

A Model for Quantifying Usability Metrics: An Effective Approach

¹Mohd.Haleem
Computer Science & Engineering
Integral University
Lucknow, India
md.haleem06@gmail.com

²Prof (Dr)Md.Rizwan Beg
Computer Science & Engineering
Integral University
Lucknow, India
rizwanbeg@gmail.com

³Syed Umar Amin
Computer Science & Engineering
Integral University
Lucknow, India
syed.umar.amin@gmail.com

Abstract-Quality comprises all characteristics and significant features of a product or an activity which relate to the satisfying of given requirements. Software quality metrics are a subset of software metrics that focus on the quality aspects of the product, process, and project. Quality in use (usability) is recognized as an important quality factor for interactive software systems. This paper presents a quality model for the quantification of usability matrices in software quality models and discusses the current approaches to usability metrics and then reviews existing usability standards and models while highlighting the limitations and complementarities of the various standards. Paper further explains how these different models can be unified into a single hierarchical model of usability, and proposes a new approach for quantifying usability of software product. It proposes a quality model as a framework for specifying and identifying quality in use components, which include different factors, criteria, metrics and data defined in different Human Computer Interface and Software Engineering models.

Keywords: Usability, Software engineering quality models, quality in use, usability metrics, Human Computer Interface.

I. INTRODUCTION

Several studies have reported the benefits of usability in the software development lifecycle (Mayhew, 1999 Landauer, 1995). Usability is important not only to increase the speed and accuracy of the range of tasks carried out by a range of users of a system, but also to ensure the safety of the user (Repetitive Strain Injury etc.). Computer magazine software reviews now include 'usability' as a ratings category [1]. There are different no of models and standards have been proposed for quantifying and assessing usability within the Human Computer Interaction (HCI) and the Software Engineering (SE) communities. Example includes the ISO/IEC 9126-1 (2001) standard [7], which identifies usability as one of six different software quality attributes; the ISO 9241-11 (1998) standard, which defines usability in terms of efficiency, effectiveness, user satisfaction, and whether specific goals can be achieved in a specified context of use; and Directive 90/270/EEC of the Council of the European Union (1990) on minimum safety and health requirements for work with computers. Usability

however, has not been defined in a consistent way across the above mention standards or for other models described. Most of these above standard or models do not include all major aspects of usability. Also they are not properly integrated into current software engineering models. However quality in use - commonly usability or user perspective of software quality - has received widespread attention within both the software engineering and human computer interaction (HCI) communities, there are few integrated software quality models that are used for specifying and measuring our current meaning of usability (McCall, 1977; Boehm,1978). The HCI community has developed different models for specifying or measuring usability. One of their weaknesses is that they are not well integrated within the software engineering models [2]. Consequence of these weaknesses is that most of the software developers do not apply any particular model correctly in the evaluation of usability. Because there are no proper clear guidelines about how various definitions of usability factors, rules, and criteria are related and how to select or measure specific aspects of usability for specific software product. All the characteristics that are needed for a product to meet its predefined usability goals, should be defined in a good quality model. The list of characteristics should include efficiency, learnability, human satisfaction and safety as well as the measurable attributes (metrics)[2]. Also a good quality in use model should explicitly define the relationships that actually can exist between the characteristics and the measurable attributes. Other requirements of a good quality in use model may include following characteristics: These characteristics are easy to understand but most of the time difficult to measure.

Functionality- An efficient quality in use model should be flexible in such a way it can be used at various steps of the software development lifecycle [2].

Flexibility-An efficient quality in use model should be flexible in such a manner so that it can be used at various steps of the software development lifecycle [2].

Usability - The model should be easy to understand by the user.

Learnability- The model should be easy to learn by the users those are involved in the lifecycle of software development including software quality engineers, usability engineers as well as user interface developers who are not necessary familiar with usability.

Automated support - A quality model should be supported by tool that can facilitate the process of gathering usability requirements as well as testing/predicting usability. The tool should also mediate the communication between usability engineers and software engineers. This is one of the major weaknesses of the current usability models that software engineering approaches can improve [2].

