E-Voting Software Quality Analysis with McCall's Method

1st Fransiskus Panca Juniawan, 2nd Dwi Yuny Sylfania, 3rd Laurentinus, 4th Rahmat Sulaiman, 5th Rendy Rian Chrisna Putra, 6th Harrizki Arie Pradana, 7th Tri Sugihartono, 8th Hengki
[1,2,3,4,5,6,7]*Informatics Engineering Department, Information Technology Faculty*
[8]*Digital Business Department, Business Economy Faculty*
ISB Atma Luhur
Pangkalpinang, Indonesia
[1]fransiskus.pj@atmaluhur.ac.id, [2]dysylfania@atmaluhur.ac.id, [3]laurentinus@atmaluhur.ac.id, [4]rahmatsulaiman@atmaluhur.ac.id, [5]rendyriancp@atmaluhur.ac.id, [6]harrizkiariep@atmaluhur.ac.id, [7]trisugihartono@atmaluhur.ac.id, [8]hengki@atmaluhur.ac.id

***Abstract*—Nowadays, software testing is important, especially to determine its quality. This study aims to test the quality of e-voting software using the McCall method framework that focuses on software Product Operation aspects. By knowing the current quality of e-voting software, we know how much positive impact the existing software has, besides that the lowest metric can be improved in the future. There are five metrics tested, namely, Correctness, Reliability, Efficiency, Integrity, and Usability. The test results show that the highest metric value is correctness with 89.85%, and the lowest metric is the efficiency with 48.3%. Average, the e-voting software is considered to be of good quality with an average score of 79,932%.**

Keywords—e-Voting, McCall, Software Quality Testing

I. INTRODUCTION

In building software, there are many aspects that must be considered. One of them is the quality of the software [1]. This is important because, in software development, quality is used as a guarantee in every software life cycle [2]. There are many methods that can be used in testing the quality of the software. One of them is the McCall method. This research will test the quality of the android-based e-voting software which has been applied to the ISB Atma Luhur for the election of BEM Chairman [3]. The purpose of testing software quality is to find out whether the implementation of the software is in accordance with the needs [1]. Currently, the e-voting software has never been tested for quality. For this reason, testing the quality of this software is important to do. McCall's method was chosen because it has 3 choices of important aspects, namely Product Operation, Product Revision, and Product Transition, which can be adjusted to the software to be tested [4]. In addition, this method has the most complete and in-depth criteria [5].

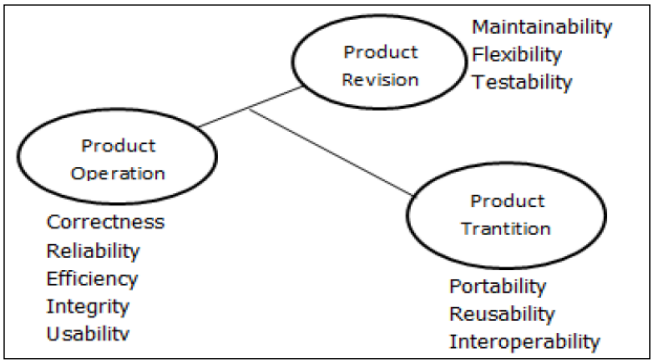

Fig. 1. McCall's Software Quality Factors

This research will focus on testing the product operation, which consists of aspects of Correctness, Reliability, Efficiency, Integrity, and Usability because it refers to the operation of the software by each user. Each of these aspects also has its own criteria as shown in Figure 2 [6].

