

Systematic Research Survey on the Impact of Software Quality Measurement Scales on User Satisfaction

Linda Susan Amos

School of Information Science & Technology (SIST)-Harare Institute of Technology, Zimbabwe

*Correspondence: Linda Susan Amos (ladylee.amos@gmail.com)

Abstract: A user is considered to be someone who would be actually using the system. The definition of a user is contextual and for any specific software project it depends on the software development methodology being developed. Satisfaction is the fulfillment of one's wishes, expectations, needs or the pleasure that is derived from using a software product therefore "user satisfaction" is a reflection of quality from different perspective than measuring defects or code structure. Measurement in software engineering makes comparison possible and when you compare you have a choice and quality is quantified in terms of values. Therefore, this paper aims to explore the impact of software measurement scales on user satisfaction.

Keywords: Software Quality, Measurement Scales, User Satisfaction

1. Introduction

Measurement in software engineering is a key aspect of good software engineering practices. According to Farooq and Nesar Ahmad everything in the world should be measurable if it is not measurable, we should make an effort to make it measurable. Measurement activities adds value to the software product and keeps us actively involved in and informed of all development phases of the process of development. It helps to make the software products more visible.

Software measurement method is a rule designed for assigning a number of identifiers to software in order to characterize it (I. I. BUCUR, 2006) [1]. There are two indicators of system success which are that software should perform as intended and the users being satisfied with the system. When this is achieved, we can then deduce that our software is of quality. The success of a software is defined in two point of views which are the users and technologist. The key elements that define success from the technologist perspective would be a software that is functionally correct, provides accurate information, operates at an optimum speed relative to the technology it is run on and meets the needs of the organization. However, from a user's perspective unless the user can use the system and use it effectively and fully. If the user cannot use the software, then it must be regarded as a failure and as a result there will be no user satisfaction achieved.

User satisfaction is an important criterion for measuring success of software and the quality of software is assessed by a number of variables. Those variables are divided into external and internal quality criteria. External is what a user experiences when running the software in its operational mode. Internal quality refers to aspects that are code dependent and that are not visible to the end users. User satisfaction is mostly determined by external quality since it is the one that is critical to the user (D. G. Kiradoo, May-June 2019) [2]. Therefore, this paper aims to answer all the questions with regards to software measurement scales to achieve quality and user satisfaction.

How to cite this paper:

Amos, L. S. (2024). Systematic Research Survey on the Impact of Software Quality Measurement Scales on User Satisfaction. *Universal Journal of Computer Sciences and Communications*, 3(1), 46–53. Retrieved from <https://www.scipublications.com/journal/index.php/ujcsc/article/view/919>

Received: March 20, 2024

Revised: April 28, 2024

Accepted: May 16, 2024

Published: May 18, 2024



Copyright: © 2024 by the author. Submitted for possible open-access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

2. Methodology Review

The process involved in conducting the review is presented in this section. The systematic literature review identifies, assesses and interprets all the papers applicable to software measurement scales, software quality and user satisfaction. The review method that was used is illustrated in fig.1 which is divided in three stages:

