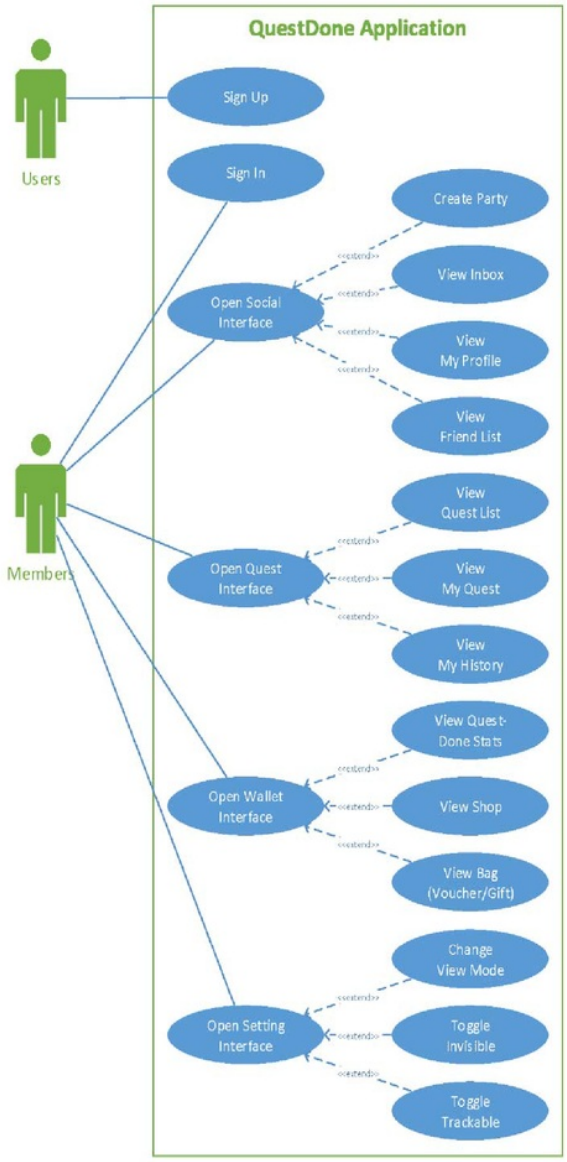

Fig. 2. Use Case Diagram of QuestDone Mobile Application

The use case design in Fig 2 is built by using the concept from Whitten and Bentley [21]. Here is the explanation of each use case:

1. Sign Up: New users will be able to register his/her new account with this function.

2. Sign In: After member registration has been complete, users can sign in and start using QuestDone application with this menu.
3. Open Social Interface: The system will open social menu interface for the users.
4. Create Party: Users will be able to create party with their existing friends. The party can complete quests together.
5. View Inbox: Users can check messages from friends in this menu.
6. View My Profile: Profile of the users can be seen in this menu.
7. View Friend List: In here, the users will be able to see all their existing friends from QuestDone.
8. Open Quest Interface: With this function, users will be able to see main menu of quest.
9. View Quest List: With this function, users can see all available quest and choose which one to complete.
10. View My Quest: After the quest has been taken by the users, those quest will be listed in this menu.
11. View My History: All finished quest will be listed in the "My History" menu. Thus, users can track all their achievements.
12. Open Wallet Interface: In this menu, users will be able to track their personal progress of the game and rewards.
13. View QuestDone Stats: Users can track their level, total point, and total quest that has been done in the game.
14. View Shop: In here, the users will be able to see all available shops that provides exchange gifts for in-game points.
15. View Bag (Voucher / Gift): All exchanged gifts (i.e. vouchers) are listed by opening this menu.
16. Open Setting Interface: In this menu, users will be able to open interface for several settings to the game.
17. Change View Mode: Users will be able to change the view in the game map into "Street View" or "Satellite View" within this menu.
18. Toggle Invisible: If the users want to be invisible to other players, they can choose this menu.
19. Toggle Trackable: If the users do not want to be trackable, they can change it by using this toggle.

