American Journal of Computer Science and Information Engineering
2015; 2(3): 28-32
Published online September 10, 2015 (http://www.aascit.org/journal/ajcsie)

AASCIT American Association for Science and Technology

Keywords
Software Quality Assurance,
Software Metrics,
ISO 9126 Standards

Received: May 29, 2015
Revised: July 29, 2015
Accepted: July 30, 2015

Applying the ISO Standard in Assessing the Quality of Software Systems

Tawfig Abdelaziz[1], Mohamed Elammari[1], Warda Bani[2]

[1]Department of Software Engineering, University of Benghazi, Benghazi, Libya
[2]Department of Computer Science, Academy of Graduate Studies, Benghazi, Libya

Email address

tawfigtawuill@uob.edu.ly (T. Abdelaziz), Elammari.m@gmail.com (M. Elammari), saja_2012r@yahoo.com (W. Bani)

Citation

Tawfig Abdelaziz, Mohamed Elammari, Warda Bani. Applying the ISO Standard in Assessing the Quality of Software Systems. *American Journal of Computer Science and Information Engineering.* Vol. 2, No. 3, 2015, pp. 28-32.

Abstract

Owing to the rapid developments in data and information systems, the software system quality is becoming an increasingly important issue. Ensuring quality in any product or service requires setting standards and ensuring that these are adhered to. In software development, using measurement and metrics ensures reliability and guarantees that the performance meets the requirements. Many software systems fail to achieve the goals they were designed for, as their weaknesses are not discovered in a timely manner. Moreover, their assessment is often not based on scientific methods. In order to mitigate this issue, in this research, the ISO standard (9126) is used to evaluate the quality of software. The findings reveal shortcomings in the construction, maintenance, evolution and reuse. A descriptive survey methodology is used, as this enables applying the standards when evaluating the quality of software systems. The aim is to develop a standardized and uniformly applicable evaluation mechanism. This theoretical part of the research is accompanied by a practical part that constitutes of preparing and distributing questionnaires to apply the criteria determined by the ISO standard. Finally, in order to present the results, reach conclusions and make some recommendations for future work in this field, the data collected is subjected to statistical analysis. The software systems are evaluated by two categories of stakeholders, developers and users. These groups are deemed most relevant, as the developers are responsible for creating the software system, while the users ultimately determine its quality.

1. Introduction

Software systems nowadays play fundamental role of our new live, where it became pervasive in all aspects of society. Software system is a written computer programs and associated documentation pertaining to the operation of a computer system that are stored in a memory.

Most software systems fail completely or partial because they don't meet all their requirements. These requirements can be the cost, schedule, quality, or requirements objectives. According to many studies, failure rate of software systems ranges between 50% – 80% and to the findings of studies performed on the software systems, most suffer from deficiencies in achieving the goals, which they were designed for. The reasons for this suboptimal performance are inadequate preparation of specialists, lack of user participation, changing requirements, unrealistic project goals, inaccurate estimates of needed resources, badly defined system requirements, poor reporting of the project's status, lack of resources, unmanaged risks, poor communication among customers,

developers, and users, poor project management, lack of Stakeholder involvement, lack of training and awareness programs, and failure to apply and conform to international standards for quality management in the design and development of systems software [2].

Ensuring quality in software systems requires not only monitoring and management, but also adherence to strict standards. It is through measurement and metrics that software systems industry will ensure that the products and services meet the requirements [8][10].

When developing software products, companies must take into account international standards that apply to the entire project life cycle, and include all components and processes. In recognition of the growing need to ensure high quality software, many terms have emerged to represent overall dimensions that can be used to measure the quality and success of the software systems. However, software systems developers must take these measures into account while also meeting the requirements of the lowest cost, less effort, and easy management. In addition, implementation and development must also be in accordance with the renewable requirements of the beneficiary, which enable these systems to achieve their objectives [7].

In this paper, metrics that assess the quality of software systems are studied in detail, according to the quality standard of the International Organization for measurements (International Standard Organization, ISO) for the software quality. This is followed by a proposal of a structure that aims to identify deficiencies inherent in most software systems. This ensures that similar issues can be avoided in the current and future software systems.

2. Research Problem

Presently, many software systems fail in achieving all the objectives they were designed for because of the lack scientific methods that can be applied during their development, aiming to identify, evaluate and eliminate any weaknesses [6]. According to the study findings, a sample of systems that were analyzed suffered from deficiencies in the maintenance or evolution and reuse.