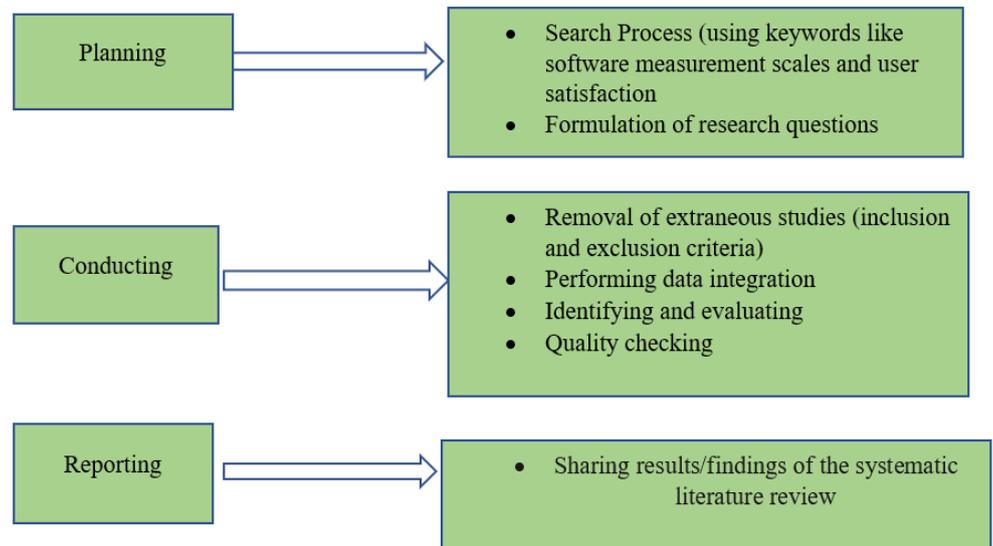


Figure 1. Systematic Review Process

3. Planning Stage for Systematic Literature Review process

3.1. Searching process

The first stage of the planning began by examining different online databases and search engines. The process of searching was split in two phases. The first phase identified Google scholar, science direct and springer. Search strings like "Software Quality", "Measurement scales", "Software Metrics and measurement for achieving quality" and "Measuring software quality to achieve user satisfaction". A lot of results were obtained using the above search string. When the first step process was completed journals or research papers that are known or talked about most were selected. In addition, the search was restricted to the years from 2001 to 2021.

3.2. Formulation of research questions from shortlisted

The following research questions were formulated (As shown in Figure 1):

Table 1. Research Questions

Number	Research questions
RQ1	Why do we need measurement in software?
RQ2	What to measure in software?
RQ3	What are the different types of measurement scales that are identified in the studies?
RQ4	How does the ISO/IEC 9126 approach relate to measurement scales in software engineering?
RQ5	What attributes can be captured in software measurement?
RQ6	How do we determine which software measurement is a good measure?
RQ7	How does software quality measurement scales influence user satisfaction?
RQ8	What are the uses of software measurements?
RQ9	What are the challenges of software measurement?

4. Conducting Stage for Systematic Literature Review Process

This stage involves the activities of removing extraneous studies from the research papers that were examined. A questionnaire was designed to score the selected studies as per each research question in [Table 1](#). A table was used to show the division of the shortlisted or identified studies as per year and source of publication.

4.1. Removal of extraneous studies

Research papers or studies were arranged in an ordinary form, and studies that were irrelevant were removed from the review. Using the research question table, the inclusion criteria were set in order to select those studies that had some content on some aspects of software measurement scales and quality.

The Inclusion Criteria that were used:

- Empirical studies that identified different types of measurement scales.
- Empirical studies that use software measurement to identify quality.
- Empirical studies that identify the ISO/IEC 9126 approach to software quality.
- Empirical studies that investigate on software quality and user satisfaction.

The Exclusion criteria that were used:

- Studies that did not focus on software quality and measurement models or scales
- Studies that did not answer the research questions that was formulated in [Table 1](#).
- Studies that do not belong in the year range of 2001 -2021

4.2. Criteria used for quality checking

A questionnaire was designed to remove papers that were irrelevant .10 studies were extracted using search engines and out of which 8 were identified that meet the inclusion criteria in section 4.1. [Table 2](#) below shows list of questions that were used for quality checking that scored the studies on 0 to 1 scale.

Table 2. Quality checking criteria

Number	Quality checking question
Q1	Does the study define software quality and measurement scales?
Q2	Whether the study clearly define software metrics and measurement for achieving quality
Q3	Whether the paper or study defines any goal or objective?
Q4	Whether appropriate results were provided in the study?
Q5	Whether the study mentions the limited and advantages of the research
Q6	Whether the study mentions how to achieve user satisfaction in software engineering?

4.3. Data Extraction

A database consisting of different features such as publication year, name of study, authors, publication source, objective of the study, results, tools and techniques was prepared.

Three attributes were majorly selected that answered the research question and listed in [Table 3](#) below:

Table 3. Attributes

Attribute	Research Question
Software Quality	RQ4
Measurement Scales	RQ3
User Satisfaction	RQ7

4.4. Distribution of the research papers

Studies were divided into 3 categories upon their source of publication. The source that were considered are:

- Journals
- Conference papers
- Others such as symposiums, different book chapters that relate to the study, workshops etc.

