5-2013

The Evaluation of Software Quality

Dhananjay Gade
University of Nebraska-Lincoln, dngade14@gmail.com

THE EVALUATION OF SOFTWARE QUALITY

By

Dhananjay N Gade

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the degree of Master of Science

Major: Industrial & Management Systems Engineering

Under the supervision of Professor Ram R Bishu

Lincoln, Nebraska

May 2013

THE EVALUATION OF SOFTWARE QUALITY

Dhananjay Namdeo Gade, M.S.

University of Nebraska, 2013

Adviser: Ram R. Bishu

Software Quality comprises all characteristics and significant features of a product or an activity which relate to the satisfaction of given requirements. The totality of characteristics of a software product depends upon its ability to satisfy given needs: for example (a) the degree to which software possesses a desired attribute or combination of attributes. (b) The degree to which a customer or user perceives that software meets their expectations, (c) the characteristics of software that determine the degree to which the software in use will meet the customer expectation. This study has three objectives. The first objective is to identify the Software quality dimensions that are relevant from the user perspective. The second objective is to identify if the software quality dimensions behave differently across different user levels. The third objective is to find out the software quality dimensions that are relevant for different softwares and generate a model for these softwares.

The data were collected for Novice and Expert users for MS WORD, MINITAB, MS OUTLOOK and GOOGLE SKETCH softwares. The analysis of variance (ANOVA) and regression analysis was performed on these data to find out the software quality dimensions that are relevant from the end users' perspective and determine the relationship between the software quality dimensions and the dependent variable; overall

software quality and Software rating. The ANOVA showed that consistency, maintainability, reliability, security, usability, and user interface (UI) aesthetics were significantly influenced by the user group.

The regression analysis showed that, For MS Word software, *Overall Software Quality (OSQ)* was significantly affected by *accessibility, security, interoperability, usability and stability*. The OSQ was significantly affected by *layout*, *security*, *interoperability*, *usability and stability*, in case of Minitab software. In case of MS Outlook software, the OSQ was significantly affected by *functionality, operability, user interface aesthetic, and maintainability*. And For Google Sketch software, the OSQ was significantly affected by *accessibility, maintainability, backward/forward convertibility (bfc), stability, operability, and precision*. From this regression analysis, it is recommended that the above mentioned significant dimensions are the important dimensions to improve the respective softwares' quality and software rating.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor Dr. Ram R. Bishu. His guidance not only remained in the academic research but also extended in many issues of my life. Throughout my graduate schooling, he provided motivation, best advice and teaching. I am profoundly indebted to Dr. Bishu for teaching me how to do research and for his selfless help throughout my MS study. Without his support, it would have been impossible to finish my studies. Furthermore, without his support, I could never have had taken up the opportunities outside the academics.

Besides my advisor, I would also like to thank my committee members, Dr. Lotfollah Najjar and Dr. Solomon To for their precious time, guidance and critical reading of this thesis.

I also thank my friends: Jake Riggle, Somnath Paramshetti, Aditya Bade,Shailesh Ganpule, Brian Neilson, RajKrishna Paul, Veera Selvan, Arvind Sundarmurthy, Vijay Krishna Musunuru, Ramdev Gowda, Shiva Bhagavatulla, Pravin Akula, Avinash Deshmukh, Benny Mathew Verghese, Vettrivel Gnaneswaran, Srinivas Venkatesan, Venkata Karthik Kolli. They all contributed a lot in making my stay in Lincoln enjoyable. They have been wonderful to me and I cannot forget their contribution in conducting this research.

I would like to thank my sister, Dipika for providing a lively and caring atmosphere for me. Lastly and most importantly, I wish to thank my parents, who have always been extremely encouraging and accommodative. I can never thank them enough for their love and sacrifice for better upbringing of me. I also thank my fiancée, Shruti and her family fort their encouragement and patience during the final stages of this work.

Table of Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Background:

These days softwares are being used increasingly in a wide variety of areas. Therefore their correct operations or working are very important for the success of businesses. One of the major concerns of the software industry is to produce high quality softwares. Therefore evaluation of software quality has always been of prime importance and highly prioritized task for software industries' professionals. Evaluation of software is a major factor in ensuring sufficient quality of the software product. This can be achieved by employing appropriate quality characteristics, taking into account of the purpose of the usage of the software product. It is very critical that every relevant software quality is evaluated using widely accepted and recognized metrics.

People/customers/end users use software to accomplish their tasks. In order to avoid failure of the software while customers are using it, the software needs to be tested thoroughly. In the past, various software quality models and metrics have been proposed and developed by different authors to measure the software quality. Software metric is a qualitative indicator of any software dimension whereas model specifies the relationship among these metrics.

The processes and methodologies that were put together to measure software quality lack uniformity. Sometimes Software Quality models are very specific that they measure particular set of quality dimensions with a certain metric. Therefore there is a need to provide the consistent system to measure software quality at a diverse and broad level.

1.2 Scope of Thesis:

This research aims to identify Software Quality (SQ) dimensions and to study their significance across different user levels. The method developed is validated through a user survey. This thesis has focused on developing a method to evaluate software quality across softwares from different domains. This research also details the development of a framework to measure the software quality and further development of a survey instrument for data collection.

Initially, the SQ dimensions will be finalized after reviewing the literature. The data would be collected using the survey instrument and will be analyzed using different linear modeling analyses (ANOVA and Regression) to yield the SQ dimensions that are relevant from the end users' perspective and the SQ dimensions that are relevant for MS Word, Minitab, MS Outlook and Google Sketch.

1.3 Overview of the Chapters

This thesis is presented in the following chapters. The chapter 2 consists of literature review on the quality, service quality, product quality and web quality. Chapter 3 explains about the rationale of this research and research objectives. Chapter 4 describes the research methodology, consisting of rationale of method and actual method. Chapter 5 presents the results of the analysis. Major findings of this research and recommendation for future study are presented in Chapter 6.

Chapter 2

LITERATURE REVIEW

2.1 Quality

The dictionary definition of quality is generally concentrated on excellence or goodness of use. However the technical definition of quality is "Fitness for the purpose" but these days because of customer focus, industrial and commercial thinking, may not fully endorse this description when software quality is described in these terms.

Quality is defined by international organizations as follows:

IEEE standard (IEEE Std 729-1983) defines the Quality as,

a. "The totality of features and characteristics of a software product that bear on its ability to satisfy given needs: for example, conform to specifications."
b. "The degree to which software processes a desired combination of attributes."
c. "The degree to which a user perceives that software meets his or her composite expectations."

Gilmore (1974) has defined the quality as "Conformance to specifications" while later Juran (1979) has defined the quality as "Fitness for use". Parasuraman et al. (1988), defines the quality as "Meeting and/or exceeding the customer expectations". Philip Crosby (1979-80) described the quality as "conformance to requirements" (requirements meaning both the product and the customer's requirements). According to American Society for Quality (ASQ) quality is a subjective term for which each person or sector has their own definition. In technical usage, 1. The characteristics of a product or service that bear on its ability to satisfy stated or implied needs; 2. A product or service free of deficiencies. Taguchi (1989) presented quality as “Quality is the loss a product causes to society after being shipped, other than losses caused by its intrinsic functions.”

