MANTECH Publications

Journal of Software Engineering & Software Testing
Volume 7 Issue 2
ISSN: 2457-0516 (Online)

The ISO/IEC Standard Outlines Ten Steps for Evaluating Software's Quality

Gourav Chattopadhyay[1], Pravat Bhatnagar[2]

Assistant Professor[1], Student[2]

Department of CSE

Chintalapudi Engineering College

Corresponding Author's Email: - *bhatnagarpravat@redffmail.com[2]*

Abstract

To evaluate the quality of software, a measurement reference model must be used in the software rating field. The standard 25030 is an example of an assessment system based on the needs of stakeholders. This study attempted to build a model that took into consideration the implicit and explicit requirements of stakeholders, users, and policymakers. Furthermore, the AHP approach was utilised to weight the indications in the model. The findings demonstrate the model's suitability for meeting the needs of Iranian users.

Keywords: *Software Quality Evaluation, Software Rating, Content Based Software Packages, Software Evaluation Standard*

INTRODUCTION

In the field of software rating, strategies based on user needs are regarded as effective ways for achieving improved interaction between supply and sale sections. A quality evaluation technique must be used to rate the quality of the created software. Taking the inclusive perspective of the capabilities of software's functions into consideration, it is impossible to assess the contents of software using it, because the model is incapable of evaluating the content. There are also different ways of software assessment that do not rely on user feedback. These approaches are more scientific, as they investigate and concentrate on standard aspects of a software function. Acceptable references define and show the qualities mentioned. ISO can be used as an example in this case. Alizadeh et al. [1] propose various

Page 64-71 © MANTECH PUBLICATIONS 2022. All Rights Reserved

groups after the extraction and localisation of software assessment elements. Beata et al. [3] and Kasunie [5] have worked on several types of measurements in this area. That is to say, the outcome of any output may be presented in a variety of ways. This can be approved or denied, calibrated, ranked, or rated (that is done in this survey). Production and outcomes can also be categorised (without priority) or rated. The output of the given studies is a ranking of software that is utilised as a consequence of previous studies.

Software is evaluated based on the major goal of categories in four groups: educational, Encyclopedic, general, and child and adolescent. Separate generic criteria, such as user interface (UI) and installation, are examined in each area. Specific weights are assigned to each feature in each software category based on the findings of the AHP model on experts.

It should be noted that if the technique of assessment is created based on the 25030 set [1-4], it will be acceptable. Using the criteria given above, this study seeks to derive a general model that may be used to measure software quality in 10 phases.

Section 2 focuses on identifying system needs and stakeholders. These sorts of requirements can also be established in two unique formats. Section 3 contains models and criteria. This section discusses ISO and its characteristics. In the preceding sections, the result of weighing and methods of calculating marks for each programme are specified, with the mark ranging from zero to one hundred (0-100). The conclusion and tasks that can be completed in the future are presented in the last section.

Requirements

The system requirements are determined in the first stage. This process is completed in five stages. The first and second phases are devoted to clarifying assessment objectives and investigating stakeholders, respectively. As the primary stakeholders, the second phase also includes an evaluation of user requirements [5, 6]. Figure 1 depicts the relationship between the system's needs and those of the stakeholders.In the first phase, the evaluation purpose was specified as software quality evaluation with qualitative requirements.

The sort of product for assessment that is connected to the evaluation purpose is determined in the second phase. The most important aspect of the evaluation process is identifying the products. For assessment

Page 64-71 © MANTECH PUBLICATIONS 2022. All Rights Reserved

purposes, this paradigm considers media software to be the product. Users' scientific and cultural understanding is increased as they are entertained by media software. They also have an impact on the psychological and cultural elements of users, both directly and indirectly.

Stakeholders

The third stage is to identify the stakeholders in the system. A stakeholder is a person who has the right to claim or be included in the system in order to meet their needs and expectations. Stakeholder requirements and expectations vary and may be categorised into three broad categories:

Software producers

This category either includes materials that are used to impact the cultural, mental, and psychological elements of users, or it attempts to entertain audiences by delivering fascinating contents and functionalities. Some necessary tools or functions are developed to do this.

End users

This category includes software audiences whose needs for the program's contents or functionality are met by utilising the software.



Figure1: Analysis of stakeholders' requirements

Policy makers and supervisors

This category includes governments and supervisory institutes whose responsibility it is to examine published digital materials or analyse the digital publication domain in order to research the status of individuals, society, and digital publishers. The outcome will be used in cultural, social, political, and security contexts.

