Development of a quality design framework for usable User Interfaces

Dr. S. Qamar Abbas
Deptt. of CSE
Integral University,
Lucknow, India

Ms. Ankur Aggarwal
Research Scholar, Deptt.of CSE
Integral University,
Lucknow, India

Abstract:
The most important aspect of modern interactive computer systems is the level of support they provide for the underlying human activity. This level of support is encompassed in the user interface with which the user interacts with the system. User interface usability is an important factor but is often considered only in terms of the ease of use of the user interface. In this paper, a new framework is designed for the user interface which will evaluate the quality of design before the final design is formalized. This paper emphasizes on the fact that usability as an objective is synonymous with the quality of use. Usability as an attribute must be designed into a product and must be considered as the highest level quality objective. A framework is proposed to establish the dependence of various parameters of usability between the user and the product resulting in quality of use in a context which in turn represents the usability of the product from user's perspective.

I. INTRODUCTION

In most major systems development companies, the basic organizational structures and development processes were defined in an earlier era, when the dialogue between computer systems and computer users did not have to be considered. Until 15 years ago, most computer system users were engineers and programmers. Developers were designing systems for their own use or for other technically proficient users. They felt no need to seek "user participation" in design. Now, however, computer use has spread to workplaces that are very unlike engineering laboratories. To bridge the widening gulf between the developer and user environments requires greater effort.

The challenge of designing better interactive systems has not gone unnoticed in software engineering. Boehm observes that the dominant waterfall model of development "does not work well for many classes of software, particularly interactive end-user applications."[1] His proposal, a "spiral model" of software development, incorporates user involvement, prototyping, and iterative design. Yourdon recently wrote, "the first, and by far the most important, player in the systems game is someone known to systems analysts as a *user*."[2] The spiral model is not yet widely used and as Boehm notes, it may be difficult to apply in some contexts. Similarly, Yourdon's observation has not been fully translated into practice. The software methods that are employed widely today were developed before interactive end-user applications became important. They do not provide for an early and continual focus on users quite the contrary. Traditional structured analysis relegates the task "establish man-machine interface" to one sub-phase of system development [3].

II. QUALITY USER INTERFACE

The objective of any User Interface developer should be to design and implement quality user interfaces. "Quality user interface" is defined as any user interface that is intuitive, easy to use and allows the user to maximize their efficiency and effectiveness when using it. Gravin has given a different approach to quality i.e. user perceived quality which is a combination of product attributes which provide the greatest satisfaction to a specified user [4]. ISO 8402 defines quality as: the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs. This definition is in terms of the characteristics of a product. To the extent that user needs are well-defined and common to the intended users it implies that quality is an inherent attribute of the product. However, if different groups of users have different needs, then they may require different characteristics for a product to have quality, so that assessment of quality becomes dependent on the perception of the user. User-perceived quality is regarded as an intrinsically inaccurate judgment of product quality. For instance, Garvin observes that "Perceptions of quality can be as subjective as assessments of aesthetics". Usability embraces user-perceived quality by relating quality to the needs of the user of an interactive product. *Quality of use* is the extent to which a product satisfies stated and implied needs when used under stated conditions. This moves the focus of quality from the product in isolation to the particular users of the product, the tasks and the context in which it is used. The quality of use is determined not only by the product, but also by the context in which it is used: the particular users, tasks and environments. The quality of use (measured as effectiveness, efficiency and satisfaction) is a result of the interaction between the user and product while carrying out a task in a technical, physical, social and organizational environment.

III. USABILITY

Usability is the capacity of the software product to be understood, learned, used and attractive to the user, when used under specified conditions. Usability is a general term that encompasses everything having to do with "ease of use." That is, how easily people can use any product's controls or displays such as a: tool, computer display, automobile, aircraft, etc. Usability also refers to the study of methods, measurement, and principles of a product's efficiency, elegance, and usefulness. In the computer industry, usability often refers to the ease of use in terms of the human-computer interaction. The clarity, intuitiveness, seamlessness, and elegance of an application or website interface design. The idea behind usability is to design products keeping the user in mind. Putting the user first in the design process results in greater efficiency, learning time, and satisfaction. The goal of optimized usability is to make a product easy to understand, easy to use, and easy to learn. The outcome of good usability is a greater likelihood of user acceptance. User acceptance is often the difference between a product's success or failure in the marketplace. Users can often reject a well engineered product with great functionality if they are unable to understand, learn, and easily use that product. Similarly the focus of the evaluation (element to be varied) may be a complete computer system, the complete software, a specific software component, or a specific aspect of a software component. Any relevant aspect of software quality may contribute to quality of use, but for interactive software ease of use is often a crucial issue. Quality of use thus provides a means of measuring the usability of a product,

Most computer software in use today is unnecessarily difficult to understand, hard to learn, and complicated to use. Difficult to use software wastes the user’s time, causes worry and frustration, and discourages further use of the software. Benefits to the employer include:

¥ Usable software increases productivity and reduces costs. Difficult to use software is time consuming to use, and not exploited to full advantage as the user may be discouraged from using advanced features. Difficult to learn software also increases the cost of training and of subsequent support.

