Development of Software Quality Assessment Model for Mobile-based Elderly Fall Detection Software

A Finandhita
Program Studi Teknik Informatika, Universitas Komputer Indonesia, Indonesia
Email: alif.finandhita@email.unikom.ac.id

Abstract. The purpose of this research is to build a software quality assessment model to evaluate the quality of mobile-based elderly fall detection software. The assessment is based on the quality factors found in the software quality model. The quality factor is adjusted to the characteristics of the software. The model is needed because the software has its own characteristics. This research consists of several stages. The first thing to do is analysing the software domain to determine its characteristics. The second is defining the software assessment needs by mapping software characteristics with the quality standards used (ISO / IEC 25010: 2011) to obtain the appropriate quality factors. The software quality metrics is determined after the quality factors is obtained. The metric to be used is Goal Question Metrics (GCM). The third is software quality weighting process, including its criterias and sub-criterias. Determination of the equation for software quality assessment is the final stage of the research. Based on the reseaech process, it can be concluded that the model developed successfully can be used to assess the software.

1. Introduction
Movement is an action or state of motion, which states the action of an individual in a situation. Age is one of the factors that can affect the quality of one's movements. Elderly prone to falling when moving, the condition of the fall of the elderly is difficult to detect. In a long period of time these conditions can have physical and psychological impacts. If emergency treatment is delayed, injuries caused by falls can result in disability, paralysis, and even death. Therefore, the role of the family is very important as an elderly support system. There are several softwares that can be used to detect falls in the elderly. From some of the software, there is one software that in addition can be used by the elderly can also be used by family members or closest relatives to receive information if the elderly falls.

Mobile-based elderly fall detection software is used as research object. The problem arises when the software is used by users, the response is slow. This can result in slower user work. Inappropriate software architecture design can affect the quality of the software [1]. On the other hand, fall detection software for the elderly requires a fast response, especially when facing an emergency. Boehm and Jones stated that errors in each newly developed software do exist scientifically so that it can make the work of the software more difficult [2]. The quality assessment model of falling detection software in the elderly is needed to maintain software quality and also for software development in the future. The ISO / IEC 25010 standard is used as the main reference for evaluating software [3]. The assessment based on these standards is carried out through the stages of defining software requirements based on the interrelationships between the characteristics of the software. The Goal Question Metrics (GQM) approach is used in assessing software quality factors through a questionnaire based on the
characteristics of the software. The results of the questionnaire can be used to weight each factor and sub-factor quality.

We propose a software quality model assessment for mobile-based elderly fall detection software in this research. The model developed can be used to assess the quality of the software. The assessment process is carried out to maintain and increase the quality of the software. In fact, there have been several studies related to software quality assessment model, as conducted by Wolski et al. [4], Vijay et al.[5], Tambotoh et al. [6], and Kaur et al [7]. There are also several studies of elderly fall detection software, as conducted by Senouci et al [8], Waheed et al. [9], and Fayad et al. [10]. However, there are no specific development of software quality model that can be used to assess the quality of mobile-based elderly fall detection software. Based on the problems previously explained, a model to assess the quality of the software is needed. The model is expected to maintain and improve the quality of mobile-based elderly fall detection software. Farmers in improving the quality and quantity of their agricultural products. Thus, the performance of the software is expected to be increased. The purpose of this research is to build a software quality assessment model to evaluate the quality of mobile-based elderly fall detection software. This research consists of several stages, The first thing to do is analysing the software domain to determine its characteristics. The second is defining the software assessment needs by mapping software characteristics with the quality standards used (ISO / IEC 25010: 2011) to obtain the appropriate quality factors. The software quality metrics is determined after the quality factors is obtained. The metric to be used is Goal Question Metrics (GCM). The third is software quality weighting process, including its criterias and sub-criterias.

2. Method

Figure 1 shows the descriptive qualitative method used to develop the model (Bradway, 2018; Nassaji, 2015).

![Figure 1. Descriptive Qualitative Method](image)

This research consists of several stages. The first thing to do is analysing the software domain to determine its characteristics. The second is defining the software assessment needs by mapping software characteristics with the quality standards used (ISO / IEC 25010, 2011) to obtain the appropriate quality factors. The software quality metrics is determined after the quality factors is obtained. The metric to be used is Goal Question Metrics (GCM). The third is the construction of the model. Evaluation of the model is done as the final stage of the research.