Table 4 below shows distribution of papers as per year of publication.

Table 4. Distribution of papers per year

Year	Number of papers per year
2001-2005	4
2006-2010	3
2011-2015	2
2016-2021	1

5. Reporting phase

5.1. Reporting of research questions

The studies were assessed or analyzed on the basis of research questions that were raised. Solutions to the research questions identified and a summary of facts are as follows:

RQ1 – Why do we need to measure in software?

Measurement is a mapping from the empirical world to the formal, relational world. A measure is the number or symbol assigned to an entity in order to characterize an attribute (Sheikh Umar Farooq, 2011) [3]. When we want to compare two things we need to measure. Measuring makes comparison possible and when you compare you have a choice. Quality is quantified in terms of values and if we are able to measure software, we are able to make choices and decisions.

RQ2 – What to measure in software?

What to measure in a software depends on the needs, purpose or the context (M.C. Lee, 2014) [4]. There are different things that we can measure in software some of which includes but not limited to:

- Size e.g. Lines of Code (LOC)
- Togetherness of modules components e.g. modularity or high cohesion
- Dependence of modules on each other
- Code complexity or organization of code
- Defects discovered per lines of code
- Time spent on different processes
- Number of requirements

RQ3 – What are the different measurement scales identified in the studies?

There five measurement scales used in the Measurement Theory which are applicable to software as well. Measurement scales are hierarchical, and each level scale possesses all the properties of lower scales as indicated in their order below. It is also possible

to convert the higher-level scales into the lower level ones (i.e. ratio to interval or ordinal or nominal; or interval to ordinal or nominal; ordinal to nominal (Sheikh Umar Farooq, 2011) [3].

Nominal Scale

Items in nominal scale are assigned to groups or categories. There is no number or quantitative information that is generated. Also, no ordering is implied. In software engineering for example, classifying recursive or non-recursive programs or programs that uses portable and non-portable features therefore, it is a qualitative difference. Nominal scale is the most primitive form of measurement. A scale is nominal if it divides the set of entities into categories with no particular ordering among them (Stevens, 1946) [5]. If we are able to classify two or more thing into different categories or groups, we can derive a sense of measure and if we are able to measure there is quality and if there is quality there is user satisfaction.

Ordinal Scale

A scale is ordinal if it divides set of entities into different categories that are ordered according to some order. There is no quantitative measurement. The higher number in an ordinal scale represents higher value e.g. programmer skill (low, medium, high) but you can't say whether high in programmer skill is 1/3 lower. In software engineering the Capability Maturity Model (CMM) phases is an example of ordinal scale e.g.:

- Initial phase
- Repeatable phase
- Defined phase
- Managed phase
- Optimized phase

Or we can measure using ordinal scale how often a software fails e.g. daily, weekly, monthly, yearly or never.

Interval Scale

The scale captures information about the intervals that separate classes. In an interval scale, the exact difference between two values is the basis for meaning statement (Sheikh Umar Farooq, 2011) [3]. In an interval scale the scale that separates classes is known e.g. based on value, you know exactly when the item crosses one class and goes to another. The classes in an interval scale are ordered. Operations like addition and subtraction is possible in an interval scale but you cannot use multiplication and division. A good example of an interval scale is measuring the programmer capability between:60th and 80th percentile of population or we can measure the size in terms of lines of code (LOC) and say that the Lines of Code are in the range 80-200 Lines of Code.

Ratio Scale

In ratio scale ordering is preserved, the size of intervals between entities as well as ratios. A good example of ratio scale is when comparing the time that two projects took to complete, we can say that Project A took twice or thrice as long as Project B.

Absolute Scale

This is done by counting entities in the set. It is the most informative in the measurement scale hierarchy. All arithmetic operations are meaningful in absolute scale measurement. There is only one way to measure in absolute scale that is counting. The length of program when measured as statement count (LOC) is not the absolute scale as you can measure it in different ways e.g. lines or words but (LOC) as statement count is absolute scale when you measure number of lines of code, number of times of failures, number of project engineers involved. A very good example of absolute scale of

measurement is done by counting number of failures observed during integration testing that can be measured only by counting the number of failures observed.