Some of the other main reasons to outline quality model are:

- 1) Effective evaluation of Expert-based usability.
- 2) By providing a basis for understanding and comparing various usability metrics though reduce the costs of usability testing.
- 3) Provide clear communication about usability measurement between software developers and usability experts.
- 4) By adopting good practices for usability measurement that are easily accessible to software developers who may not have strong backgrounds in usability engineering.

II. DEFINING: USABILITY

Usability has been defined in a different manner in that makes it a confusing concept. To explain usability concept, various definitions of usability from three different standards are listed here:

- 1) “A set of attributes that bear on the effort needed for use and on the individual assessment of such use, by a stated or implied set of users” (ISO/IEC 9126, 1991)[5]
- 2) “The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use” (ISO 9241-11,1998)
- 3) “The ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component” (IEEE Std.610.12-1990)

There are also some more definitions included in some standards or given by different authors including more specific attributes such as facts, aspects, factors, of usability.

A. Usability attributes of various standards or models

There are some of the following usability attributes discussed in Table-1 of various models and standards.

III. ANALYSIS AND OBSERVATION OF EXISTING USABILITY STANDARDS AND MODELS

Usability measurement is not generally related with similar research in software engineering measurement: relating data or concern quality attributes such as reusability or reliability (Curtis, 1980). In this part, we will try to analyze existing usability standards or models and their contributions.

A. Usability in ISO standards

In human computer interaction community, many definitions of usability and frameworks for its specification and measurement exist. Many standards that address explicitly usability are also available (Bevan95). Among them, we list the following:

Constantine & Lockwood (1999)	Efficiency in use	Learnability	Rememberability	Reliability in use	User satisfaction
ISO 9241-11 (1992)	Efficiency				Satisfaction
Schneiderman (1992)	Speed of performance	Time to learn	Retention over time	Rate of errors by users	Subjective satisfaction
Nielsen (1994)	Efficiency of use	Learnability	Memorability	Error/Safety	Satisfaction
Preece et al. (1994)	Throughput	Learnability		Throughput	Attitude
Shackel (1991)	Effectiveness	Learnability			Attitude

Table 1 Usability attributes in various models Ref [1]

1) **ISO 9241-11(1988)** - This standard defines usability as a high level quality objective: “The extent to which, a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use”. This model suggests different metrics. The major limitation of this standard as quality model is that it is so abstract, and the relationships between metrics and usability objectives are not explicitly defined.

2) **The ISO/IEC 14598-1(1999)** - This standard suggests a model for studying and measuring quality in use from the internal software attributes in a particular context of use. [5]

3) **The ISO/IEC 9126(2001)** - This standard breaks software quality into six broad categories functionality, reliability, usability, efficiency, maintainability and portability. These can be further broken down into sub characteristics that have measurable attributes. [7]

Software quality attributes are the cause - quality in use, the effect. (Or at least should be) The objective of quality in use is to achieve quality in software product. The user’s needs in terms of usability goals are expressed as a set of requirements for the behaviour of the product in use (for a software product, the behaviour of the software when it is executed.) These requirements will depend on the characteristics of each part of the overall system including hardware, software and users. The requirements should be expressed as metrics that can be measured when the system is used in its intended context, for instance by measures of effectiveness, efficiency and satisfaction. This model is incomplete that it only addresses the software quality attributes and their impact on usability, this

model can be used as a foundation for an integrated model that combines usability and software engineering models.

B. Usability in context with software quality models

The following are the most important models that introduce usability as one of the software quality factors:

1. **Boehm model** - This model is one of the first quality models for software quality (Boehm, 1978). He proposed a multilevel hierarchy or a tree of software criteria. He suggested that a software product is usable if it is portable, maintainable and at the next level he decomposed those criteria to the other factors for example, he decomposed maintainability into testability, understandability, and modifiability.[3]

McCall model - This model [6], is also called GE model (Fenton, 1997) or FCM (Factor, Criteria and Metric) was proposed by McCall in 1977). It is based on three uses of a software product, i.e. product revision, product operations and product transition. For every one of those uses, this model defines different factors that describe the external view of the system, as it is perceived by end-users. Each factor is further decomposed into criteria that describe the internal view of the product as perceived by software developer. Criteria could be common between different factors.