2.2 Importance of Quality for a Business:

In almost every part of business, from the products and processes to the human resources and the whole management team, quality is always of prime importance. This is to make sure that the company’s products and services meet the required quality standards. These large companies give a lot of attention to quality because companies know that the quality of the product or service that will be provided ultimately impacts the brand. To retain customers for repeat business, a company must sell products that live up to the customer's expectations. If a customer has a good experience, then customers are likely to come back and spend money in business again the next time when the customers are in need of your products or services. A customer must feel like the product or service he bought from your company was worth the price. When customers feel overpaid for the quality or product received, business will likely not get any repeat customers in the future.

A company's reputation relies heavily on the quality of its products or services. If those expectations are met by the customers who buy their products, the company maintains its reputation. Customers, who receive a lower-quality product than expected, will complain to friends, family and co-workers about how the product or service did not live up to expectations, which will ultimately lower the organization’s/products’ consumer reputation.

In addition to this, if customers are unhappy with the quality of product, then the organization might have to face loss of market value, and even legal issues.

Therefore in order to satisfy end users, the product should be made of high quality. The following section explains in a brief about importance of end user satisfaction.

2.3 Service Quality:

Like product quality, the service quality is also important. It cannot be ignored or underestimated. The service quality is perceived by customers. And this might vary from customer to customer and from the quality of service actually delivered. Parasuraman et al. (1985) proposed a GAP model in which the author identified over 200 factors which determine service quality by conducting interviews with the end users and provided different techniques to gauge a firm's performance. "Service is a social act which makes which takes place in direct contact between the customer and representatives of the service company.", Says Parasuraman et al(2006). In their proposed GAP model, "Gap between expected service and experienced service" is highlighted. This concludes that experience of customer will define the service quality but not the actual deliverables' (product) quality.

Brady and Cronin (2001) mentioned that "work on service quality can best be described as divergent." As per the Woo and Ennew (2005) studies, SERVQUAL is the most dominantly used model for evaluating the service quality. In spite of being criticized by other scholars, the SERVQUAL is the most commonly used model for evaluating the service quality.

2.4 Measures of Service Quality:

There are measures of service quality that can be applied to services. Every service provider commits to deliver its service in a certain time frame. Therefore 'On time delivery' becomes the important measure of service quality. Another measure of the service quality is 'Effectiveness'. All services are supposed to accomplish something like provide information, repair an appliance, process a transaction, or develop a program, among others. If it is to determine if the service was effective, then this is an important measure. Services that are up and running must be concerned with availability. Examples include utilities providing water, electricity, gas, telephone, or other resources exactly when they're needed. Being down for a few hours can cause millions of dollars in losses and huge claims.

2.5 End User Satisfaction:

User satisfaction has been discussed by many researchers. Davis et al (1989) defines end user satisfaction as "the degree to which a person believes that using a particular system would enhance his or her job performance". Conrath & Mignen (1990) suggests that the impact of user expectations should be considered when assessing user satisfaction.

Customer satisfaction continued to be the subject of considerable research in the nineties and has been defined and measured in various ways (Oliver, 1997).End user satisfaction may be defined as the fulfillment of end user's expectations after using/consuming the product.

Feedback collection from the customer is essential for the supplier to ascertain customer satisfaction and scope for improvisation (Sugandhi, 2002). The advantages of high customer rating and satisfaction are enormous in the long run.

Also the business can be expanded by increasing the end user satisfaction, ultimately resulting in high number consumers. As end user satisfaction rises, so does customer repurchase intention (Anderson, 1994).

2.6 Product Quality:

The dictionary definition of product quality is "The collection of features and characteristics of a product that contribute to its ability to meet given requirements." During the year 1950, as a part of further development, new methods were evolved to control the Product's quality such as Statistical Quality Control and Statistical Process Control. From 1960, these methods were applied or introduced in the service industries for the first time. After the emergence of Japan and European countries as new leading markets, the focus began to shift on total product quality and production systems. Product quality can be viewed from three perspectives. First, the manufacturer is accountable for design, engineering, and manufacturing aspects of the product.

Any deviation from the standard design and engineering principles may lead to low product quality. The second perspective is end user's perspective. Users' views will be considered in this perspective. If the product quality meets users' requirements and expectations, then the users will rate it as high quality and reliable product. The third view is the product itself as a system. This will count towards the product's functionality and operational characteristics. In conclusion, the product quality is an outcome of manufacturer’s, end users’ and product’s functionality. Quality function deployment (QFD), and Total quality management (TQM) are some of the latest approaches toward the improvement of product quality. In addition to this, product quality can also be judged based on the number of warranty claims and associated costs.

The discussions in the above sections give an impression that there is no standard formula or mechanism or even a model which measures quality and fits for all types of service industries, especially in software industry. There is a need of more research in service industries. And among all of the service industries, software industry is growing and developing very fast. Therefore there is a need to maintain the software cost and quality proportion balanced and to improve the software quality.

2.7 Software Quality

Software quality is a very abstract term. It is relative to define its presence, but its absence is noticeable. Thus the thirst to improve the software quality goes up. Wikipedia defines the software quality as "In the context of software engineering, software quality measures how well software is designed (quality of design), and how well the software conforms to that design (quality of conformance). It is often described as the 'fitness for purpose' of a piece of software."

Software quality can be defined as; “How well the software complies with or conforms to a given standard or requirements, based on functional requirements or specifications. That attribute can also be described as the fitness for purpose of a piece of software or how it compares to competitors in the marketplace as a worthwhile product (Pressman, Scott- 2005).

According to IEEE Standard Glossary of Software Engineering Terminology [2], software quality is defined as, “The degree to which a system, a component, or process meets specified requirements.”

Testing of the software improves the customer satisfaction, reduced cost of development, reduced cost of maintenance, reduced time to market. In turn these all benefits will increase the profitability of the organization. Therefore evaluation of software quality is one of the key things in the success of organization.

2.8 Web Quality

According to Powell, web applications “involve a mixture between print publishing and software development, between marketing and computing, between internal communications and external relations, and between art and technology”. Web applications are a mixture of information, functionalities and services.

These days there is awareness and recognition among the large scientific and professional groups about the multi-faces of web applications. A web application encompasses computing, architecture, navigations, aesthetic, content presentations, security and heterogeneous environment. The distinction line between software and web application is getting very thin, as most of the software are trying to go on the web using cloud computing. However, like software, web applications have their own features, such as *content and document oriented, user centered, search and browse, worldwide accessibility, security, information, Intellectual property rights etc.* Like software, web applications also have executable source code, architecture, design pattern, structured data.

Most of the existing software quality models proposed are based on the ISO 9126 standards and in addition to that they are extended to suit the need of the software component. Therefore it becomes necessary to understand the ISO -9126 model. One part of this model is applicable to internal and external quality of software product while another model is intended for quality in use of a software product. The model is based on the six characteristics *functionality, reliability, usability, efficiency, maintainability, and probability.* As shown in Figure 2-1, each of these characteristics will have their own sub characteristics. The characteristics defined are applicable to every kind of software. The ISO-9126 feels to be more accurate and complete that the others and is free of drawbacks.