From here, it can be concluded that one of the reasons behind the failure of many software systems to achieve their objectives is the lack of communication between software developers and customers. In some cases, the customer cannot determine the requirements clearly, causing the developers to make assumptions that lead to subsequent issues. However, given that the developers are under pressure to deliver the finished product fast and at a minimal cost, they may not always adhere to the scientific stages of building software systems in a manner that ensures quality and consumer satisfaction [9].

The problem affecting the production of many software systems is absence of exact specifications and effective methods that can be used to detect and rectify errors. In addressing these issues, software project management plays a key role [1]. Effective management can ensure that standards that can be used to measure the quality of software are in place. Having an evaluation mechanism throughout the project life cycle provides a knowledge base that can be used to detect deficiencies in software systems.

Assessment of the software quality should focus on the results in terms of the benefits to the targeted beneficiaries of services. In addition, it should employ accurate analysis methods, be independent in estimation and based on consultation, and its findings should be reported in a transparent fashion. Accordingly, it is clear that the assessment involving precise analysis and assessment of the performance of the achievements of the institution and its software and projects is the most optimal solution [4]. The criteria below can be used for evaluating software systems:

1. Conformity with the strategic objectives of the institution, including systems containing software.
2. Establishing a link between the needs of the institution, and the software system users.
3. Ensuring quality and consistency of the software system design.
4. Evaluating the overall performance, particularly with respect to the quantitative and qualitative goals and objectives of the outputs.
5. Ensuring efficiency and cost-effectiveness, including the impact of software operations.
6. Ensuring feasibility of the continuation of benefits and improvements to be incorporated into software systems in the future.
7. Identifying factors that contributed positively or negatively in the production of software systems, and the underlying reasons.
8. Ensuring effectiveness of these software systems.

The methods used in the evaluation include analyses of specific criteria mentioned above, along with the comparisons between similar software systems. The data is collected through questionnaires and personal interviews with users and developers. In all cases, the successful evaluation is based on individual judgments, based on experience and available information [5].

3. Hypothesis of the Study

The aim of the study was to determine the level of the quality of software systems by applying criteria used in the field of software quality assurance. The research conducted was based on a descriptive approach for investigative standards that must be met in software systems in order evaluate tools in terms of both theoretical and analytical perspectives. The data required to meet these objectives was collected through questionnaires that were distributed to the institutions involved in the study, along with personal interviews with some participants. The gathered data was subjected to statistical analysis, which helped answer the research question, and offer some recommendations for future work.

4. Research Methodology

The methodology used in this research combined field work and analytical methods, as this allowed the views and opinions of study participants to be quantified and analyzed. The questionnaire items mainly focused on the various aspects of international ISO standards and their applicability in software system development. This served as a framework for evaluating software systems in order to increase their quality. The standard of interest for this research is ISO 9126. The questionnaire was divided into four parts, corresponding to the following topics: quality model, external standards, internal standards, and quality standards during use. Model quality standard ISO 9126 classifies the quality of software systems in a structured set of characteristics [3].

This study aimed at identifying deficiencies that could contribute to the failure of software systems by asking a series of questions, including:

- What technique (mechanism) is used in developing the software system?
- How do the end-users adopt the software?
- What is the extent of the operability of the software?
- What is the extent of readability and understanding of the software source code?
- What is the suitability of the software to the environment in which it is designed to operate? What is the extent to which the software is updated when changes occur in the environment?
- How do you take advantage of the current software design when building a new software system?
- What are the reasons that affect the life of the software system?
- What is the extent of the difficulty in maintaining the software system?
- What is the scope of the repairs in the software system in order to correct errors and modify it?

To answer these and other questions that may arise during this in-depth study, a questionnaire was distributed to the study participants, allowing the identification of reasons behind the failure of software systems to achieve the desired design goals.