IEEE 1061 (1998)- Standard on Software Quality Metrics Methodology. This standard provides a methodology for establishing quality requirements as well as identifying, implementing, analyzing and validating process and product of software quality reviews existing usability standards and models metrics [4]. This methodology applies to all software at all phases of any software life cycle. This standard does not prescribe any specific metric. The model suggests a hierarchy including different levels for quality factors, quality sub factors and metrics as well.

IV. REASON FOR OUTLINING QUALITY MODEL

There are many reasons for outlining a model for quantifying usability. Quality models that discussed above have some common limitations. They all are vague in defining the lower-level usability metrics that are required to obtained satisfactory measures of higher-order software quality factors. There is also relatively little information about the method to select a set of usability factors or metric [1]. These problems can easily solve through a better tool support for use by individuals who are given the responsibility to evaluate usability. Second, most of the software quality models just described, including the ISO/IEC 9126 standard, are static [2]. That is, none of these models really describes the relation between phases in the product development cycle and appropriate usability measures at specific project milestones. It is important to be able to relate software measures to project tracking and to target values at the time of delivery of the software. Also, none of these models give any clear guidelines concerning the application of usability measures and attributes in the identification and classification of risk (Hyatt and Rosenberg, 1996). Third, it can be rather

difficult to apply usability standards in practice, that it is difficult to decide exactly how to measure the usability of a particular application. Specifically, when it is not clear how usability factors, criteria, and metrics defined in different standards or models are related or whether one set of metrics may be more advantageous than others [3]. The proposed model should support the relations among sets of factors, criteria, and metrics, again in a consistent and clear way. The model should also be flexible and generic enough in order to help developers or testers who may not be already familiar with usability metrics to create a concrete, step-by-step measurement plan for different types of applications.

V. PROPOSED QUALITY IN USE MODEL

We propose quality in use model that is a hierarchical model like the other software engineering models discussed in the previous section. The difference is that, it decomposes different levels of usability, called factors, criteria and metrics. Our quality in use model follows the IEEE 1061 (1998) standard (Software Quality Metrics Methodology), which outlines methods for establishing quality requirements as well as identifying, implementing, analyzing, and validating both process and product quality metrics (Schneidewind, 1992; Yamada et al., 1995). The main motto for this model is to provide a consistent model for usability factors, criteria, and metrics for educational and research purposes. This model serves as a foundation under which other models for usability measurement can be derived. It basically combines the existing models.

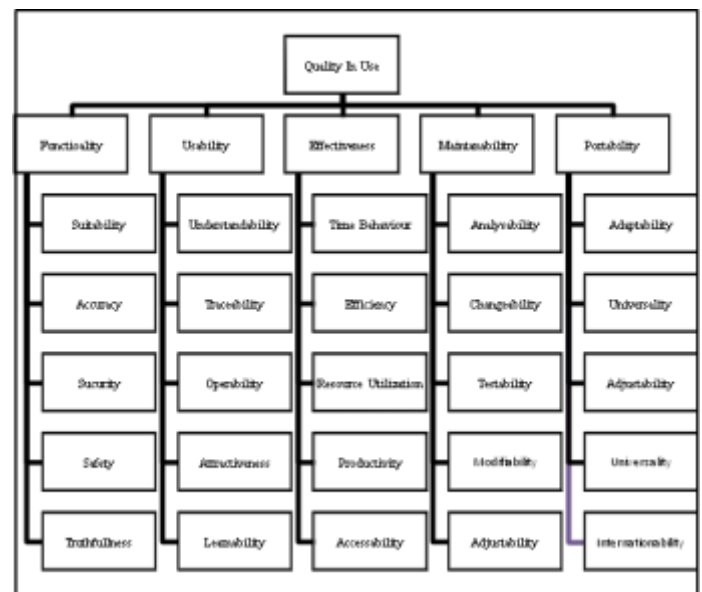


Fig 1: Quality in Use Model

A. Usability model, context of use attribute

Usability is generally a relative measure of whether a software product enables a particular set of users to achieve specified goals in a specified context of use [2]. This means that usability

can vary from one user to another one and in general with the context of use.