Characteristic	**Sub characteristics**
Functionality	Suitability
	Accuracy
	Interoperability
	Security
	Compliance
Reliability	Maturity
	fault Tolerance
	Recoverability
Usability	Understandability
	Learnability
	Operability
Efficiency	Time Behavior
	Resource Behavior
Maintainability	Analyzability
	Changeability
	Stability
	Testability
Portability	Adaptability
	Installabiltiy
	Co-Existence
	Replace ability

Figure 2-1. ISO/IEC 9126 characteristics 2004

McCall et al proposed a software quality factor framework in 1977 and distinguished the quality dimensions in three categories.

1. Product Operation factors:
    a. This consisted of *correctness, reliability, efficiency, integrity and usability.*
2. Product revision factors:
    a. This includes of *maintainability, flexibility, and testability.*
3. Product transition factors.
    a. This is made of *portability, reusability, and interoperability*.

Research sponsored by the United States Air Force led to propose a software measurement model which would consist of a comprehensive, hierarchical definition of software quality. Their framework consists of, at the highest level, quality factors that were defined which were appropriate for software acquisition mangers to use as help in specifying SQ objectives for their software. These high level factors were then divided into criteria which were software directed until the specific metrics were proposed that relate to the factors. By making these measurements, the author believed that a corresponding measurement will be obtained for the respective quality factor. To quantify the measures, the author identified major factors by considering the user, who was the program manager or acquisition manager. In order to identify and define the factors, the approach was to understand how the end user views the end product. The data was collected by asking questions to users which would represent each attribute, such as for *maintainability*; the question asked was *Can I fix it?* The measurements were taken during the development effort of the software. This way the McCall et al collected the empirical data. The establishment of relationship between the set of metrics related to a quality factor and a rating of the quality factor was achieved by regression analysis.

$r_f = c_1m_1 + c_2m_2 + c_3m_{3+}$ …..

Where:

$r_f =$ rating of a quality factor.

c_i = regression coefficients.

m_i = various measurements identified as relating to the quality factor, f.

This relationship was then used as a predictor. This model, quantifies the definition of software quality, aids data collection.

Boehm et al(1976) proposed a model which was more or less similar to McCall model. Boehm described his model in a hierarchical way and divided each characteristic into sub characteristics, as shown in figure 2-2.

ISO 9126	Functionality	Suitability
		Accuracy
		Interoperability
		Security
		Standards
	Reliability	Maturity
		fault Tolerance
		Recoverability
	Usability	Understandability
		Learnability
		Operability
	Efficiency	Time Behavior
		Resource Behavior
	Maintainability	Analyzability
		Changeability
		Stability
		Testability
	Portability	Adaptability
		Installabiltiy
		Conformance
		Replace ability

Figure 2-2.Boehm quality model

Boehm et al (1976) first identified and classified a set of characteristics which are important for software. Then they considered a FORTRAN based software and developed candidate metrics for assessing the degree to which the software has the identified and defined characteristics. Boehm et al then went on to investigate the correlation between characteristics and associated metrics with the software quality and also quantifiability, which was done by developing an algorithm. In order to determine if there are overlaps, dependencies, shortcomings etc., the author evaluated each candidate metric with respect to the above mentioned criteria and with respect to its interactions with other metrics. The author concludes by saying that careful attention to characteristics of software quality can lead to significant savings in software life cycle costs. Author also claims that this research provides for the first time that a clear, well defined framework for assessing the software quality using the consistent and mutually supportive sets of definitions, distinctions, guidelines.

Deutsch et al (1998) divides the software quality into two categories, software procedure quality and software product quality. Software procedure quality consists of *technology, tools, personnel, organization and equipment* whereas software product quality consists of *document clarity, design traceability, integrity, program reliability, organization, test integrity.*

Word et al (1999) considers a broad and general approach towards software quality. The author proposed the following four major categories.

1. User based: To be evaluated by end users.
2. Product delivery based: To be evaluated by the designer.
3. Manufacturer based: Evaluate the development process, process quality control.
4. Organization based: Consists of project costs, resources, production time.

Dromey's Model(1995) tried to build quality carrying properties into software. This model tried to connect tangible and less tangible characteristics. This model talks more about where to find the defects. The author initially defines *a set of structural forms, a set of quality carrying properties and a set of high level quality attributes*. In the process of building a constructive model of software product quality, the author started by defining the relationship among these three sets of entities. The author considered C programs and used the PASS (Program Analysis and Style System) tool to get a comprehensive report on the quality of programs. Dromey added a new characteristic *reusability* in his proposed model. This model was also based on McCall model. In the end author concludes that this model provides an explicit process for building quality-carrying properties into software.

Sharma et al. (2008) derived a quality model which was based in ISO/IEC 9126. Their model was from the perspective of Component Based Software Development. The author included *track ability, complexity, reusability, and flexibility* as new sub dimensions in their model. For the analysis part, the authors used Analytical Hierarchy Process (AHP) and for weight values, conducted a survey.

Chang et al (2008) proposed the directions to evaluate software quality by the use of fuzzy theory and AHP. These authors also based their model upon ISO/IEC 9126 quality model. Instead of taking a conventional way of weighing the values either by survey or interviews, the author used a new approach of fuzzy theory to get relative weights of characteristics and sub characteristics.

Another component based quality model was proposed by Alvaro et al (2005). This model also follows the ISO/IEC 9126 model.

Alvaro et al investigated a Software Component Certification framework with the aim of acquiring quality in software components. Their framework, *the component quality model*, was composed of four modules. A) A component quality model. B) A certification techniques framework. C) A metrics framework. D) A certification process. This model was classified into two classes: the quality characteristics that are observable at runtime and the quality characteristics that can be observable during the product life cycle.

However the author added new sub characteristics to the existing ISO/IEC 9126 model and removed some sub characteristics from the same to develop the consistent model. The newly added sub characteristics are *self-contained, configurability, scalability, and reusability*. The *maintainability* and *analyzability* sub characteristic was removed for ISO model. The author discussed about the metrics used to measure the SQ attributes. The metric consisted of A) Presence B) Values C) Ratio. First metric tells whether the attribute is present or not, the second metric indicates the exact values of the component information and the third metric describes percentages. The author attempted to investigate the component certification area in order to: A) define component quality model; B) determine the certification techniques to evaluate the necessary characteristics C) A framework for defining a set of metrics.

Rawesdah et al (2006) proposed a model which was again based on ISO/IEC 9126 model. Their objective was to build one model suitable to work for a variety of COTS based systems. The authors identified a small set of agreed upon; high level quality attributes and decomposed each attribute in a top down approach into a subset of subordinate attributes to harness the COTS evaluation requirements. The authors have removed *fault tolerance, configurability, scalability, and reusability* from ISO-9126 model and added a new sub characteristic called *manageability* in order to maintain the consistency with COTS.