As mention before, QuestDone mobile application has never been measured by any software size method. In order to count its cost and software size, 7 steps use case method is used for the estimation process.

The first step is to count the Unadjusted Use Case Weight (UUCW) value as shown in Table III. The weight is filled with 5, 10, and 15 based on their category: simple, average, and complex, respectively. Based on equation (1), the UUCW score for Table III is 120.

TABLE III. UNADJUSTED USE CASE WEIGHT (UUCW)

Use Case	Category	Weight
Sign Up	Simple	5
Sign In	Simple	5
Open Social Interface	Average	10
Create Party	Complex	15
View Inbox	Simple	5
View My Profile	Simple	5
View Friend List	Average	10
Open Quest Interface	Simple	5
View Quest List	Simple	5
View My Quest	Simple	5
View My History	Simple	5
Open Wallet Interface	Simple	5
View QuestDone Stats	Simple	5
View Shop	Average	10
View Bag(Voucher/Gift)	Simple	5
Open Setting Interface	Simple	5
Change View Mode	Simple	5
Toggle Invisible	Simple	5
Toggle Trackable	Simple	5
Unadjusted Use Case Weight (UUCW)		120

The second step is to calculate the Unadjusted Actor Weight (UAW). Because all actors in the QuestDone apps are using graphical user interface, they belong to the complex category. Based on equation (2), the UAW score for Table IV is 6.

TABLE IV. UNADJUSTED ACTOR WEIGHT (UAW)

Category	Weight	Actors	Count	Weight * Count
Simple	1	-	0	0
Average	2	-	0	0
Complex	3	Users, Members	2	6
Unadjusted Actor Weight (UAW)				6

The third step, Unadjusted Use Case Points (UUCP) is counted by adding Unadjusted Use Case Weight (UUCW) and Unadjusted Actor Weight (UAW). Thus, the value of UUCP is 120 + 6 = 126 based on equation (3).

The fourth step is to count the Technical Complexity Factor (TCF) value. There are 13 factors that affect the result of this value. The details of this calculation (i.e. Technical Factor) can be seen in Table V.

TABLE V. TECHNICAL COMPLEXITY FACTOR (TCF)

Factor	Description	Weight	Value	Weight * Value
T1	Distributed System	2.0	1	2.0
T2	Response Time/Performance Objectives	1.0	5	5.0
T3	End-user Efficiency	1.0	4	4.0
T4	Internal Processing Complexity	1.0	1	1.0
T5	Code Reusability	1.0	1	1.0
T6	Easy Installation Process	0.5	5	2.5
T7	Easy to Use	0.5	5	2.5
T8	Portability to Other Platforms	2.0	2	4.0
T9	Easy to Change	1.0	3	3.0
T10	Concurrent/Parallel Processing	1.0	1	1.0
T11	Security Features	1.0	3	3.0
T12	Access for Third Parties	1.0	5	5.0
T13	End-user Training	1.0	1	1.0
			Total of Technical Factor	35.0

Based on equation (4), the value of Technical Complexity Factor (TCF) value of QuestDone mobile application is 0.6 + (0.01 * 35) = 0.95.

The fifth step is count the Environmental Factor (EF) value of this application. The detail calculation for EFactor can be seen in Table VI below.

TABLE VI. ENVIRONMENTAL FACTOR (EF)

Factor	Description	Weight	Value	Weight * Value
E1	Familiarity with Development Process Used	1.5	2	3.0
E2	Application Experience	0.5	2	1.0
E3	Object-Oriented Experience of Team	1.0	3	3.0
E4	Lead Analyst Capability	0.5	5	2.5
E5	Motivation of the Team	1.0	3	3.0
E6	Stability of Requirements	2.0	3	6.0
E7	Part-Time Staff	-1.0	2	-2.0
E8	Difficult Programming Language	-1.0	5	-5.0
			Total of EFactor	11.5

Based on equation (5), the value of Environmental Factor (EF) value of QuestDone mobile application is 1.4 + (-0.03 * 11.5) = 1.06.