RQ4 – How does the ISO/IEC 9126 approach relate to measurement scales in software engineering?

The approach to software quality is best exemplified by fixed quality models such as ISO/IEC 9126 (As shown in Figure 2):

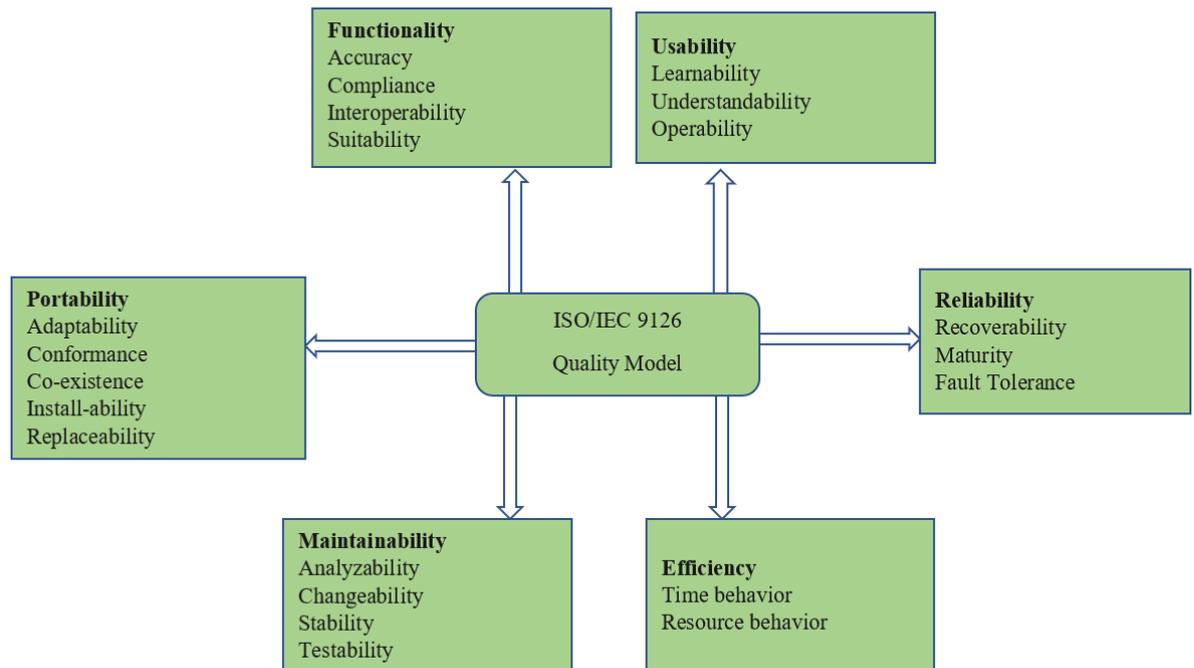


Figure 2. ISO/IEC 9126

A Software is considered of quality if consists of the following attributes: Portability, Maintainability, Efficiency, Usability, Reliability and Functionality. User satisfaction is a reflection of quality from different perspective than measuring defects or code structure (D. G. Kiradoo, May-June 2019) [2]. Therefore, measurement scales are Measurement Theories that are applicable to software and all the five scales can be applied differently to measure the software quality attributes.

RQ5 -What attributes can be captured in software measurement?

The attributes that can be measured in a software includes but not limited to the following:

- Software quality
- Effort or resource estimates
- Progress of software development
- Complexity of difficulty of the software

RQ6- How do we determine which software measurement is a good measure?

What to measure in software depends on the goals of your measurement and what you want to measure that is your needs, context or the purpose so, what to measure varies depending on what you are trying to achieve or solve.

RQ7- How does software quality measurement scales influence user satisfaction?

User satisfaction is a reflection of software quality from different perspective than measuring defects and or the code structure which means that satisfaction of the user in software is only achieved when the software is of quality and quality of software is determined by making use of different Measurement Theories in software engineering.

RQ8- What are the uses of software measurement?

Measurements in software can be used to determine the following:

How good is the design of the software-Bad designs are mostly hard to understand, implement, maintain, and evolve.

How complex is the code-Complex code is mostly hard to understand test, debug and change.