As this model is based on Commercial off the shelf (COTS) components, it was essential to make the distinction between internal and external metrics. Therefore the author considered external metrics as appropriate for COTS components. Based on the quality attribute, the author used different stakeholders (Users). The author considered project manager for manageability and quality assurance people for functionality, as stake holders. The author has just presented the new model for COTS based components. The limitation of this model is, that it does not measure internal attributes of software quality.

Unlike the ISO/IEC 9126, the FURPS model (1992) consists of five characteristics i.e. *functionality, usability, reliability, performance and supportability.* The name of the model itself stands for these characteristics. This model consists of two steps; one is about priority and second is about defining the characteristics which are measurable.

In the above literature review it is evident that a thread of commonality has existed in the basic model that is being used and the dimensions/characteristics and sub characteristics that are being used to measure the SQ. Some of the dimensions are used in all the models which convey their importance in measuring software quality while few are relevant based on what is approach towards the model development. This research looks at all the available models, investigates the commonality between all these models and uses it to prepare a set of dimensions/characteristics and a unique model to evaluate the software quality from an end users' perspective.

Chapter 3

RATIONALE OF THE RESEARCH

3.1 Critique of the Literature:

There have been several models discussed above. Following is the summary of their study and the drawbacks of their study.

1. One of the important contributions of McCall model for software quality is defining the relationship between SQ dimensions. However, this model did not consider one of the major attributes of software quality, that is *functionality*. By definition, *functionality* is what a product, such as a software application or computing device, can do for a user. Therefore functionality is important factor to evaluate the software so that it would meet users' needs.
2. Boehm's (1976) model proposed a SQ model based on the users' needs but did not give any suggestions about measuring the software quality characteristics mentioned in that model. There can be more research done on the measurement of SQ dimensions. Therefore this research shows a need to evaluate the software quality.
3. FURPS (1992) model decomposed the SQ dimensions into two categories, a) Functional, b) Non-functional. One major drawback of this model is that, it does not consider one of the important quality attribute *portability*. Portability is the ability of the software to work in different environments. And users' computing environment might keep changing and accordingly software also needs to adapt to new operating/computing environment *portability* can be a major attribute and

therefore cannot be neglected. This study considers *portability* as one of the major attribute.

4. ISO-9126 (2004) looks to be more accurate, complete and does not fall short as other models do. However, it has not provided the clarity of how certain software quality aspects/attributes can be measured. This can, however, be the best model in comparison to the other proposed models.
5. Bertoa's (2002) model does not discuss other significant characteristics/sub-characteristics like interfaces, versions and reusability. However, later models developed are based on this model.
6. Alvaro's quality model was a huge advancement in the development of software quality models. This model was much similar to Bertoa's model. Alvaro's model has added *self-contained*, business, *configurability* and *scalability* attributes which were not present in Bertoa's model. This model introduces a new attribute, *business*. This attributes consists of sub-characteristics like d*evelopment time, cost, marketing time, and target customers.* However these characteristics may not be measure of the quality of the software.
7. Rawesdah' quality model added a new main characteristic called *Manageability*, which describes about quality management. As per the author's description, this attribute seems to be more process oriented and not the product oriented. It can be a good software quality model if the product quality attributes are considered rather than the process quality attributes, because the end user may not be concerned about the process the software is developed with.

8. Dromey’s model tried to increase the understanding of the relationship between characteristics and sub-characteristics. Therefore this model could not focus on how to measure the software quality. It has established the relationship between characteristics and sub-characteristics. This research therefore studies this relationship and evaluates the software quality.

3.2 Need of Work:

Once the software quality dimensions are proposed, the measurement of these dimensions in order to determine the software quality is not easy as it is in other industries, like process or manufacturing. As discussed in the previous section, some of the studies could not explain how to measure the proposed SQ dimensions. Some of the other studies could not consider some of the SQ dimensions which could be important. And very few studies have considered the ultimate end users of the software. Very few studies considered end users but these studies considered software developer, program manager, and software testers as end users for their studies. There is a need to understand the perspective of the ultimate end user. In order to consider the wide range of end users, this study considers users of different user level such as expert and novice. Therefore this study tries to propose a model with which SQ can be measured from the end users’ perspective.

3.3 Objectives:

Software characteristics might vary based on how the softwares be used. Therefore quality attributes also need to vary from software to software. The objective are,

1. Identify SQ dimensions that are relevant from the user perspective.
2. Do the SQ dimensions behave differently across different user level?
3. What are the SQ dimensions relevant to different software types and generate a model for different softwares?

Chapter 4

RESEARCH METHODOLOGY

4.1 Rationale of Method:

One of the objectives of this research is to identify software quality dimensions from the end users' perspective. The tool used to collect data is a web based survey method. The survey method is the most popular method used when a perspective of large numbers of people needs to be understood. The questionnaire instrument was chosen to collect the responses (gather information from participants). This method helps end users or participants to immediately share their perspective/experience via the questionnaire.

Privacy of participants is a major factor in order to encourage people to participate and gather perspectives as accurate as possible. Questionnaires help to maintain the privacy of participants as the responses can be kept anonymous. This is particularly relevant while gathering personal information. Also, once the data collection is done, this data can be used as a reference for future studies.

4.2 Actual Method:

DEVELOPMENT OF THE FRAMEWORK

A literature review of SQ models shows an obvious commonality among the above mentioned SQ models. Most of the models have considered ISO-9126 as the base model while evaluating the software quality. This research investigates the extent of overlap among different SQ models in evaluating SQ considering the end users' perspective.

The following steps were involved in the development of the model and they are discussed in detail in the later sections of this thesis.

STEP 1: To find common characteristics.

STEP 2: Finalize and concisely define each characteristic from the end users' perspective.

STEP 3: Developing survey instrument.

STEP 1: TO FIND COMMON CHARACTERISTICS

In 1991, International Standards Organization (ISO) published a software quality model, ISO-9126, to evaluate the software quality. This model published twenty eight dimensions of SQ.

Subsequent researchers in the field of SQ have based their work on either the ISO-9126 model or they introduced a new model in evaluating SQ. This study, after extensive literature research, identified the following summary of the most common dimensions and the number of occurrences that have appeared in different SQ models. This study mostly referred to literatures published from 1991- 2010.

DIMENSIONS	ISO 9126 (1991)	Behshid Behkamal et al (2008)	A. Rawasdeh et al(2006)	McCall et al(1977)	A. Alvaro et al(Nov 2010)	J. Yahaya et al(2010)	Avadhesh Kumar et al (Sept 2009)	Ioannis Samoladas et al(2008)	Bartoa et al(2003)	Grady and Caswell FURP's Model	Total
Maintainability	X	X	X	X	X	X	X	X	X	X	10
Testability	X	X	X	X	X	X	X	X	X	X	10
Functionality	X	X	X		X	X	X		X	X	8
Interoperability	X	X	X	X	X	X	X		X		8
Maturity	X	X	X		X	X	X	X	X		8
Usability	X	X	X	X	X	X	X		X		8
Efficiency	X	X	X	X	X	X	X		X		8
Suitability	X	X	X		X	X	X		X		7
Accuracy	X	X			X	X	X		X	X	7
Reliability	X	X	X		X	X	X		X		7
Recoverability	X	X	X		X	X	X		X		7
Understandability	X	X	X		X	X	X		X		7
Learnability	X	X	X		X	X	X		X		7
Operability	X	X	X		X	X	X		X		7
Time-based Efficiency	X	X	X		X	X	X		X		7
Resource Based Efficiency	X	X	X		X	X	X		X		7
Changeability	X	X	X		X	X	X	X			7
Portability	X	X		X	X	X	X			X	7
Security	X	X				X	X	X		X	6
Analyzability	X	X				X	X	X	X		6
Adaptability	X	X			X	X	X			X	6
Compliance		X	X		X		X		X		5

Continued...