¥ Usable software increases employee satisfaction. Difficult to use software reduces motivation and may increase staff turnover.

¥ In Europe, employers have an obligation to meet the requirements of the Display Screen Equipment Directive (CEC 1990) which requires software in new workstations to be "easy to use" and to embody "the principles of software ergonomics" [5].

Usability is the quality of use in context. The effectiveness, efficiency, safety, flexibility and satisfaction are the usability parameters with which specified users can achieve specified goals in specified environments. However the attributes that a product requires for usability depend on the nature of the user, task and environment.

Effectiveness:

Effectiveness refers to correctness, accuracy and completeness. Error-free completion of tasks is important in both business and consumer applications. Measures of effectiveness relate the goals or sub-goals of using the system to the accuracy and completeness with which these goals can be achieved. For example if the desired goal is to transcribe a 2-page document into a specified format, then accuracy could be specified or measured by the number of spelling mistakes and the number of deviations from the specified format, and completeness by the number of words of the document transcribed divided by the number of words in the source document.

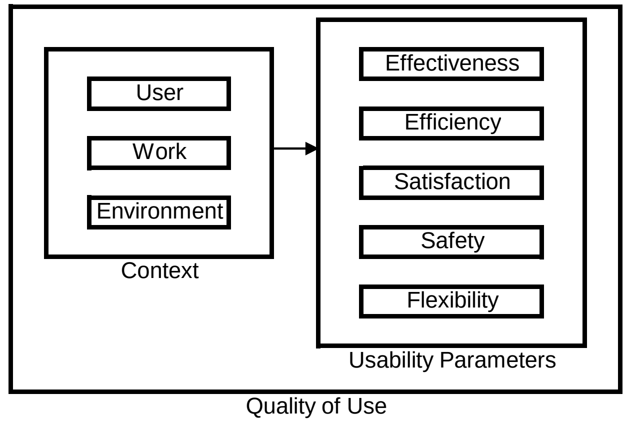

Figure 1: Quality of Use

Efficiency:

Efficiency refers to the measure of resources expended. How quickly a user can perform work is critical for business productivity. Measures of efficiency relate the level of effectiveness achieved to the expenditure of resources. The resources may be mental or physical effort, which can be used to give measures of human efficiency, or time, which can be used to give a measure of temporal efficiency, or financial cost, which can be used to give a measure of economic efficiency.

Satisfaction:

Satisfaction refers to comfort and acceptability of use. Comfort refers to overall physiological and emotional responses to use of the system. Acceptability of use measure overall attitude towards the system or user’s perception of specific aspects. Satisfaction is the extent to which expectations are met. Satisfaction is a success factor for any products with discretionary use; it’s essential for maintaining workforce motivation. Measures of satisfaction describe the perceived usability of the overall system by its users and the acceptability of the system to the people who use it and to other people affected by its use. Measures of satisfaction may relate to specific aspects of the system or may be measures of satisfaction with the overall system. Measures of satisfaction can provide a useful indication of the user’s perception of usability, even if it is not possible to obtain measures of effectiveness and efficiency. User satisfaction can be measured by the extent to which users have achieved their pragmatic and hedonic goals. ISO/IEC CD 25010.2 suggests the following types of measure:

• *Linkability:* the extent to which the user is satisfied with their perceived achievement of pragmatic goals, including acceptable perceived results of use and consequences of use.

• *Pleasure:* the extent to which the user is satisfied with their perceived achievement of hedonic goals of stimulation identification and evocation and associated emotional responses [6][7].
• *Comfort:* the extent to which the user is satisfied with physical comfort.
• *Trust:* the extent to which the user is satisfied that the product will behave as intended.

Satisfaction is most often measured using a questionnaire.

Safety:

Safety refers to acceptable levels of risk of harm to people, business, data, software, property or the environment in the intended contexts of use. Safety is concerned with the potential adverse consequences of not meeting the goals. There are no simple measures of safety. Historical measures can be obtained for the frequency of health and safety, environmental harm and security failures. A product can be tested in situations that might be expected to increase risks. Or risks can be estimated in advance.

Flexibility:

Flexibility is the ease with which a system or component can be modified for use in applications or environments other than those for which it was specifically designed [8]. It is also defined as the ability of a software application to deal with exceptions to the process model at the runtime and to cope with periodic changes to the process model. Classic and contemporary literature in software design recognizes the central role of flexibility in software design and implementation.