5. Institutions that Served as the Study Sample

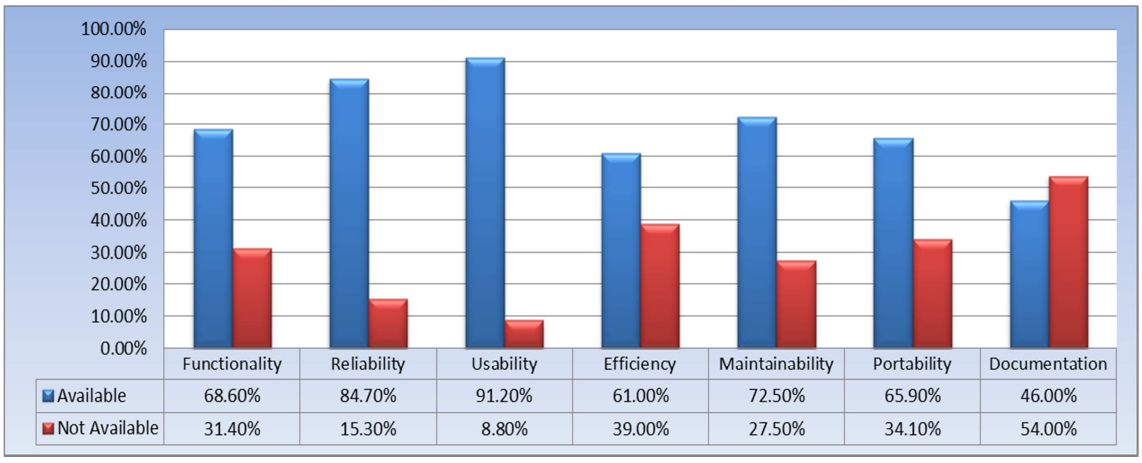
Many domestic companies and institutions that provide services using the software systems agreed to take part in this research. These companies employ specialized staff, including systems analysts, designers and programmers. Similarly, they rely on users to test the operation of the software, provide feedback and print reports. The main responsibilities of the software engineers are technical systems design, development, maintenance and support.

The questionnaires were distributed to 36 software companies, and resulted in 45 engineers and 73 users as study participants. Subsequent analyses of their responses revealed that, when determining software quality, Usability, Reliability, Portability, Maintainability, Functionality, and Portability are the most common criteria. In addition, more than half of the respondents indicated that the main drawback is the lack of standard documentation pertaining to the software systems.

6. Results

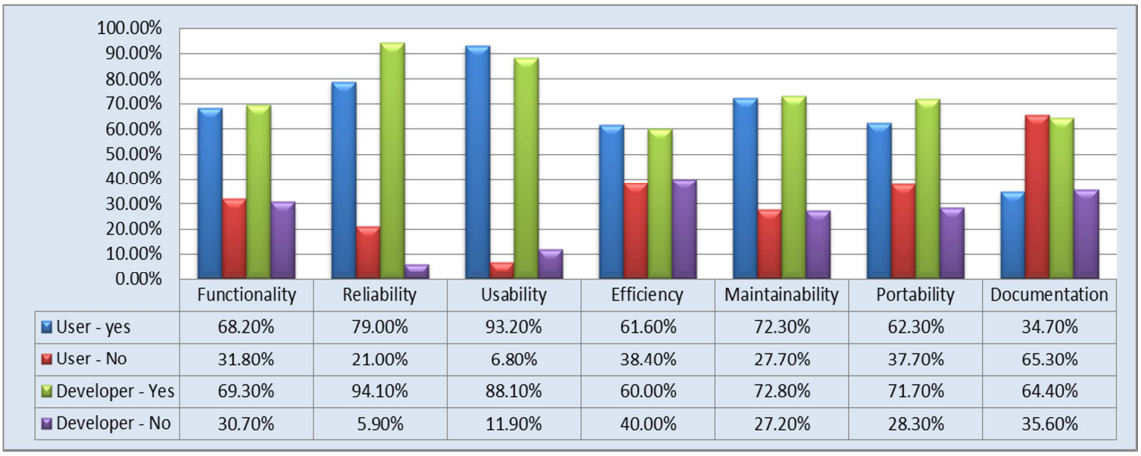
Overall, the study participants indicated that, in evaluating the software system quality, they rely on usability (91.2%), reliability (84.7%), maintainability (72.5%), functionality (68.6%), portability (65.9%), and efficiency (50%). On the other hand, most cited lack of documentation as the main obstacle to meeting these requirements. Figure 1 shows the availability of standards, as indicated by the survey results.

The views of the users and developers about the availability or lacks of standards under consideration are summarized in Figure 2.



	Functionality	Reliability	Usability	Efficiency	Maintainability	Portability	Documentation
Available	68.60%	84.70%	91.20%	61.00%	72.50%	65.90%	46.00%
Not Available	31.40%	15.30%	8.80%	39.00%	27.50%	34.10%	54.00%

Figure (1). *The adherence to the seven main software quality criteria.*



	Functionality	Reliability	Usability	Efficiency	Maintainability	Portability	Documentation
User - yes	68.20%	79.00%	93.20%	61.60%	72.30%	62.30%	34.70%
User - No	31.80%	21.00%	6.80%	38.40%	27.70%	37.70%	65.30%
Developer - Yes	69.30%	94.10%	88.10%	60.00%	72.80%	71.70%	64.40%
Developer - No	30.70%	5.90%	11.90%	40.00%	27.20%	28.30%	35.60%

Figure (2). *Software functionality with respect to the main criteria of the seven components of the ISO.*

Finally, the results shown in Table 1 reveal the application of ISO 9126 standards in the development of the software systems under study. As can be seen, 67.6% of the users and 74.4% of the developers confirmed the availability of standards in general. On the other hand, 32.4% of users and 25.6% of developers pointed to the lack of such standards in software systems under study. Finally, the results of the Chi-square analysis confirmed the existence of significant differences in the answers provided by the users and developers on the application of standards in general.