DIMENSIONS	ISO 9126 (1991)	Behshid Behkamal et al (2008)	A. Rawasdeh et al(2006)	McCall et al(1977)	A. Alvaro et al(Nov 2010)	J. Yahaya et al(2010)	Avadhesh Kumar et al (Sept 2009)	Ioannis Samoladas et al(2008)	Bartoa et al(2003)	Grady and Caswell FURP's Model	Total
Fault Tolerance	X	X			X	X	X				5
Stability	X	X			X		X	X			5
Replace ability	X	X			X	X	X				5
Compatibility		X	X						X	X	4
Complexity		X	X				X		X		4
Installabiltiy	X	X				X	X				4
Conformance	X					X	X				3
Reusability				X	X		X				3
Attractiveness	X	X									2
Configurability					X					X	2
Scalability					X		X				2
Standards	X										1
Self-Contained					X						1
Feature set										X	1
Capabilities										X	1
Generality										X	1
Effectiveness								X			1
Frequency/severity of failure										X	1
Recoverability										X	1
Predictability										X	1
Mean Time to Failure										X	1

Continued...

DIMENSIONS	ISO 9126 (1991)	Behshid Behkamal et al (2008)	A. Rawasdeh et al(2006)	McCall et al(1977)	A. Alvaro et al(Nov 2010)	J. Yahaya et al(2010)	Avadhesh Kumar et al (Sept 2009)	Ioannis Samoladas et al(2008)	Bartoa et al(2003)	Grady and Caswell FURP's Model	Total
Human Factor										X	1
Aesthetics										X	1
Consistency										X	1
Documentation										X	1
Code Reducibility							X				1
Modularity							X				1
Coexistence		X									1
Deploy ability					X						1
Manageability			X								1
Quality Management			X								1
Integrity						X					1
Data Protection						X					1
User Factor						X					1
User's Perception						X					1
User Requirement						X					1
Business									X		1
Time to Market									X		1
Targeted Market									X		1
Development Time									X		1
Development Cost									X		1
Performance									X		1
Speed									X		1
Resource Consumption									X		1
Throughput									X		1
Response time									X		1
Supportability									X		1
Extensibility									X		1

Continued...

DIMENSIONS	ISO 9126 (1991)	Behshid Behkamal et al (2008)	A. Rawasdeh et al(2006)	McCall et al(1977)	A. Alvaro et al(Nov 2010)	J. Yahaya et al(2010)	Avadhesh Kumar et al (Sept 2009)	Ioannis Samoladas et al(2008)	Bartoa et al(2003)	Grady and Caswell FURP's Model	Total
Serviceability									X		1
Installabiltiy									X		1
Localizability									X		1
Integrity				X							1
Flexibility				X							1
Mailing list quality								X			1
Documentation quality								X			1
Developer base quality								X			1

Table 4-1: Tracing SQ Dimensional Commonality in Existing frameworks

Table 4-1 provides a summary of the most common dimensions and the frequency with which they have appeared in the mentioned SQ frameworks. Maintainability and Testability appeared in all the frameworks. Functionality, Interoperability, Maturity, Usability, Efficiency showed their presence in 8 out of 10 frameworks. Accuracy, Reliability, Portability, Understandability, showed their presence in 7 out of 10 models. Security, Analyzability and Adaptability showed up in 6 models out of 10 models. Recoverability and documentation with their presence each once in the list were at the bottom in the frequency table.

STEP 2: FINALIZE AND CONCISELY DEFINE EACH CHARACTERISTIC FROM THE END USERS' PERSPECTIVE.

After identifying the most frequently occurring 21 SQ dimensions, it was necessary to consider them from the end users' perspective. Considering this, three new sub characteristics Layout, Usefulness, and Interface Aesthetic were proposed as a new addition to the list. This took the number of characteristics to 24.

A focus group of five graduate and undergraduate students was used to understand their perspective towards these attributes. These students were treated as end users. The first part of the experiments required the students to play with the four softwares; 'MS WORD', 'MINITAB', 'GOOGLE SKETCH', 'MS OUTLOOK' and to complete the survey questionnaire. After completing the individual surveys, the scores for each attribute were analyzed. After that, the definition of attributes was discussed with the focus group and gathered their input to find out the relevance of these attributes from end user's perspective.

The attributes were defined along the line with definitions published in literature and except 'accuracy' and 'accessibility'; each attribute had one question designed. The 'accuracy' and 'accessibility' had two questions each designed. Since 'accuracy' and 'accessibility' attributes were difficult to measure in one question, having two questions would give better understanding of the users' perspective. For other attributes, only one question was considered to get the understanding of users' perspective. For this research, the end user was considered to be someone who uses the software for either their business needs or to accomplish their daily tasks. The end users consist of students, professors, and professional users of these softwares.

Input from the focus group suggested that ‘Usefulness’ is as same as ‘usability’. According to the focus group, usability has a broader sense than that of ‘usefulness’. Therefore ‘Usability’ was kept and ‘Usefulness’ was removed. The ‘Layout’ and ‘Interface Aesthetic’ were strongly recommended as necessary attributes from the end users’ perspective. The final list contained 23 SQ attributes plus two additional qualitative attributes, ‘Software rating’ and ‘Overall Quality’. This takes the list of dimensions to 25. The definition of these dimensions is given in the table 4-2. The dimensions are defined in line with the definitions understood in the available literature research. Table 4-2 explains the meaning of each dimension.

Dimension	Definition
Accessibility	The ability of software to let users access regardless of their disability or severity of impairment
Accuracy	The degree of correctness with which the results are produced.
Adaptability	The ability of the software to adapt to changes in its working environment.
Analyzability	How well software can be analyzed, when required.
Availability	The extent to which a system can continue to work when a significant component or set of components goes down.
Backward/forward Convertibility	Ability to be use the document/code in the both newer and older versions.
Consistency	Producing the same behavior every time.
Documentation	Availability of resources to learn about the software.
Functionality	An aspect of what a software application can do for a user.
Hardware Dependency	The dependency of software on hardware for efficient running.
Interoperability	the ability of diverse software systems to work together
Layout	The position of graphical components.
Maintainability	A set of attributes that bear on the effort needed to make specified modifications.
Operability	Ability to be operable.
Portability	A set of attributes that bear on the ability of software to be transferred from one environment to another.
Precision	The ability to reproduce the same computational result as many times as requested.
Reliability	A set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time.
Security	Ability for data encryption.
Stability	Ability to be stable under any conditions like high load, complex operations.
Time and Resource based efficiency	Ability to respond (Response time) and memory, disk utilization.
Understandability	The ease of learning/understanding the software.
Usability	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.
User Interface Aesthetic	The external look of the software components like color, size of components.
Software Rating	Recommendation of software to others.
Overall Quality	Total quality of software.