IV. FRAMEWORK FOR QUALITY USER INTERFACE DESIGN:

A fundamental reality of application development is that the user interface of any application is the system to the user. What user wants is for developers to build applications that meet their needs and that are easy to use. In software engineering community, the term usability has been more narrowly associated with user interface design. ISO/IEC 9126, developed separately as a software engineering standard, defined usability as one relatively independent contribution to software quality associated with the design and evaluation of the user interface and interaction.

The best way to ensure the improved usability is to use an orderly and well defined framework that is specifically geared to produce quality results. The ideal way to specify and measure usability would be to specify the features and attributes required to make a product usable and measure their effect on the implemented product. The problem with usability of user interface is that it is very difficult to specify what these features and attributes should be because the nature of the features and attributes required depends on the context in which the product is used. It is not meaningful to talk about the usability of the product, as usability is a function of the context in which the product is used. The characteristics of the context (users, task and environment) may be as important in determining usability as the characteristics of the product itself. Changing any relevant aspect of the context of use may change the usability of the product. For instance, the user interface may be improved by conforming the good dialogue design practices or the fit between the user and the rest of the overall system may be improved through means such as selection and training of users or good task design. A product which is usable by trained users may be unusable by untrained users. Aspects of working environment such as lighting, noise, or workstation design may also affect usability.

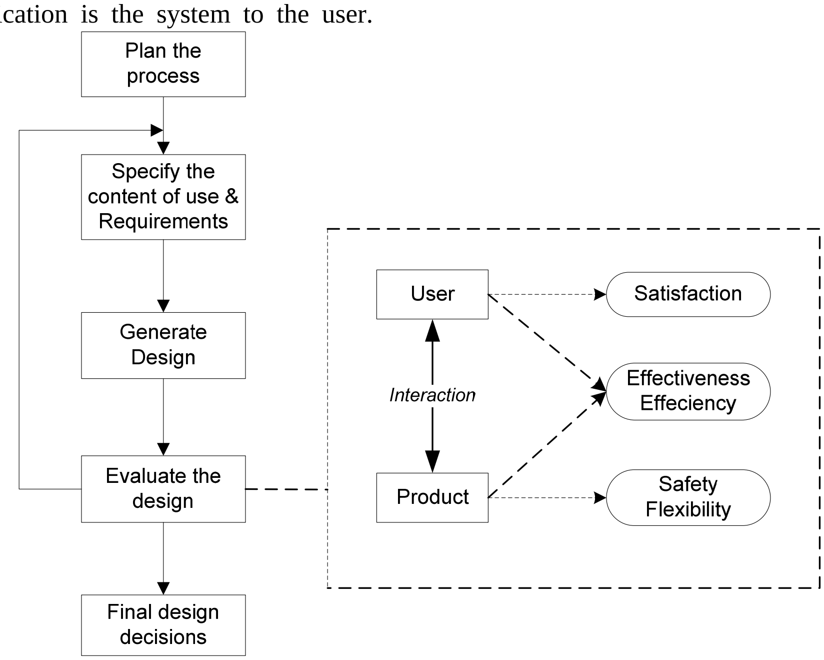

Figure 2: Framework for Quality User Interface Design

A new framework is introduced which defines a process for user interface design defining various attributes required for quality user interface which are related to both the user and the product. In the proposed framework, the first step is to plan the process which is human centered which will be helpful to the user while using the interface. After this, the context of use will be specified and the requirements for the application are determined. After defining the requirements, a design for the application is produced. This design solution for the interface has to be evaluated for quality. The overall objective is to achieve quality of use for both end users and the support users. The quality attributes like satisfaction, effectiveness and efficiency when combined together defines the functional feedback of the interface. The above defined framework represents the parameters like efficiency, effectiveness, safety, satisfaction and flexibility under technical, physical and organizational environment in a context and the dependence of user and product on these parameters resulting in quality of use. Efficiency refers to the resources such as time, money or mental effort that have to be expended to achieve intended goals. Effectiveness is the extent to which the intended goals of use of the overall system are achieved. Satisfaction is the extent to which the user finds the overall system acceptable. Satisfaction may further be subdivided into further attributes which represents satisfaction related to different goals, physical satisfaction and satisfaction with security. Safety is the acceptable levels of risk or harm to people, data, software, business in the intended context of use.

After the evaluation of design for quality is completed, if the quality achieved is the desired quality, the final quality design is produced. And if the desired quality is not achieved, the requirements are again checked and classified properly fro a quality design. The above framework defined provides a means of measuring the usability of the product by specifying various parameters of quality of use.