3. Results and Discussion

Good software must be usable and useful [11]. Mobile based software is very commonly used today [12]. Therefore, software built on mobile architecture must have specific and efficient capabilities, so that the functionalities contained in it can be run properly. Some important characteristics of good mobile based software are fast start up time, responsive to user input, and defined purpose [12]. These characteristics are very important for user productivity when using mobile based software. Mobile
devices have a different architecture to desktop and laptop computers. Mobile devices only rely on RAM as a center for data storage and program execution. In addition, there is flash memory for long-term storage. In the process of developing mobile-based software, memory management becomes very important because of the limited storage media. Mobile-based software is more often connected to the internet (online) compared to desktop devices. Therefore, mobile based software must be reliable and can handle resources better to avoid memory leakage so that the performance can improve [13].

The mobile based elderly fall detection software that is the object of this study used sensor fusion to detect elderly movements. This sensor combines two different types of sensors with almost the same characteristics to get a new function [14]. The two types of sensors combined are the accelerometer sensor and the gyroscope sensor. With these sensors, the movement of the elderly when carrying a smartphone can be detected. When the elderly fall, the value of acceleration and angular velocity changes dramatically. Thus, the software will detect the condition and then provide notification information to family or closest relatives who are connected with the software. Figure 2 shows elderly fall detection system architecture.

**Figure 2. Elderly Fall Detection System Architecture**

The majority of fall detection systems require specific design software. However, the need for good software remains the same [15]:

1. Reliability
2. Ease of use
3. Restriction of false positives.

The main reason for the many failures in the fall detection system is the rejection by monitoring services contained in the software used, which is caused by the number of false alarms [16]. When a software failure occurs, it will affect the quality of the software.

The elderly fall detection software is mapped against the ISO / IEC 25010 standard based on the domain and its characteristics [1]. The process is carried out to obtain software quality criteria along with the sub-criteria that correspond to the software. This is important to do, so that the assessment process can run well. Figure 3 shows the results of the mapping process.
Figure 3. The Results of the Mapping Process

Each quality factor obtained from the mapping results has their respective criteria as shown in table 1.

| Criteria            | Sub-Criteria                                |
|---------------------|---------------------------------------------|
| Functional Suitability | Functional Completeness                     |
|                     | Functional Correctness                      |
|                     | Functional Appropriateness                  |
| Usability           | Appropriateness                              |
|                     | Recognizability                             |
|                     | Operability                                  |
|                     | Use Error Protection                         |
|                     | Use Interface Aesthetics                     |
|                     | Accessibility                                |
| Performance Efficiency | Time Behaviour                              |
|                     | Resource Utilization                         |
|                     | Capacity                                     |
| Maintainability     | Reusability                                  |
|                     | Analysability                                |
|                     | Testability                                  |
| Reliability         | Maturity                                     |
|                     | Availability                                 |
|                     | Fault Tolerance                              |
| Security            | Confidentiality                              |
|                     | Integrity                                    |
|                     | Authenticity                                 |
| Portability         | Replaceability                               |

The Goal Question Metric (GQM) approach is a method for driving goal-oriented measures throughout a software organization. It can be used to measure software quality characteristics and sub-
characteristics values. Several metrics can be used for several different questions with the same purpose [17]. Figure 4 shows the GQM hierarchical structure.

![The GQM Hierarchical Structure](image)

**Figure 4.** The GQM Hierarchical Structure

In this study, GQM is used to form questions that refer to the characteristics and sub-characteristics of software quality that have been obtained previously. Likert and Guttman scales are used to calculate the value obtained based on the questions formed [18]. Table 2 shows the form of GQM used in the quality assessment model of mobile-based elderly fall detection software.

| Goal | Obtain value of software quality sub-criterias |
|------|-----------------------------------------------|
| Questions | The questions made refer to the definitions of each software quality sub-criterias |
| Metrics | M1 = Likert Scale Weight \( \times \) Results of Respondents' Answers \\
| | \( x = \sum_{n}^{Total} \frac{x}{n} \) |
| | M2 = Guttman Scale Weight \( \times \) Results of Respondents' Answers \\
| | \( x = \sum_{n}^{Total} \frac{x}{n} \) |