Table 4-2 Definition of SQ Dimensions

The table 4-2 gives the definitions of software quality dimensions considered for this study. *Accuracy* of software can be defined as the ability of software to produce the accurate results whereas *precision* is the ability of the software to reproduce the same results under the same conditions/inputs. Availability of software is the extent to which software can be ready to use even when significant components fail. Interoperability of software is the ability of different softwares to work together. Portability can be defined as the ability of software to work with different environment such as different operating systems. Reliability of software is the capability of software to maintain the expected level of performance under stated conditions and inputs over the periods. Security of software is very essential for the privacy of the users and the data. Software can be said to be secure enough when it can protect users' data.

STEP 3: DEVELOPING SURVEY INSTRUMENT:

The first draft of the survey was developed in such a way that each participant user was asked to spend some time and to perform some of the basic tasks, or whatever tasks they wanted to do, and then to answer the total 27 questions regarding the 25 dimensions. The questions were designed by keeping in mind the definition of attributes. The pilot test was conducted to arrive at the final survey instrument.

Pilot study:

For the pilot study, a group of seven people consisting of undergraduate and graduate male and female students was formed. There were 3 undergraduate and 4 graduate participants. Subject recruitment was by invitation, and based upon experience with the software. Students used (level of using) these types of softwares anywhere between two and eight hours per day. Some of them were using it 2 hours a day and some of them were using 7-8 hours a day.

The pilot study included a disclaimer, a welcome page, an instruction page, survey questions and a thank you page. Some of the major changes made based on the feedback received from the pilot study were:

- Five point Likert scale was changed to six point Likert scale because there were few questions to which students did not have answer. To avoid recording incorrect responses, it was more feasible to have a choice of selecting 'Don't know'
- Questions were rephrased to reduce the ambiguity.
- Task scenarios were added to the survey, and each participant performed one task only.

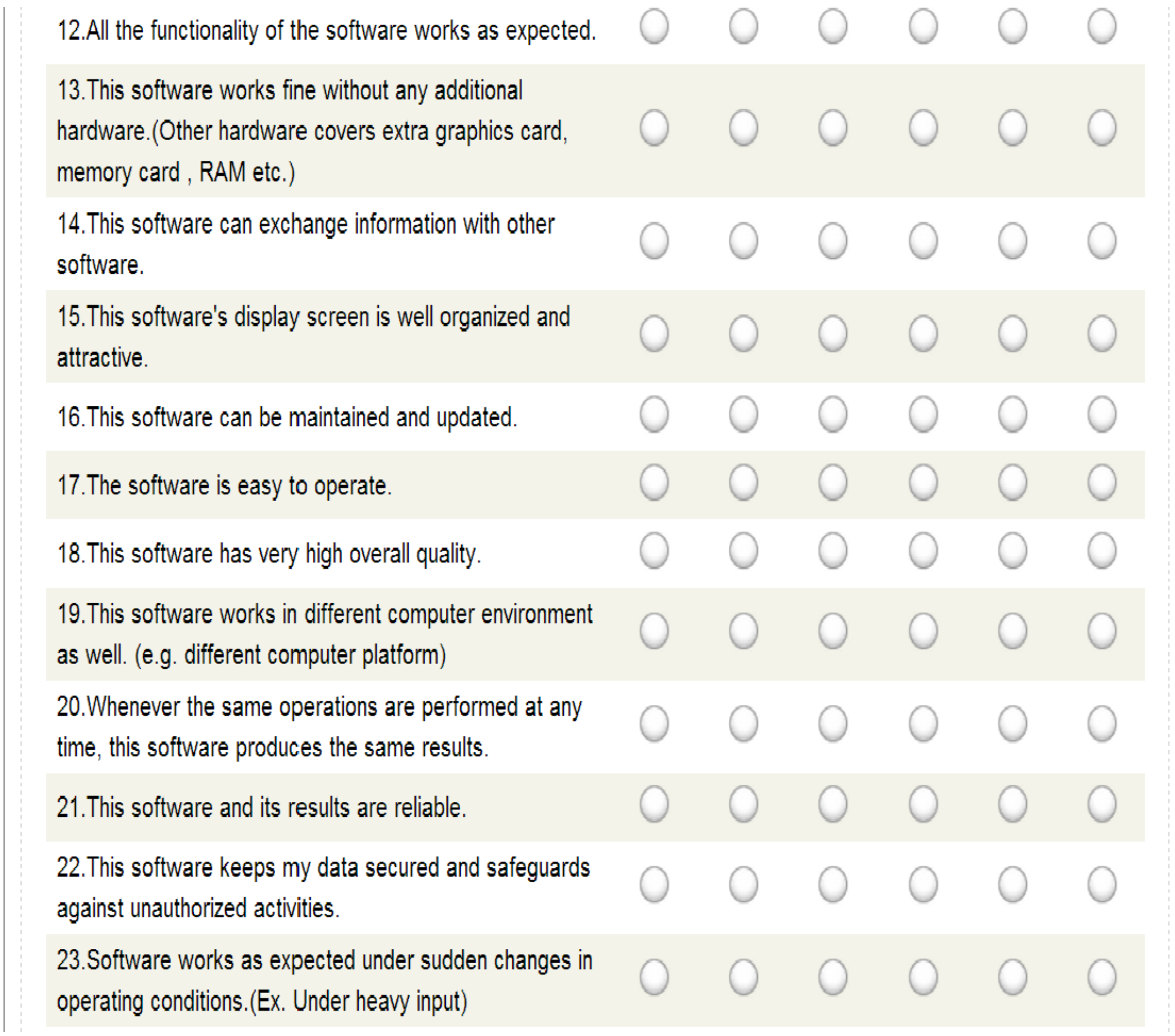
Task scenarios were added in such a way that users needed to complete the tasks first before moving on to the questionnaire. The main reason was, to let the user have the most recent experience before filling the questionnaire. The estimated time for completing the survey was 15-20 minutes. The 27 questions are shown in Appendix A.

The final survey included 5 demographic questions, 27 questions focused on measuring the 25 SQ attributes.

Analysis:

DESCRIPTION OF SURVEY QUESTIONS:

The survey questionnaire consisted of 27 questions for each software which measured 25 Software Quality dimensions. The questions were the same for each software.



12.All the functionality of the software works as expected.

13.This software works fine without any additional hardware.(Other hardware covers extra graphics card, memory card , RAM etc.)

14.This software can exchange information with other software.

15.This software's display screen is well organized and attractive.

16.This software can be maintained and updated.

17.The software is easy to operate.

18.This software has very high overall quality.

19.This software works in different computer environment as well. (e.g. different computer platform)

20.Whenever the same operations are performed at any time, this software produces the same results.

21.This software and its results are reliable.

22.This software keeps my data secured and safeguards against unauthorized activities.

23.Software works as expected under sudden changes in operating conditions.(Ex. Under heavy input)

Figure 4-1Snapshot of a page from actual survey.