Table (1). *Distribution of the answers of the respondents about the availability of the standard ISO 9126.*

Job	No	Yes
User	%32.4	67.6%
Developer	25.6	74.4

Figure (3). *The availability of the application of ISO 9126 standards in the programming systems under study.*

7. Conclusion and Recommendations

The study findings revealed that the analyzed software systems suffer from significant weaknesses, in particular, lack of documentation criterion. Therefore, to rectify this issue, it is necessary to ensure that the developers are documenting software requirements, as well as the design phase of the system. Finally, they should produce the user manual and an operation guide for each software system.

With respect to applying quality standards according to ISO 9126, it is evident that the users and developers that took part in the survey differed in terms of their views on the application of these standards in the local software systems. Therefore, it is necessary to perform the full implementation of ISO 9126 standards to reach the optimum software utilization.

In addition, the analyzed software suffers from weaknesses in varying degrees in implementing certain criteria, such as functionality, efficiency and maintainability. This is due to the lack of understanding of these criteria, which makes it difficult to apply them correctly in the software systems. Therefore, all standards must be well understood to facilitate their application. Furthermore, we found that most respondents felt that software met the reliability criterion, likely due to the knowledge and understanding of the users and developers of this standard.

Based on the results of the study, the following recommendations can be made:

- It is necessary to perform periodical assessments of each software system.
- It is necessary to adopt a flexible model for evaluating the software, based on all the previously discussed criteria, as this allows the system strengths and weaknesses to be identified in a timely manner.
- It is necessary to standardize the evaluation criteria and disseminate this information to all stakeholders, so that the companies are able to obtain a final evaluation report that is clearly defined.
- Meetings, lectures and workshops should be held regularly, in order to disseminate and explain the concepts and foundations of the evaluation process according to the quality standards for the software application.

References

[1] Alfahd, N. , (2010) "Criteria for evaluating the software", *The national family to assess education*. Ministry of Education of Saudi Arabia.

[2] Eveleens, & Verhoef, (2010) "The Rise and Fall of the Chaos Report Figures", *IEEE Software*, January/February 2010..

[3] ISMF-ICT 3.06, (2008) "Application software for the information technology and communications", *Standards of information technology and communications, Publications of Software AG, Planet and News phone Association. The European Union*.

[4] ISO/IEC 25010, (2011) "Systems and software engineering Systems and software Quality Requirements and Evaluation (SQuaRE) System and software quality models".

[5] Maddah, S., (2009) "reparation of educational software and straightened", *Umm Al Qura University, College of Education, Kingdom of Saudi Arabia*.

[6] Nafo, A. O., (2007) "Application of audit and investigation on the local software systems", *Master Thesis, Academy of Graduate Studies, Benghazi*.

[7] Schulmeyer G., (2007) "Handbook of Software Quality Assurance", (4th ed.)

[8] Sommerville, I., (2010) "Software Engineering", (9th ed.) *ADDISON-WESLEY*.

[9] Wiegers, E., (2013) "Software Requirements" (3rd ed.), *Developer Best Practices, Microsoft Press.*

[10] Wikipedia, (2008) "Software Quality", Retrieved from http://en.wikipedia.org

Biography

Tawfig M. Abdelaziz
Department of Software Engineering, Faculty of Information Technology, University of Benghazi, , Benghazi- Libya
He is an assistant professor at the department of software engineering and a dean of faculty of Information Technology, University Of Benghazi, Libya. He is interesting in Agent systems, software Quality Assurance, software project management and Formal methods.

Mohamed M. Elammari
Department of Software Engineering, Faculty of Information Technology, University of Benghazi, Benghazi- Libya
He is a full professor and a head of the department of software engineering at faculty of Information Technology, University Of Benghazi, Libya. He is interesting in Agent systems, Object Oriented Analysis and Design, Software Project Management and Data Security.

Warda Bani
Department of Computer Science, Academy of Graduate Studies, Benghazi, Libya
She is a postgraduate student at the department of computer science at Academy of Graduate Studies, Benghazi, Libya. He is interesting in software Quality Assurance.