Weighting process is needed so that the value of software quality sub-criteria can be calculated. The Rank Order Centroid (ROC) method is used for this purpose [19]. The following is the explanation for the ROC equation: If \( C_{1} \geq C_{2} \geq C_{3} \geq ... \geq C_{n} \), Then \( W_{1} \geq W_{2} \geq W_{3} \geq ... \geq W_{n} \). Furthermore, if \( k \) is the number of criteria, then:

\[
W_{1} = \frac{1 + \frac{1}{2} + \frac{1}{3} + ... + 1/k}{k}
\]

\[
W_{2} = \frac{0 + \frac{1}{2} + \frac{1}{3} + ... + 1/k}{k}
\]

\[
W_{3} = \frac{0 + 0 + \frac{1}{3} + ... + 1/k}{k}
\]

In general, the ROC weighting can be formulated as follows:

\[
W_{k} = \frac{1}{k} \sum_{i=1}^{k} (1/i)
\]
\[ W = \text{Weighting Value} \]
\[ k = \text{Number of criteria} \]
\[ i = \text{Alternative value} \]

The \( k \) value is obtained from the number of links between domains and software criteria with ISO / IEC 25010. The number of links from the mapping process that has been done is 7. Thus, the weighting results can be seen in Table 3.

### Table 3. Software Quality Criteria Weighting Calculation Results

| Criteria          | Weighting Calculation |
|-------------------|-----------------------|
| Functional Suitability | \( \frac{1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7}}{7} = 0.37 \) |
| Maintainability    | \( \frac{1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7}}{7} = 0.23 \) |
| Usability          | \( \frac{1 + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7}}{7} = 0.15 \) |
| Security           | \( \frac{1 + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7}}{7} = 0.11 \) |
| Reliability        | \( \frac{1 + \frac{1}{5} + \frac{1}{6} + \frac{1}{7}}{7} = 0.07 \) |
| Performance        | \( \frac{1 + \frac{1}{6} + \frac{1}{7}}{7} = 0.04 \) |
| Portability        | \( \frac{1}{\frac{1}{7}} = 0.02 \) |

Sub-criteria weighting process is needed to calculate the value of software quality sub-criteria. The value of \( k \) owned by the software quality sub-criteria varies according to the number of software quality sub-criteria owned by each criterion. Thus, the weighting results can be seen in Table 4.

### Table 4. Software Quality Sub-Criteria Weighting Calculation Results

| Criteria           | Sub-Criteria                  | Weighting Calculation |
|--------------------|-------------------------------|-----------------------|
| Functional Suitability | Functional Completeness     | \( \frac{1 + \frac{1}{2} + \frac{1}{3}}{3} = 0.61 \) |
|                     | Functional Correctness        | \( \frac{1 + \frac{1}{2} + \frac{1}{3}}{3} = 0.28 \) |
|                     | Functional Appropriateness    | \( \frac{1}{3} = 0.11 \) |
| Performance Efficiency | Time Behaviour               | \( \frac{1 + \frac{1}{2} + \frac{1}{3}}{3} = 0.61 \) |
|                     | Resource Utilization          | \( \frac{1 + \frac{1}{2} + \frac{1}{3}}{3} = 0.28 \) |
Based on the results of determining the GQM metric in table 2 as well as the results of weighting the criteria and sub-criteria of software quality in table 3 and table 4, the quality of mobile-based elderly fall detection software can be assessed using the weight summation method [19]. The method has the following equation:

\[ \text{score } (Ci) = \sum_{i=1}^{n} W_i . X_i (Ci) \]
Table 5 shows the categories that can be used for quality groupings based on the final results of each assessment of software quality criteria

| Categories | Interval |
|------------|----------|
| High       | 3.36 – 5 |
| Medium     | 1.68 – 3.35 |
| Low        | 0 – 0.167 |

4. Conclusion
The developed model can be used to assess the quality of mobile-based elderly fall detection software. The model provides quality criteria and sub-criteria that have been adjusted to the characteristics of the software. Each criterion and sub-criterion has its own weight. The weight can be used to calculate the value of software quality based on criteria and sub-criteria, or the quality of the software in general.