The above figure 4-1 is the snapshot of one of the survey pages. The detailed survey link is attached in the appendix A. Each question in the survey questionnaire represents one SQ dimension. The question 12 tries to find out the users' opinion on the *functionality* dimension of the software. Question 14 is about the *interoperability* dimension of the software. Users' perspective on the *layout* of the software is known by the question 15 'This software's screen is well organized and attractive.' The *reliability* of the software is perceived by the question 21 'This software and its results are reliable.' Question 22 'This software keeps my data secured and safeguards against unauthorized activities.' records users' perspective about the *security* of software. Que.17 'This software is easy to operate.' represents the *Operability* of the software.

For the analysis purpose, the respondents had to reply to each question. The responses available for each question were 'Strongly Agree', 'Agree', 'Neutral', 'Disagree', 'Strongly Disagree' and 'Don't Know'. For one software and one user level, there was only one respondent.

For data analysis, the between subjects design was used. Because, the data collection was done in such a way that each subject is tested under only one condition i.e. each participant was allowed to complete the survey for one software only. This way the data was collected for four types of software and two levels of users, Novice and Experts. In this study, different levels (i.e. Novice and Experts) for the same factor are considered. In this case, the factor is software type i.e. MS WORD, MS OUTLOOK, MINITAB, GOOGLE SKETCH and levels are user levels i.e. Novice and Expert. As one of the objectives of this study is to know whether SQ dimensions vary across different user levels. Therefore, between subjects design is used.

Chapter 5

RESULTS

The previous chapter explains in detail the rationale of method, and the development of software quality framework. It also explains number of questions in the questionnaire, data collection method, and type of design for analysis. Chapter 5 discusses the demographics and the results of the analysis.

DEMOGRAPHICS:

A total of 160 participants responded to the survey questionnaire with 20 subjects in each cell. This is shown in Table 5-1.

	MS Word	**MS Outlook**	**Minitab**	**Google Sketch**	**Total**
Novice	20	20	20	20	80
Expert	20	20	20	20	80
Total	40	40	40	40	160

Table 5-1: Data Distribution- Balanced Cell

Demographic division is shown in Table 4-4. The number of female respondents was 57(35%) compared to 103 (65%) male participants. There were 87 graduate students with a response rate of 54% while undergraduates , faculty/staff and other professions had around 20%,0.01% and 23% response rates respectively. Approximately, 76% of the respondents' population were between 19-20 years of age, and 19% of the respondents population were between 31-45 years of age. This study was planned to have equal numbers of users who use these software types less than 2 hours daily and for 5 or more than 5 hours daily, so the percentage of Novice/casual users and Expert users is 50%. The variations between Novice and Expert users are distinguished based on the number of hours, the user uses the software. If the user uses software for more than five hours per day then that user was termed as an Expert/Power user while if the user uses software for less than two hours per day then that user was termed as a Novice user. These two types of user levels were chosen because this would help us to reach to a broad distribution of subjects. Expert users will have better insight of the software while Novice users will have less insight of the software than Expert users. In order to understand the broad perspective of users, this study had considered Novice and Experts level.

	Female	**Male**		
Gender	57	103		
	<2hours/day	**>5hours/day**		
Usage	80	80		
	Undergraduate	**Graduate**	**Faculty/Staff**	**Other Profession**
Academic Status	33	87	2	38
	19-30 years	**31-45 years**	**46-60 years**	**>60 years**
Age	123	31	6	0

Table 5-2 Demographic statistics

5.1 ANOVA Summary

The data was tested for its normality and homogeneity. The data was found to be normally distributed and homogeneous. Table 5-3 summarizes main effects and interaction effects of four independent variable: ‘software type’, ‘user group’, ‘software nested under user group’ and ‘subject nested under interaction of software and user group'. Each 25 dimensions is a dependent variable. The ANOVA was performed using SAS software.

	Dimension	Mean	Significance		
			SW	Group	Subjects(SW*Group)
Y1	Accessibility	2.05	NS	NS	NS
Y2	Accuracy	2.45	NS	NS	NS
Y3	Adaptability	2.43	NS	NS	NS
Y4	Analyzability	2.58	NS	NS	NS
Y5	Availability	2.46	NS	NS	NS
Y6	Bwd./fwd. Convertibility	2.66	0.025	NS	0.0268
Y7	Consistency	2.66	0.021	0.067	0.0398
Y8	Documentation	2.55	NS	NS	NS
Y9	Functionality	2.37	NS	NS	NS
Y10	Hardware Dependency	2.1	NS	NS	NS
Y11	Interoperability	2.57	NS	NS	NS
Y12	Layout	2.74	.0006	NS	0.0006
Y13	Maintainability	2.44	NS	0.0074	0.0021
Y14	Operability	2.57	NS	NS	NS
Y15	Portability	3.03	NS	NS	0.0003
Y16	Precision	2.68	NS	NS	NS
Y17	Reliability	2.72	NS	NS	NS
Y18	Security	2.25	NS	0.0435	0.0152
Y19	Stability	2.74	0.035	NS	0.0056
Y20	Time and Resource based efficiency	2.72	NS	NS	NS
Y21	Understandability	2.04	NS	NS	NS
Y22	Usability	2.43	NS	0.01	NS
Y23	User Interface Aesthetic	2.43	NS	0.0161	NS
Y24	Software Rating	2.77	NS	NS	NS
Y25	Overall Quality	2.31	NS	NS	NS

Table 5-3 ANOVA summary for Software, group and subjects.

Notations: SW: Software; Group: User group, S: Significant, NS: Non-Significant

Summary results in table 5-3 show clearly that seventeen out of twenty five dimensions are not significantly impacted by any of the independent factors. These dimensions are *Accessibility, Accuracy, Adaptability, Analyzability, Availability, Documentation, Functionality, Hardware Dependency, Interoperability, Operability, Portability, Precision, Reliability, Time and Resource based efficiency, Understandability, Software Rating and Overall Quality.* The value of $\alpha = 0.05$

Software has a significant effect on dependent measures/dimensions *Bwd/Fwd. Convertibility, Consistency, Layout, and Stability.*

Software type has a significant effect on dependent measures backward/forward compatibility, consistency, layout and stability. The mean values of these dimensions across four software types are plotted in below figure 5-1, 5-2, 5-3 and5-4 respectively.