It should be mentioned that the scope of software qualitative requirements in this

Page 64-71 © MANTECH PUBLICATIONS 2022. All Rights Reserved

study includes needs stated by users and policymakers. Once the stakeholders have been identified in the fourth stage, their wants and expectations, as well as recognised limitations on the clients' and ultimate users' requirements, are considered. There are two types of user needs: implicit and explicit requirements.

Explicit requirements

The users set the explicit needs. The completion of this sort of demand leads to a greater knowledge of the capacity of the programme that consumers utilise. When utilising software, an end user follows two goals and two essential aspects. In other words, two essential criteria, known as explicit requirements, drive people to acquire software. This sort of requirement is associated with software content and the operation of certain functionalities. Some of the functions accessible in media software include: media dissemination, teaching, and research (offering research content, sound and video processing function, computational function).

Implicit requirements

They are stated, but they are not the true needs. According to the study area of media software makers and judgement specialists, the following implicit demands for any programme were identified:

1. **Packaging appearance**

Because the customer receives the media software product in the form of an in-store package, the look of the software packaging is one of the user requirements. Any media programme, in fact, can be termed a "commercial off the shelf software product."

2. **Internal consistency and installation:**

Because media software is, by definition, software, two essential aspects must be addressed while utilising it. Internal efficiency and consistency must be met. In other words, it possesses the qualities of dependability, efficiency, maintainability, and security without any failure or fault. These features are required in order to install, run, activate, and uninstall a programme, as well as to have the right software type and address agreement.

3. **User interface**

It is a viewable and touchable component of software with which the user interacts directly, and it comprises information channels that allow communication between the user and the computer. In media software, the user interface is often one of two types: choice interface or user graphical interface. User interface is another implicit user need.

Page 64-71 © MANTECH PUBLICATIONS 2022. All Rights Reserved

4. Support:

Receiving supported services is one of the users' necessities, as the bulk of media software users are regular people. The fifth stage determines system requirements after determining stakeholder needs. A system frequently comprises several pieces with precise specifications that react to the system's various functions. To function, the system needs must be converted into the requirements of various system parts. The outcome of the requirement definition process is known as stakeholders requirements. The quality criteria are defined in this stage for each of the items defined in the previous step (extracted from the user needs).

Model and Criteria

The reference model and its criteria are created and decided at this stage. Based on the specified needs, two characteristics have been determined: content and function. The other four criteria, including software packaging, internal consistency, user interface, and supported services, will be determined in relation to the inferred needs.

The software quality model is completed by establishing criteria (attributes) for each of the six previously mentioned features (in three layers). The criteria (attributes) were selected in the sixth phase. An attribute is a natural trait of a species or thing that may be quantified and qualitatively determined by human or machine techniques.

Attributes are classified into two types: permanent attributes found in nature and acquired attributes of a system, process, or product (such as product price, product owner). The acquired attribute is not a system, method, or product's intrinsic qualitative attribute.

A software product's quantity and quality are determined by establishing some criteria that are connected to the quality attributes. In the seventh stage, the quality traits and standards for each attribute are defined in three levels. The quality model is constructed in the second phase using the criteria developed in the previous step. The quality of a system is determined by the quality of its constituents, and the quality of software is determined by its ability to satisfy implicit and explicit criteria under certain situations.

The model's quality is described as a collection of established qualities and their relationships that give a framework for determining quality standards for assessment purposes. The quality model is

Page 64-71 © MANTECH PUBLICATIONS 2022. All Rights Reserved

utilised as a framework to ensure the conformance of all quality elements with internal aspects as well as the user perspective. In relation to the retrieved criteria from the previous stage, the following quality model was extracted, and secondary characteristics were defined in each basic quality attribute. This paradigm defines two dimensions of quality:

- Internal software quality: software package, internal consistency, user interface, content, function, and support are all included.
- Usability: user ideas concerning software components are gathered.

Defined quality attributes encompass all quality factors for the majority of media software products, therefore they may be utilised to ensure comprehensive quality coverage. The following step is to create a quality model to determine the weight for each feature and criterion. One of the fundamental issues that arises during the weighing process is determining the significant differences between various media software. For example, in children's software, because of the need for beauty in software appearance, experts consider the user interface to be the most significant part, yet in encyclopaedia software, the major feature is the software's content.

As a result, at the next (eighth) phase, all software packages have been identified. Thus, taking into account the subject of research as well as expert opinion, the following main forms of content-based software were defined: Encyclopedia software, Training software, Children software, General software, and Functional software are all examples of software.