V. EVALUATION OF QUALITY PARAMETERS USING QUANTITATIVE MEASURES:

The evaluation of user interface quality can be done in a quantitative way by measuring the various quality parameters. Different types of user interfaces can be quantified and distinguished by the general concept of “interaction points”. Regarding to the interactive semantic of "interaction points" (IPs), different types of IPs must be discriminated (see also Denert, 1977). An interactive system can be distinguished in a dialog manager and an application manager. So we distinguish dialog objects (DO) and application objects (AO) and dialog function (DF) and application function (AF). A dialog context (DC) is defined by all available objects and functions in the actual system state. All dialog objects (functions, resp.) in the actual DC are perceptible (PD, PF) or hidden (HD,HF). Following are the three ratios to calculate Functional Feedback, Application Flexibility and Dialog Flexibility [10].

Functional Feedback: To activate the amount of “feedback” of an interface, a ratio is calculated: number of PFs (#PF=#PDFIP+#PAFIP) divided by number of HFs(#HF=#HDFIP+#HAFIP) per dialog context. (HDFIP is the hidden function interaction point of a dialog manager. PDFIP is the perceptible representation of a HDFIP.HAFIP is the hidden functional interaction point of the application manager and PAFIP is the perceptible representation of a HAFIP) This ratio quantifies the average “amount of functional feedback” of the functional space. The number of all different dialog contexts is abbreviated as D.

$$FB = 1/D \sum_{d=1}^{D} (\#PFd / \#HFd) * 100\%$$

Application Flexibility: To quantify the flexibility of application manager, we calculate the average number of HAFIPs per dialog context.

$$DFA = 1/D \sum_{d=1}^{D} (\#HAFIPd)$$

Dialog Flexibility: To quantify the flexibility of the dialog manager, we calculate the average number of HDFIPs per dialog context.

$$DFD = 1/D \sum_{d=1}^{D} (\#HDFIPd)$$

To quantify the above defined measures, the following values were found from an application having an I/O interface with both dialog manager and application manager.

D=20, P(AFIP)=360, P(DFIP)=520, H(AFIP,DFIP)=60

Base on the above values, we calculate the functional feedback, application flexibility and the dialog flexibility.

FB= 1/20(360/60+520/60)*100% = 73.3%

DFA= 1/20(360) = 18

DFA= 1/20(520) = 26

We interpret this result with the effect that flexibility must exceed a threshold to be effective (DFD,DFA>15) [11]. These calculated values of the attributes measure the interactive quality of the user interface.

VI. CONCLUSION:

The conventional assumption that quality is an attribute of a product is wrong, as the attributes required for quality will depend on how the product is used. The new framework introduced defines a user interface design process defining various attributes required for quality user interface related to both user and the product. The reviews with users of their requirements and the design of the final product in an iterative manner makes the user an active part of the process and results in a higher level of system usability and user satisfaction. Quality is generally treated as a property of a product, thus the product view of quality seeks to identify those attributes which are designed into a product or evaluated to ensure quality. The evaluation process to quantify usability attributes and the interactive quality of user interface enhances the quality of user interface.

VII. REFERENCES:

[1] Boehm, B., 1988. A spiral model of software development and enhancement. *IEEE Computer, 21,* 5, 61-72.

[2] Yourdon, E., 1989. *Modern structured analysis.* NY: Yourdon Press, Inc.

[3] De Marco, T., 1978. *Structured analysis and system specification.* NY: Yourdon Press, Inc.

[4] Garvin (1984) *What does "product quality" really mean?* Sloane Management Review, Fall 1984.

[5] Bevan N (1991a) Enforcement of HCI? Computer Bulletin, May 1991.

[6] Hassenzahl, M. (2003) The thing and I: understanding the relationship between user and product. In Funology: From Usability to Enjoyment, M. Blythe, C. Overbeeke, A.F. Monk and P.C. Wright (Eds), pp. 31 – 42 (Dordrecht: Kluwer).

[7] Norman, D. (2004) Emotional design: Why we love (or hate) everyday things (New York: Basic Books).

[8] IEEE. Standard Glossary of Software Engineering Terminology 610.12-1990, Vol.1. Los Alamitos: IEEE Press, 1999.

[9] Denert, E. (1977). Specification and design of dialogue systems with state diagrams. In: *International Computing Symposium 1977* (E. Morlet and D. Ribbens (Eds.)), 417-424. North-Holland, Amsterdam.

[10] M. Rauterberg (1995). Usability evaluation: an empirical Validation of different measures to Quantify interface attributes. *Swiss Federal Institute of Technology (ETH) Nelkenstrasse 11, CH–8092 Zuerich*

[11] Rauterberg, M. (1992). An empirical comparison of menu-selection (CUI) and desktop (GUI) computer programs carried out by beginners and experts. *Behaviour and Information Technology* **11**(4), 227-236.