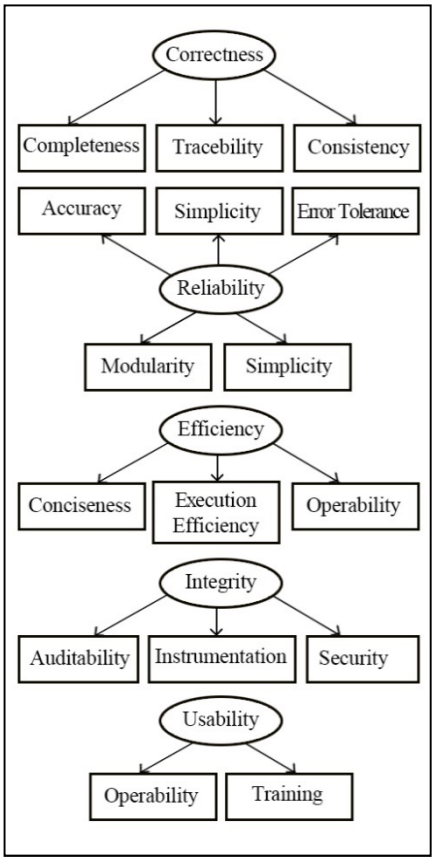

Fig. 2. Criteria Relation of Product Operation

Several previous studies conducted web-based software testing using the McCall method. One of them is by integrating operability and training metrics in the Usability aspect of the software [6]. Furthermore, there is software testing that focuses on Product Operations with its five metrics. The results of this study are quite good, with a value of 55.22% [5] and had good value, with an average of 77.1% [7]. Another research tested the academic information system software [8] and produced good values, but contrary to that, other studies have deficiencies in the correctness, efficiency, and integrity metrics [2]. The Usability aspect is the focus of testing by prioritizing the operability and training metrics on the bonus system software and having good results [9].

McCall's method is also used to test the BAQ (Business Activity Queries) module on an ERP system called Epicor. The results of testing are the correctness, and usability metrics have a good value [10]. In addition, the McCall method can also be combined with white box, black box, alpha testing, and beta testing in conducting the test which has excellent test results [4]. The McCall method can also be combined with the AHP and SMART methods in the decision support system. This study consists of 12 criteria to determine the best lecturer. The result is good, with a value of 78.2% [11].

II. METHODOLOGY

This research was conducted with quantitative methods by prioritizing the objectivity of the data in the form of numbers/scores/values or statements that are assessed and analyzed by statistical analysis [9]. In the process, this research was carried out in several stages, namely:

A. Data Collecting

At this stage, we collect data that will be used to support research. Data are also obtained from literature studies that collect theory testing and previous research. Data is also obtained from books or scientific articles from journals related to the theme taken. Furthermore, information is collected from the software that would be tested. The object of research is the Android-based e-voting software for BEM Chairman, which is used by ISB Atma Luhur students in selecting their BEM chairman every year.

B. Population

The population of this research is 763 active students from semester 2 to 6 at ISB Atma Luhur. Samples taken from this study is using simple random sampling, amounting to 60 people.

C. Questionnaire Analysis

The next stage is to analyze the results of the questionnaire to get the quality value of the software. Tests are carried out using the McCall Method, which focuses on Product Operations, which consists of five criteria. Namely, Correctness, Reliability, Efficiency, Integrity, and Usability [6].

Equation 1 is a formula used to measure the factors of software quality in this study.

$$Fq = c_1 * m_1 + c_2 * m_2 + \cdots + c_n * m_n \quad (1)$$

Where:
Fq = Software Quality Factor
c_1 = The first scale that depends on the product
m_1 = The first metric that affects the software quality factor

Equation 2 is used to test the validity of the population sample, which uses a 95% confidence level.

$$MoE = z * s/\sqrt{n * (N-n)/(N-1)} \quad (2)$$

Where:
N = Size of Population
n = Size of Sample
s = Standard Deviation
z = Level of Confidence (95% = 1.96 and 99% = 2.58)

$$MoE = 1.96 * 0.5/\sqrt{60 * (763-60)/(763-1)}$$

$$MoE = 1.96 * 0.5/\sqrt{60 * \frac{703}{762}}$$

$$MoE = 0.02$$

From the calculation of the margin of error with a confidence level of 95%, the result is 0.02%. So it can be concluded that the number of correspondents used in this study is in accordance with the standards [9].

III. RESULT AND DISCUSSIONS

The results of the questionnaire were obtained, then it will be analyzed using the McCall method, which focuses on product operation with the five metrics used as in Figure 2 with a brief discussion that will be described as follow.

A. Correctness

This test is to determine whether the software meets the specifications and mission objectives of the user [6]. This metric consists of three test factors, namely Completeness, Traceability, and Consistency.