This includes user profiles (i.e., who are the users), task characteristics, hardware (including network equipment), software, and physical or organizational environments. Therefore, the measurement of usability requires that we should know in advance the characteristics of the target users and the kinds of tasks they will carry out with the system. Lack of knowledge about either users or tasks may lead to the inability to formulate a realistic usability measurement plan. It may also result in a product that only by chance happens to meet user needs.

Table 2 are context of use attributes from the ISO 9241-11 (1998) standard. These attributes concern characteristics of users and tasks and of technical, physical, social.

A. Usability Factors

The following are factors that are included in QUIM (Donyaee and Seffah, 2001):

Effectiveness: The degree of accuracy and completeness with which the user achieves a specified task in a certain context.

Efficiency: The amount of resources expended in relation to the accuracy and completeness with which the user achieves a goal.

Satisfaction: Freedom from discomfort and positive attitude towards the use of the software product.

Internationality: The degree to which software can be used for the global marketplace, taking into account variations in regions, population stereotypes, languages, and cultures.

Accessibility: The degree to which software can be used comfortably by a wide variety of people, including those who require assistance technologies like screen magnifiers or voice recognition.

Productivity

Safety

In our quality in use model we have included some other important usability factors such as:

Learnability: Capability of the software product to enable users that they can productively use the software product right away and can quickly learn then other new functionalities.

Trustfulness: or the faithfulness a software product offers to its users.

Universality: This concerns whether a software product accommodates a diversity of users with different cultural backgrounds.

Adjustability: capacity of a system to be adapted or to adapt itself to the context.

Portability: The capacity of a system to be work in different platforms.

Modifiability

B. Usability Criteria

Basically criteria are the sub-factors and these sub factors can be measured with the help of a set of metrics. The following are examples of criteria.

1. Attractiveness: Indicator expressing the extent of which user likes the software during the operation.

2. Consistency (Lin, 1997): Attributes that bear on the uniformity of user interface.

3. Minimal Action (Lin, 1997): The extent to which user needs to take minimal effort to achieve a specific task.

4. Minimal Memory load (Lin, 1997): The extent to which user needs to keep minimal amount of information in mind to achieve a specified task.

5. Completeness: The extent to which the user can complete a specified task.

Component	Relevant data
Users characteristics	Psychological attributes including cognitive style (e.g., analytic vs. intuitive, attitude towards the job, motivation to use the system, habits and motor-sensory capabilities) Knowledge and experience including native language, typing skills, education and reading level, system experience (e.g. knowledge of computer and OS), task experience (e.g., knowledge of the domain), application experience (e.g., knowledge of similar applications)
Task characteristics	Frequency Duration and importance Task flexibility/pacing Physical and mental demands Complexity as perceived by the user Task structure (e.g., highly structured vs. unstructured) Task flow including input/start conditions, output/finish conditions, and dependencies Relation to business workflow
Technical environment characteristics	Hardware capabilities and constraints Network connection Operating system Supporting software
Physical environment	Noise level, privacy, ambient qualities, potential health hazards, and safety issues
Organizational environment	Structure of the operational teams and the individual staff members' level of autonomy Work and safety policies Organizational culture Feedback to employees on job quality
Social environment	Multi- or single-user environment Degree of assistance available Interruptions (e.g., disturbing environment)

Table 2 Examples of context of use attributes from ISO 9241 11 (1998)

- 6. Time behaviour:** Capability to consume appropriate task time when performing its function.
- 7. Flexibility:** Whether the user interface of the software product can be tailored to suit users' personal preferences.
- 8. Operability:** Amount of effort necessary to operate and control of a software product.
- 9. Resource utilization:** Capability to consume appropriate amounts and types of resources when the software performs its function (ISO/IEC 9126-1, 2001)[5].
- 10. Attractiveness:** Capability of the software product to be attractive to the user (e.g., through use of color or graphic design; ISO/IEC 9126-1, 2001)[5].
- 11. Likeability:** User's perceptions, feelings, and opinions of the product (Rubin, 1994)
- 12. Simplicity:** Whether extras elements can be eliminated from the user interface without significant information loss.
- 13. Privacy** Whether users' personal information is appropriately protected.
- 14. Security:** Capability of the software product to protect information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access (ISO/IEC 12207, 1995)