Weighting and Rating

The final stage of model development is to weigh and rate the levels. To establish the weights of traits and criteria, the AHP approach was used in the weighing procedure (applicable for any software). Finally, the quality of the programme was assessed using a population sample to produce rating levels based on the proposed quality measurement.

The weights of criteria were established using the AHP approach in the ninth phase. This is a well-known multi-criteria decision-making strategy. This strategy may be utilised to make a decision while taking into account several competing options and decision criteria. The criteria might be both qualitative and quantitative in nature. The difficulty is handled by creating a hierarchical structure. So, the first step is to create a decision tree with three tiers. The top level of the tree

Page 64-71 © MANTECH PUBLICATIONS 2022. All Rights Reserved

symbolises the decision maker's purpose, while the last level shows competing choices that are compared to one another. The goal is to calculate the weights based on the relative relevance of the given choices. The most essential level of the tree is the intermediate level, which represents certain criteria for comparing competitive options. Table 1 displays the primary six criteria and their associated weights for four different types of software.

Table 1: The weighting factors for characteristics of different software using AHP method

	Encyclopedic	Training	Children	General
Packaging	%5.3	%5.3	%55.3	%55.1
Internal Consistency	%5.3	%5.3	%5.7	%53.5
User Interface	%55.5	%57.5	%55.1	%55.5
Function	%55.5	%52.3	%57.5	%22.3
Content	%35.5	%27.2	%23.5	%55.7
Supportive service	%3.3	%7.1	%3.1	%5.5
Total	%511	%511	%511	%511

This level has many tiers. The AHP technique necessitates pairwise comparisons using a tilde. In this regard, the following processes were carried out using "Expert Choice" software:

Tenth step Determine rating level through following three processes (Table 2):

1. The software quality model was applied to evaluate 190 pieces of software, and the findings were obtained.
2. The findings were transmitted to some persons who were absolutely unfamiliar with the software evaluation procedure and had no prior knowledge of the software. They were tasked with categorising the transmitted findings.
3. Finally, the categorization list from the second stage was compared to the values from the first step. The highest and lowest values were found, and all values were normalised.

Table 2 shows the ranking based on the user needs. There are two more stars, "exportable" and "certain innovation and intelligence," that are complimentary stars (not stated in Table 2) with the goal of improving product quality.

Table 2: Range of scores corresponding to each star

Number of stars	Scores obtained
No star	$0 \le \alpha < 30$
One Star	$30 \le \alpha < 50$
Two star	$50 \le \alpha < 70$
Three star	$70 \le \alpha \le 100$

Page 64-71 © MANTECH PUBLICATIONS 2022. All Rights Reserved

CONCLUSION AND FUTURE WORKS

In this research, a quality evaluation methodology based on internal software quality and quality in use was proposed, taking into account the standard 25030. To construct the model, a standard technique was derived from standard 25030. Furthermore, this model is created from media software elements considering the cultural requirements of media software users. Furthermore, a scientific approach for measuring the quality of material in a measurement reference model and defining quality characteristics is provided, which covers all quality elements for the majority of media software. As a result, the model's use may almost guarantee total quality coverage. In future studies, the weighting of third and fourth functions can be considered to reduce the impact of personal judgements and to use more quantitative factors.

REFERENCES

1. Hassan Alizadeh, Hossein Afsari and Bahram Sadeghi Bigham, Rating System for Software based on International Standard Set (Square) 25000 ISO/IEC, Third International Conference on Contemporary Issues in Computer and Information Sciences. (CICIS 2012) , (2012) pp. 584-588.
2. Beata Czarnacka-Chrobot, The ISO/IEC Standards for the Software Processes and Products Measurement, SoMeT (2009), 187-200.
3. Beata CzarnackaChrobot, Analysis of the Functional Size Measurement Methods Usage by Polish Business Software Systems Providers, IWSM/Mensura (2009), 17-34.
4. A. Abran L. Buglione, The Software Measurement Body of Knowledge, Proceedings of 1st Software Measurement European Forum (SMEF), Rome (2004).
5. M. Kasunic, The State of Software Measurement Practice: Results of 2006 Survey, Software Engineering Institute, Carnegie Mellon University, Pittsburgh (2006).
6. CMMI Product Team, CMMI for Development, Version 1.2, Software Engineering Institute, Carnegie Mellon University, Pittsburgh (2006).
7. ISO/IEC 90003:2004 Software Engineering, Guidelines for the application of ISO 9001:2000 to computer software, ISO, Geneva (2004).

Page 64-71 © MANTECH PUBLICATIONS 2022. All Rights Reserved