1) Completeness: This factor is to find out how far the implementation achievement and the required functions are achieved [8]. The software has 23 tested features. There are 21 features that match their input and output and two features that don't match. Thus, the completeness value is 21/23 or 91.3%, with the conclusion that the software has a very good completeness value.

2) Traceability: This factor is the ease of referring back to the implementation or component of the program to the needs of the software user [6]. From 23 software features tested, there were 18 features that matched the system and its analysis documents and five features that were incompatible. The traceability value is 18/23 or 78.26%, with the conclusion that the software has a good suitability value.

3) Consistency: This factor is the degree of use of the appropriate design techniques and documentation in all software development projects [8]. Tests carried out on four software menus. The total remark of suitability is four menus. So the consistency value is 4/4 of 100% with the conclusion that the software has excellent consistency between the analysis document and the system. Based on the results of testing the three Correctness factors, the percentage of the final result can be calculated as in equation 3.

$$Correctness = \frac{Completeness + Tracebility + Consistency}{3} \quad (3)$$

$$Correctness = \frac{91.3 + 78.26 + 100}{3}$$

$$Correctness = 89.85\%$$

It can be concluded that the Correctness value of the software is very good [10].

B. Reliability

This test is carried out to determine how far the software can be expected to perform its functions with the required accuracy. This test has five factors, namely, Accuracy, Simplicity, Error Tolerance, Modularity, and Consistency.

1) Accuracy: This factor is used to determine the accuracy value in program computation and control [8]. Tests are carried out on the existing menu in the software. From 29 software features, 24 features have compatibility between input and output computations, while five features do not match. The accuracy test result is 24/29 or 82.75%, with the conclusion that the accuracy of the software is very good.

2) Simplicity: This factor is a test of simplicity in the use of source code, which refers to how many a module is called, if it is greater or equal to the number of calls for a module, the easier it is to understand (simple) [6]. The total features tested were 30 features with 21 simple features and nine features that were not simple. The simplicity value is 21/30 or 70%, with the conclusion that the level of simplicity in using the source code in the software is good.

3) Error Tolerance: This factor is a test carried out to determine the tolerance value for errors that occur in software [8]. Tests are carried out on 29 software features where there are 26 suitable features so that the error tolerance value is 26/29 or 89.65% with a very good conclusion intolerance of errors that occur during use.

4) Modularity: This test is carried out to determine how independent the modules which are components of the software [6]. The test was carried out on five classes with one independent class and four dependent classes. The value of the modularity factor is 1/5 or 20%, with the conclusion that the software has a very poor value for module independence.

5) Consistency: This test is the same as described in the previous Correctness metric test. So the consistency value is 4/4 of 100% with the conclusion that the software has excellent consistency between the analysis document and the system. Based on the test results of the five Reliability factors, the final result can be calculated as equation 4.

$$Reliability = \frac{Accuracy+Simplicity+Error\ Tolerance+Modularity+\ Consistency}{5} \quad (4)$$

$$Reliability = \frac{82.75 + 70 + 89.65 + 20 + 100}{5}$$

$$Reliability = 72.48\%$$

From the calculation, it can be concluded that the software Reliability value is good [10].

C. Efficiency

This test aims to determine the number of computational resources and program code needed by the software to perform its functions. This metric has three factors, namely Conciseness, Execution Efficiency, and Operability.

1) Conciseness: This factor is a test to determine the conciseness of the program in line with commands (LOC) measures [8]. This test uses 4 out of the total 67 classes that have 3419 LOC by using the calculation of equation 6. To determine the number of classes used can be with equation 5.

$$Classes\ X = \frac{Total\ Declaration\ of\ Classes\ X}{Total\ LOC} \quad (5)$$

$$Classes\ X = \frac{67}{3419} = 0.0196$$

$$Conciseness = \frac{Total\ Classes\ Value}{Total\ Classes}\ x\ 100\%$$
$$Conciseness = \frac{0.0196}{4}\ x\ 100\% = 49\% \quad (6)$$

The conciseness test result is 49%, which indicates that the average density of LOC is good enough.