The sixth step is to count the Use Case Points (UCP) value by multiplying the value of UUCP, TCF, and EF. Hence, the value of UCP is 126 * 0.95 * 1.06 = 126.88 based on equation (6).

The seventh step is to count the Effort Estimation (E) value by multiplying the value of UCP with the value of Person Hours per Use Case Points (PH per UCP).

TABLE VII. LIST OF SOFTWARE COMPLEXITY

Category	Person Hours per Use Case Points
Simple / Low	1 – 20
Complex	21 – 40
Very Complex / High	more than 40

Based on our interview with the lead developer of QuestDone mobile application, the PH per UCP value is 7 hours. Therefore, QuestDone mobile application is categorized as simple/low project. The category can be found in the Table VII [19], [22]. Based on equation (7), the estimation hours to build QuestDone mobile application is approximately 126.88 * 7 = 889 hours.

IV. RESEARCH RESULT AND DISCUSSION

After all pre-requested calculations have been done, the software size of QuestDone mobile application can be determined by looking at the use case points. Based on Table VIII [19], [20], [22], the software size of this mobile application is medium. It is because the UCP of QuestDone mobile application is 126.88 and it belongs to the range 100 – 299 (medium size software).

TABLE VIII. SOFTWARE SIZE CATEGORY

Software Size Category	Use Case Points
Small	less than 100
Medium	100 – 299
Large	300 – 799
Extreme	more than 799

In order to count the cost of this project, we need to convert the effort estimation from hourly into monthly. Generally, software developer works from nine to five daily or 8 hours per day. Moreover, there are 5 working days in a week. And there are 4 weeks in a month. Hence, the effort estimation (days) is 889 ÷ 8 = 111.13 days. The effort estimation (weeks) is 111.13 ÷ 5 = 22.23 weeks. And, the effort estimation (months) is 22.23 ÷ 4 = 5.56 months.

Based on the survey conducted by JobPlanet [23], [24], the average monthly developer (i.e. software engineers) salary in Indonesia is IDR 4,050,000 or USD 303 (1 USD = IDR 13,342). Hence, the cost of employee wage in this application is far cheaper than the project from Iskandar et al. [19]. This is because our application can be built by using fresh graduate of Android

developers. On the other hand, the project of Knowledge Management Portal from Iskandar requires experienced Microsoft SharePoint software engineers. In short, the project value of QuestDone Mobile Application is 5.56 × 4,050,000 = IDR 22,502,813. It is approximately equivalent to 1,686 USD (1 USD = 13,342). The summary of this information can be seen in Table IX.

TABLE IX. SUMMARY OF QUESTDONE SOFTWARE SIZE MEASUREMENT USING USE CASE POINTS ESTIMATION TECHNIQUE

Measurement Items	QuestDone Mobile Application
Unadjusted Use Case Weight (UUCW)	120
Unadjusted Actor Weight (UAW)	6
Unadjusted Use Case Points (UUCP)	126
Technical Complexity Factor (TCF)	0.95
Environmental Factor (EF)	1.06
Use Case Points (UCP)	126.88
Software Size	Medium
Software Complexity PH per UCP	7
Effort Estimation (hours)	889
Effort Estimation (days)	111.13
Effort Estimation (weeks)	22.23
Effort Estimation (months)	5.56
Average Monthly Developer Salary in Indonesia (IDR)	4,050,000
Average Monthly Developer Salary in Indonesia (IDR)	303
Project Value (IDR)	22,502,813
Project Value (USD)	1,686

V. CONCLUSIONS

In this paper, we have presented software size calculation of QuestDone Mobile Application with Use Case Points (UCP). UCP software size metrics measures the functionality of software by looking at its use case diagram. Based on our calculation, QuestDone is a medium size software with 126.88 total point of UCP. Moreover, the effort estimation of this application is 889 hours or equivalent to 5.56 months. With average monthly salary equal to IDR 4,050,000 (USD 303), the cost to build this application is IDR 22,502,813 (USD 1,686).