Acknowledgement
We acknowledged all parties for supporting this research by providing the required data.

References
[1] J. Tian, “What Is Software Quality?,” pp. 15-26, 2005.
[2] D. Galin, Software quality assurance: from theory to implementation, Software Engineering, IEEE, 2004, p. 616.
[3] “ISO/IEC 25010,” [Online]. Available: http://iso25000.com.
[4] Wolski, Marcin, Bartosz Walter, Szymon Kupiński, and Jakub Chojnacki. "Software quality model for a research-driven organization—an experience report." Journal of Software: Evolution and Process 30, no. 5 (2018): e1911.
[5] Vijay, T. John, D. M. G. Chand, and D. H. Done. "Software quality metrics in quality assurance to study the impact of external factors related to time." International Journal of Advanced Research in Computer Science and Software Engineering 7, no. 1 (2017).
[6] Tambotoh, Johan JC, Sani M. Isa, Ford Lumbar Gaol, Benfano Soewito, and Harco Leslie Hendric Spits Warnars. "Software quality model for Internet of Things governance." In 2016 International Conference on Data and Software Engineering (ICoDSE), pp. 1-6. IEEE, 2016.
[7] Kaur, Simarpreet, Sumeet Kaur Sehra, and Sukhjit Singh Sehra. "A framework for software quality model selection using topsis." In 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), pp. 736-739. IEEE, 2016.
[8] Senouci, Benaoumeur, Imen Charfi, Barthelemy Heyrman, Julien Dubois, and Johel Miteran. "Fast prototyping of a SoC-based smart-camera: a real-time fall detection case study." Journal of Real-Time Image Processing 12, no. 4 (2016): 649-662.
[9] Waheed, Shaikh Abdul, and P. Sheik Abdul Khader. "A novel approach for smart and cost effective IoT based elderly fall detection system using Pi camera." In 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCCIC), pp. 1-4. IEEE, 2017.
[10] Fayad, Moustafa, Ahmed Mostefaoui, Samir Chouali, and Salima Benbernou. "Fall Detection Application for the Elderly in the Family Heroes System." In Proceedings of the 17th ACM International Symposium on Mobility Management and Wireless Access, pp. 17-23. 2019.
[11] Gillies, Alan. Software quality: theory and management. Lulu. com, 2011.
[12] Salmre, Ivo. Writing mobile code: Essential software engineering for building mobile applications. Addison-Wesley Professional, 2005.
[13] Zahra, Sobia, Asra Khalid, and Ali Javed. “An Efficient and Effective New Generation Objective Quality Model for Mobile Applications.” International Journal of Modern Education and Computer Science 5 (4): 36–42. https://doi.org/10.5815/ijmecs.2013.04.05. 2013.
[14] Yoon, Ja-Woong, Yeon-Sik Noh, Yi-Suk Kwon, Won-Ki Kim, and Hyung-Ro Yoon. "Improvement of dynamic respiration monitoring through sensor fusion of accelerometer and gyro-sensor." Journal of Electrical Engineering and Technology 9, no. 1 (2014): 334-343.

[15] K Doughty, R Lewis, and A McIntosh. The design of a practical and reliable fall detector for community and institutional telecare. Journal of Telemedicine and Telecare, 6(1):150–154, 2000.

[16] N Noury, A Fleury, P Rumeau, A.K. Bourke, G. ´o Laighin, V Rialle, and J.E. Lundy. "Fall detection - principles and methods". pages 1663–1666. Proceedings of the 29th Annual International Conference of the IEEE EMBS, August 2007.

[17] V. R. Basili, G. Caldiera dan R. H. D., “The goal question metric approach,” Encyclopedia of Software Engineering, vol. 2, pp. 528-532, 1994.

[18] W. Budiaji, “LIKERT ( The Measurement Scale and The Number of Responses in Likert Scale ),” vol. 2, no. 1, pp. 125-131, 2013.

[19] E. Roszkowska, “Rank ordering criteria weighting methods,” Optimum. Studia Ekonomiczne, vol. 5, no. 5, pp. 1-20, 2013.