2) Execution Efficiency This factor is used to calculate the run-time efficiency of the software when used on the device [6]. The calculation includes the RAM used when installed until the software is run. Testing was carried out using the realtime profilers feature on Android Studio with equation 7.

$$Execution\ Efficiency = \frac{Average\ RAM\ Used}{Total\ RAM}\ x\ 100\% \quad (7)$$

From equation 7, the value is 12.8%. This means that the software is quite efficient [8].

3) Operability: This factor tests the ease of operation of the software. The value of this factor is obtained by distributing questionnaires, as described in chapter 2. The results were 83.2%, with the conclusion that users strongly agree with the ease of use of the software. Based on the test results of the three Reliability factors, the percentage of the final results can be calculated as follows:

$$Efficiency = \frac{Conciseness+Execution\ Efficiency+Operability}{3} \quad (8)$$

$$Efficiency = \frac{49\% + 12.8 + 83.2}{3}$$

$$Efficiency = 48.3\%$$

It can be concluded that the Efficiency value of the software is still quite good.

D. Integrity

This test aims to determine the number of computational resources and program code needed by the software to perform its functions. This metric has three factors, namely, Audibility, Instrumentation, and Security.

1) Audibility: This factor is to test whether the software meets predetermined standards [8]. From 39 features tested, 31 were in accordance with the specified standards. The calculation of the audibility value uses equation nine as follows:

$$\%\ Audibility = \frac{Total\ Map}{\sum Total\ Function}\ x\ 100\% \quad (9)$$

$$\%\ Audibility = \frac{31}{39}\ x\ 100\% = 79.49\%$$

The result of audibility testing is 79.49% of the features that meet the predetermined standards are categorized as good.

2) Instrumentation: This factor is used to monitor and identify operating errors of the program [8]. There are 41 features tested, of which 36 features identify program operation errors. The calculation is performed using the following formula:

$$\%\ Instumentation = \frac{Total\ Instrument}{Total\ Item\ Instrument\ Given} x\ 100\% \quad (10)$$

$$\%\ Instumentation = \frac{36}{41} x\ 100\% = 87.8\%$$

From equation 10, the value is 87.8%, which means that the software is very good for identifying program operation errors.

3) Security: This factor tests the availability of mechanisms to control and protect software and data from unauthorized parties [8]. There are five security activities that were tested with the results of 4 features that have security. By using the equation, the security value is calculated as follows:

$$\%\ Security = \frac{Total\ Function\ Secured}{\sum Total\ Function} x\ 100\% \quad (11)$$

$$\%\ Security = \frac{4}{5} x\ 100\% = 80\%$$

The conclusion of security testing is good. Furthermore, based on the results of testing the three Integrity factors, the percentage of the final results is obtained using equation 12.

$$Integrity = \frac{Audability + Instrumentation + Security}{3} \quad (12)$$

$$Integrity = \frac{79.49\% + 87.8 + 80}{3}$$

$$Integrity = 82.43\%$$

It can be concluded that the value of software integrity is very good.

E. Usability

This factor is for studying, operating, preparing input, and interpreting the output of the software [6]. There are two metrics for this test, namely Operability and Training.

1) Operability: This metric is used to test the ease of operation of the software. This test has been discussed in the previous Efficiency Factor and has the same value, namely 83.2%, which means that users strongly agree that the software is easy to use.

2) Training: This factor is to know the extent of how far the software helps new users understand the system [8]. There are five menus, 4 of which have training. The results are calculated using equation 13.

$$Training = \frac{Total\ Function\ Explained}{\sum Total\ Function} x\ 100\% \quad (13)$$

$$Training = \frac{4}{5} x\ 100\% = 80\%$$

The result of testing is that the software is good at providing ease of understanding by new users.

Based on the test results of the two usability factors, the percentage of the final results is obtained using equation 14.

$$Usability = \frac{Operability + Traingin}{2} \quad (14)$$

$$Usability = \frac{83.2\% + 80}{2} = 81.6\%$$

The conclusion from the usability aspect is that the software is very good at providing ease of use.