Based on those estimated values, project owners can analyze the required resources of projects. Thus, they can spend their resources (i.e. time, money, people, etc.) more efficiently. Moreover, we hope this paper can give an insight to other similar mobile application projects about duration of the project and cost to build the software.

REFERENCES

[1] S. Costello, "How Many Apps Are in the App Store?" [Online]. Available: https://www.lifewire.com/how-many-apps-in-app-store-2000252. [Accessed: 29-Aug-2017].

[2] Statista, "Number of available apps in the Apple App Store from July 2008 to January 2017." [Online]. Available: https://www.statista.com/statistics/263795/number-of-available-apps-in-the-apple-app-store/. [Accessed: 29-Aug-2017].

[3] Statista, "Number of available applications in the Google Play Store from December 2009 to June 2017." [Online]. Available: https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/. [Accessed: 29-Aug-2017].

[4] P. Zheng and L. M. Ni, "Spotlight: the rise of the smart phone," *IEEE Distrib. Syst. Online*, vol. 7, no. 3, p. 3, 2006.

[5] R. Islam, R. Islam, and T. Mazumder, "Mobile application and its global impact," *Int. J. Eng. Technol.*, vol. 10, no. 6, pp. 72–78, 2010.

[6] A. Meola, "Mobile gaming is about to become the undisputed king of the jungle." [Online]. Available: http://www.businessinsider.com/mobile-gaming-will-surpass-legacy-gaming-in-2016-2016-4/?IR=T. [Accessed: 29-Aug-2017].

[7] L. Hjorth and I. Richardson, "Games of Being Mobile: The Unruly Rise of Mobile Gaming in Japan," in *Mobile Gaming in Asia*, Springer, 2017, pp. 21–33.

[8] R. Sutoyo, M. Yoshep, A. Susanto, and A. Kurniadinata, "Questdone Application With Social Networking Features As the Actual World Interaction Media on Android Smartphone," *CommIT (Communication Inf. Technol. J.*, vol. 6, no. 2, pp. 53–62, 2012.

[9] R. K. Clemmons, "Project Estimation With Use Case Points," *J. Def. Softw. Eng.*, pp. 18–22, 2006.

[10] A. B. Nassif, L. F. Capretz, and D. Ho, "Calibrating use case points," in *Companion Proceedings of the 36th International Conference on Software Engineering - ICSE Companion 2014*, 2014, pp. 612–613.

[11] G. Karner, "Resource estimation for objectory projects," *Object. Syst. SF AB*, vol. 17, 1993.

[12] B. Anda, H. C. Benestad, and S. E. Hove, "A Multiple-Case Study of Software Effort Estimation based on Use Case Points," in *International Symposium on Empirical Software Engineering*, 2005, pp. 407–416.

[13] A. J. Albrecht, "Measuring application development productivity," in *Proc. of the Joint SHARE/GUIDE/IBM Applicaiton Development Symposium*, 1979, pp. 83–92.

[14] L. Wu, "The comparison of the software cost estimating methods," *Univ. Calgary*, pp. 2–12, 1997.

[15] G. Levesque, V. Bevo, and D. T. Cao, "Estimating software size with UML models," in *Proceedings of the 2008 C3S2E conference on - C3S2E '08*, 2008, p. 81.

[16] L. Lavazza and G. Robiolo, "Introducing the Evaluation of Complexity in Functional Size Measurement: a UML-based Approach," in *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement - ESEM '10*, 2010, p. 1.

[17] S. Sarwo, F. L. Gaol, B. Soewito, and H. L. H. S. Warnars, "Object Oriented Metrics to measure the quality of software study case Request online System," pp. 2–6, 2016.