C. Usability Metrics

The IEEE metrics standard defines a software metric as “a function whose inputs are software data and whose output is a single numerical value that can be interpreted as the degree to which the software possesses a given attribute that affects its quality “ [4]. By reviewing existing usability standard and model we have identified more than 100 usability matrices in which some of them are countable data and some are simple functions .Some examples of previously defined validated metrics, are general enough, so they could be applied to most software and context of use. For detailed explanation and examples of calculation, one may refer to the mentioned reference in every case.

Essential Efficiency (EE; Constantine and Lockwood, 1999):

$$EE = 100 \times S_{\text{essential}} / S_{\text{enacted}}$$

Where S_{enacted} $S_{\text{essential}}$ = the number of user steps in the essential use case narrative

S_{enacted} = the number of steps needed to perform the use case with the user interface design (rules for counting the number of enacted steps has come in the reference)

Estimates how closely a given user interface design approximates the ideal expressed in the use case model.

Task effectiveness(Bevan and Macleod 1994): Task effectiveness is calculated by given formula

$TE = \text{Quantity} \times \text{Quality} / 100$ Where Quantity is the proportion of the task completed and Quality is the proportion of the goal achieved

Task Concordance (Constantine and Lockwood, 1999):

Measures how well the expected frequencies of tasks match their difficulty, favours a design where more frequent tasks are made easier.

$$TC = 100 \times D/P$$

$$P = N(N - 1)/2$$

Where

N = is the number of tasks being ranked,

D = is the discordance score, i.e., the number of pairs of tasks whose difficulties are in the right order minus those pairs whose difficulties are not in right order

VI. CONCLUSION AND FURTHER RESEARCH

We have discussed many standards or conceptual models that have described in the beginning of this paper concerning usability as main aspects of software quality. Further we have also discuss the similarities, differences, and limitations of different models for analysing, specifying and measuring usability, considering the software engineering quality models with the human computer interaction standards for usability.

We also proposed a quality in use model that integrate usability factors, criteria, metrics, and data mentioned in various standards or models for software quality.

The model can be helpful in creating usability measurement plans where specific metrics are identified and defined given various higher-level usability goals, such as efficiency, satisfaction, and learnability. These plans include information about how to collect the data needed to calculate metrics associated with the overall usability goals. One of the major goals of developing this model is to keep it simple, easy and understandable as possible, in such a way that it can reduce software development risks in a less expensive manner.

In future this quality in use model can be modified in order to expand its use to broader field of engineering.

REFERENCES

- [1] Ahmed Seffah, Mohammad Donyaee, Rex B, Kline Harkirat K. Padda. Usability measurement and metrics: A consolidated model
- [2] Donyaee, M., Seffah, A., OUIIM: An Integrated Model for Specifying and Measuring Quality in Use, *Eighth IFIP Conference on Human Computer Interaction*, July 9-13, 2001, Tokyo, Japan.
- [3] Boehm, B.W., Abts, C., Brown, W., Chulani, S, Clark, B.F., Steece, B., Brown, A.W., Chulani, S., and Abts, C. 2000. Software, Prentice-Hall.J. 1978. Characteristics of Software Quality, New York: North Holland.
- [4] IEEE Std. 1061 (1998), IEEE Std. 1061 (1998), Software Quality Metrics Methodology.
- [5] ISO 9126(1991): Software Product Evaluation – Quality Characteristics and Guidelines for Their Use.
- [6] J. A McCall, P. K, Richards and G. F. Wallers, “Factors in software quality “, Griffiths Air Force Base, N. Y: Rome Air Development Center Air Force Systems Command, 1977.
- [7] ISO / IEC, IS 9126-1, “Software Engineering – Product Quality – Part 1: Quality Model”, Geneva Switzerland: International Organization for Standardization, 2001.