From the calculation results, the correctness metric value is 89.85%, the metric reliability value is 72.48%, the metric efficiency value is 48.3%, the metric integrity value is 82.43%, and the metric usability value is 81.6%. After obtaining all the values from the five aspects, the quality factor can be calculated using equation one so that the results are 74,932% and in the good category [5]. This result is similar to previous research which has good category results [7]-[11].

IV. CONCLUSION

From a series of quality tests that have been carried out, it can be concluded that the McCall method can be applied to test the quality of e-voting software. From the five focused Product Operation factors generally have good results based on McCall's quality theory. The metric with the best quality was correctness with 89.85%, and the metric with the lowest quality was the efficiency with a value of 48.3%. This shows that e-voting software is still lacking in terms of efficiency, especially in terms of conciseness, execution efficiency, and operability. By knowing the shortcomings of e-voting software, further research can resolve these deficiencies.

ACKNOWLEDGMENT

The author would like to thank ISB Atma Luhur for funding this research so that it can be completed properly.

REFERENCES

[1] A. Basu, *Software Quality Assurance, Testing, and Metrics*. PHI Learning Pvt., 2015.

[2] A. Hidayati, E. Oktariza, F. Rosmaningsih, and S. A. Lathifah, "Analisa Kualitas Perangkat Lunak Sistem Informasi Akademik Menggunakan McCall," *Multinetics*, vol. 3, no. 1, pp. 47–51, 2017, doi: 10.32722/vol3.no1.2017.pp48-53.

[3] F. P. Juniawan, "RSA Implementation for Data Transmission Security in BEM Chairman E-voting Android Based Application," in Proceedings - 2016 1st International Conference on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2016, 2016, pp. 93–98, DOI: 10.1109/ICITISEE.2016.7803054.

[4] D. P. Mandala and A. Dewanto, "Uji Kelayakan Sistem Informasi Unit Kesehatan Sekolah Berbasis Website Di Smk Muhammadiyah 1 Bantul Dengan Faktor Kualitas Mccall," *Elinvo (Electronics, Informatics, Vocat. Educ.*, vol. 2, no. 2, pp. 195–203, 2017, doi: 10.21831/elinvo.v2i2.17314.

[5] C. Juliane, R. Dzulkarnaen, and W. Susanti, "Metode McCall ' s untuk Pengujian Kualitas Sistem Informasi Administrasi Tugas Akhir (SIATA)," *J. Resti*, vol. 3, no. 3, pp. 488–495, 2019.

[6] B. Sugiantoro and M. Mustakim, "Analisa Usabilitas Sistem Deteksi Akses Pornografi Pengguna Internet Menggunakan Metode Mccall'S,"

J. Online Inform., vol. 2, no. 1, pp. 56–62, 2017, doi: 10.15575/join.v2i1.60.

[7] A. Andrianti, "Pengukuran Kualitas Aplikasi Rekap Indikator Mutu Harian RS Bhayangkara Jambi Menggunakan Metode McCall," vol. 14, no. 1, pp. 24–34, 2020.

[8] M. Agus, "Pengujian Sistem Informasi Akademik menggunakan McCall's Software Quality Framework," *J. Inform. Sunan Kalijaga*, vol. 1, no. 1, pp. 47–57, 2016.

[9] A. H. Arif and M. U. Siregar, "A Usage of McCall's Software Quality Analysis on the Bonus System of PT Surya Pratama Alam," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 3, no. 1, pp. 63–72, 2018, DOI: 10.14421/jiska.2018.31-07.

[10] F. Gunadi and S. R. Widianto, "Evaluasi Kualitas Pelaporan Manajemen pada Sistem Epicor Perusahaan Manufaktur Berbasis McCall," *J. MULTINETICS*, vol. 6, no. 1, pp. 21–31, 2020.

[11] D. Mahdiana and N. Kusumawardhany, "The Combination of Analytical Hierarchy Process and Simple Multi-Attribute Rating Technique for the Selection of the Best Lecturer," *Proceeding - ICoSTA 2020 2020 Int. Conf. Smart Technol. Appl. Empower. Ind. IoT by Implement. Green Technol. Sustain. Dev.*, 2020, DOI: 10.1109/ICoSTA48221.2020.1570615695.