How much effort will be required? -The effort required to develop the software that is (design, implement, test, debug and deploy). Software measures, help in decision making in various life cycle phases. The software might be corrected, but the design might be complex e.g. we can have 25 designs of the same software, so we should be able to measure and choose the best design out of the 25.

RQ9 – What are the challenges of software measurement?

Measurement of software is challenging in nature as shown in [Figure 3](#) below:

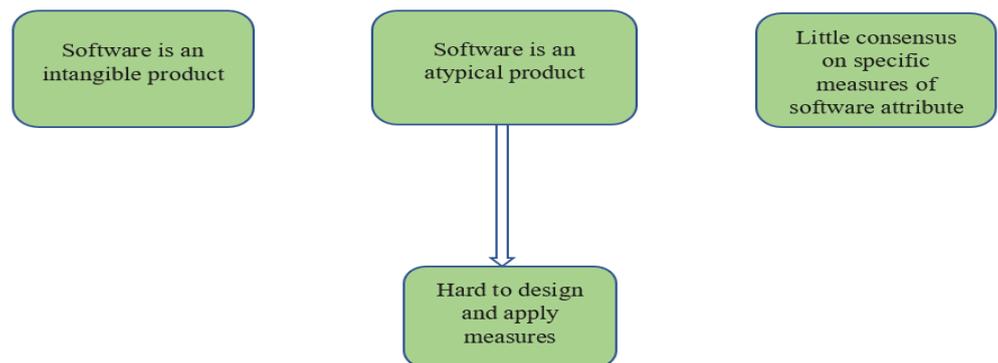


Figure 3. Challenges in the design of software measures

Software is an intangible product meaning that it is atypical in nature when compared to other industrial products, in that it varies greatly in terms of size, complexity, design techniques, test methods and applicability.

6. Discussion

User satisfaction is an important factor for measuring the success of software of which the success of software is defined in two point of views one for users and the other one for technologist. Technologist measures the success of software by its functionality, accurate information that the software provides and the speed relative to technology it operates on. Users measures the success of software by their ability to use it effectively and fully (Adel Khelifi, 2004) [6]. What to measure in software depends on the needs, purpose or the or the context. The five measurement scales used in measurement theory which are applied to software engineering are nominal scale, ordinal scale, interval scale, ratio scale and absolute scale.

7. Conclusion

What to measure in software differs on the context and the purpose or the goals of what you are trying to achieve or measure in a software. Software measurement scales play a crucial role in software engineering as they aid us to deliver high quality and reliable software. If quality is achieved in software, we can then talk of user satisfaction as it is a reflection of quality of the software from different perspective other than measuring defects and the structure of the code (Moshe Zviran, 2003) [7]. Unfortunately, there are challenges in the design of software measures as software is atypical in nature and it has little consensus on specific measures of software attributes.

References

- [1] I. I. BUCUR, "On Quality & Measures in Software Engineering," *Journal of Applied Quantitative Methods*, vol. Volume 1, no. *Journal of Applied Quantitative Methods*, 2006.
- [2] D. G. Kiradoo, "Software Engineering Quality to Enhance the Customer Satisfaction Level of the Organization," *International Journal of Advanced Research in Engineering and Technology (IJARET)*, vol. Volume 10, pp. 297-302, May-June 2019.
- [3] S. M. K. Q. & N. A. Sheikh Umar Farooq, "Software Measurements and Metrics: Role in Effective Software Testing," *International Journal of Engineering Science and Technology (IJEST)*, vol. Volume 3, 2011.
- [4] M.C. Lee, *British Journal of Applied Science & Technology*, vol. Volume 4, 2014.
- [5] O. t. T. o. S. o. Measurement, "On the Theory of Scales of Measurement," *American Association for the Advancement of Science*, vol. Vol. 103, pp. 677-680, 1946.
- [6] A. A. L. B. Adel Khelifi, "A System of References for Software Measurements with ISO 19761 (COSMIC-FFP)," *A System of References for Software Measurements with ISO 19761 (COSMIC-FFP)*, 2004.
- [7] Z. E. Moshe Zviran, "Measuring IS User Satisfaction: Review and Implications," *Communications of the Association for Information Systems*, vol. Volume 12, pp. 81-103, 2003.