[18] G. B. Ibarra and P. Vilain, "Software Estimation Based on Use Case Size," in *2010 Brazilian Symposium on Software Engineering*, 2010, pp. 178–187.

[19] K. Iskandar, F. L. Gaol, B. Soewito, H. L. H. S. Warnars, and R. Kosala, "Software size measurement of knowledge management portal with use case point," *Proceeding - 2016 Int. Conf. Comput. Control. Informatics its Appl. Recent Prog. Comput. Control. Informatics Data Sci. IC3INA 2016*, pp. 42–47, 2017.

[20] D. Kurniadi, S. W. Harco Leslie Hendric, F. L. Gaol, and B. Soewito, "Software Size Measurement of Student Information Terminal with Use Case Point," *IEEE Int. Conf. Appl. Comput. Commun. Technol.*, 2017.

[21] J. L. Whitten and L. D. Bentley, *System Analysis & Design Methods*, 7th Editio. 2007.

[22] IT_BINUS_Department, "BINUS IT Software Development Documentation." Jakarta, 2016.

[23] R. K. Nistanto, "Ini Dia, Daftar Gaji Pekerja TI di Indonesia," 2016. [Online]. Available: http://tekno.kompas.com/read/2016/04/14/09062057/Ini.Dia.Daftar.Gaji.Pekerja.TI.di.Indonesia. [Accessed: 23-Aug-2017].

[24] R. A. Arjanti, "Ragam Profesi TI dengan Gaji Rata-rata Paling Tinggi di Indonesia," 2016. [Online]. Available: http://blog.id.jobplanet.com/ragam-profesi-ti-dengan-gaji-rata-rata-paling-tinggi-di-indonesia/. [Accessed: 23-Aug-2017].

Check 27. Measurement of QuestDone mobile application.pdf

ORIGINALITY REPORT

17%	9%	15%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Bilkent University Student Paper	3%
2	Dede Kurniadi, Sasmoko, Harco Leslie Hendric Spits Warnars, Ford Lumban Gaol. "Software size measurement of student information terminal with use case point", 2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom), 2017 Publication	2%
3	Submitted to University of Minnesota System Student Paper	2%
4	Hady Pranoto, Cuk Tho, Harco Leslie Hendric Spits Warnars, Edi Abdurachman, Ford Lumban Gaol, Benfano Soewito. "Usability testing method in augmented reality application", 2017 International Conference on Information Management and Technology (ICIMTech), 2017 Publication	2%

5	en.wikipedia.org Internet Source	2%
6	Tomas Urbanek, Zdenka Prokopova, Radek Silhavy, Veronika Vesela. "Prediction accuracy measurements as a fitness function for software effort estimation", SpringerPlus, 2015 Publication	2%
7	Submitted to Northcentral Student Paper	1%
8	Herman Bedi Agtriadi, Natalia Chandra, Harco Leslie Hendric Spits Warnars, Ford Lumban Gaol. "Software size measurement with use case point for employee application software at STT-PLN", 2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom), 2017 Publication	1%
9	Didik Haryadi, Harisno, Victory Haris Kusumawardhana, Harco Leslie Hendric Spits Warnars. "The Implementation of E-money in Mobile Phone: A Case Study at PT Bank KEB Hana", 2018 Indonesian Association for Pattern Recognition International Conference (INAPR), 2018 Publication	1%
10	Huanzhen Dong. "Monitoring Software Projects	1%

with Earned Value Analysis and Use Case Point", Seventh IEEE/ACIS International Conference on Computer and Information Science (icis 2008), 05/2008
Publication

11	Submitted to University of Hertfordshire Student Paper	1%
12	Karto Iskandar, Ford Lumban Gaol, Benfano Soewito, Harco Leslie Hendric Spits Warnars, Raymondus Kosala. "Software size measurement of knowledge management portal with use case point", 2016 International Conference on Computer, Control, Informatics and its Applications (IC3INA), 2016 Publication	1%

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		