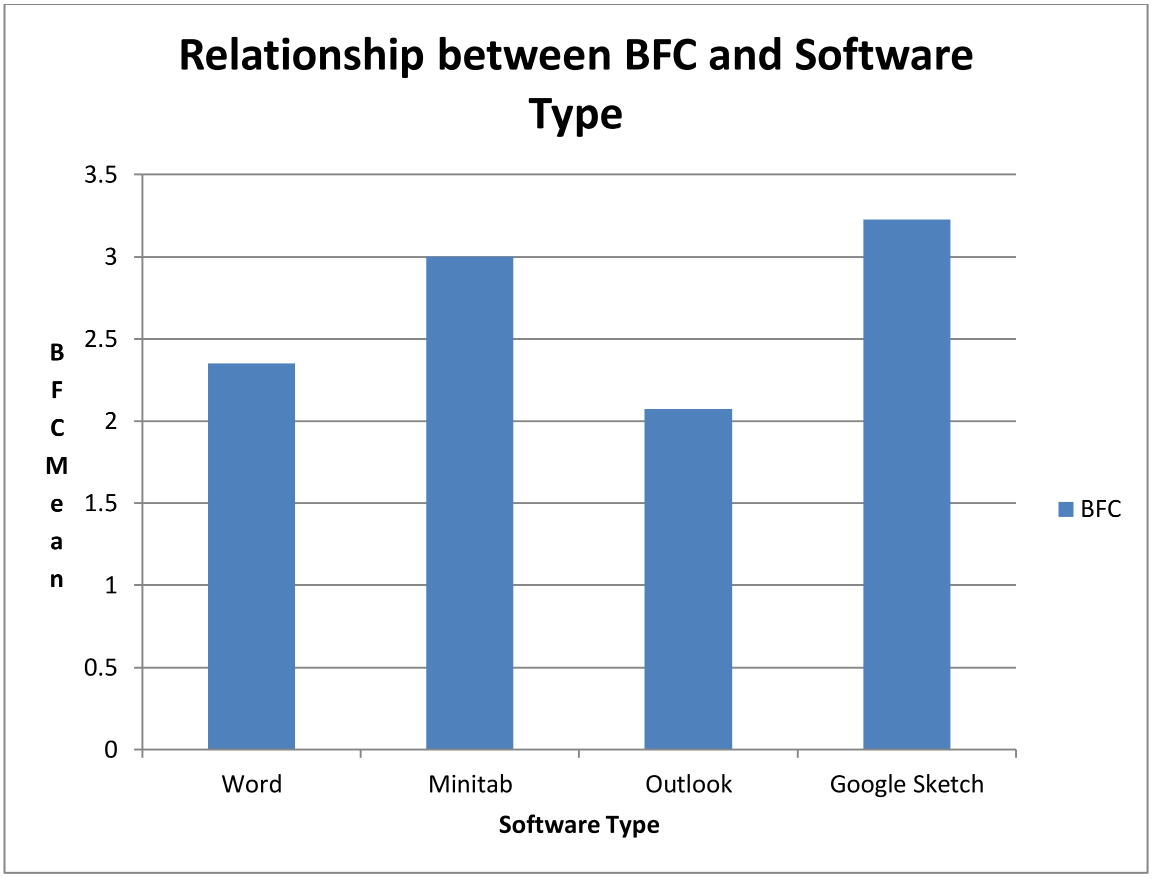


Figure 5-1: Relationship between Backward/forward compatibility and Software Type

Backward/forward compatibility dimension (Shown in figure 5-1) was statistically significant with the software type. The Tukey test was performed to compare the means of four softwares. Based on the Tukey test results, it can be said that there was a statistically significant difference between the means of MS Word and Google Sketch. MS Outlook is also significantly different from Minitab and Google Sketch.

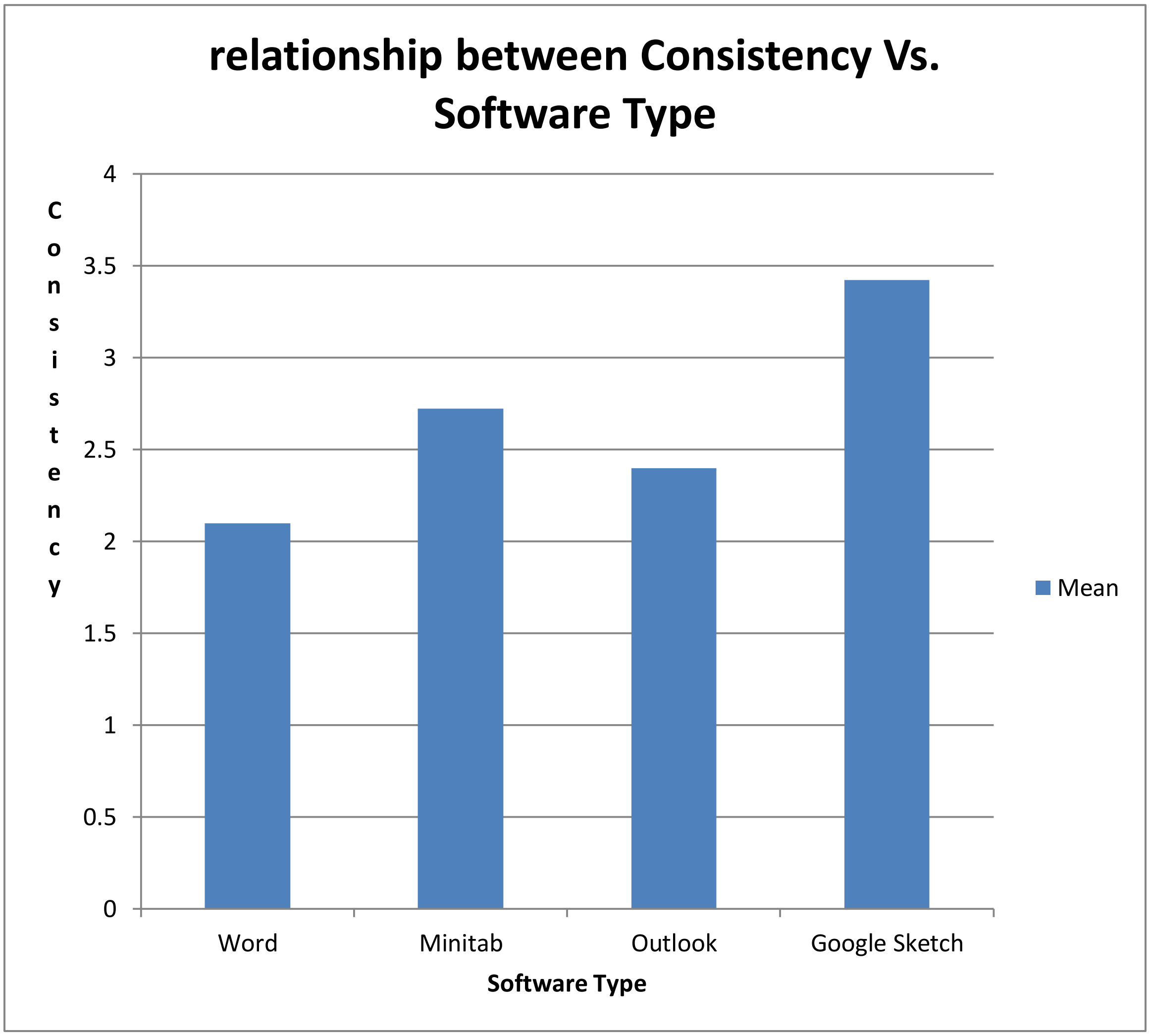


Figure 5-2: Relationship between Consistency and Software Type

Consistency dimension (Shown in figure 5-2) was statistically significant with the software type. From the results of the Tukey test, it was found that there is a significant difference between the means of MS Word and Google Sketch, Minitab and Google sketch softwares.

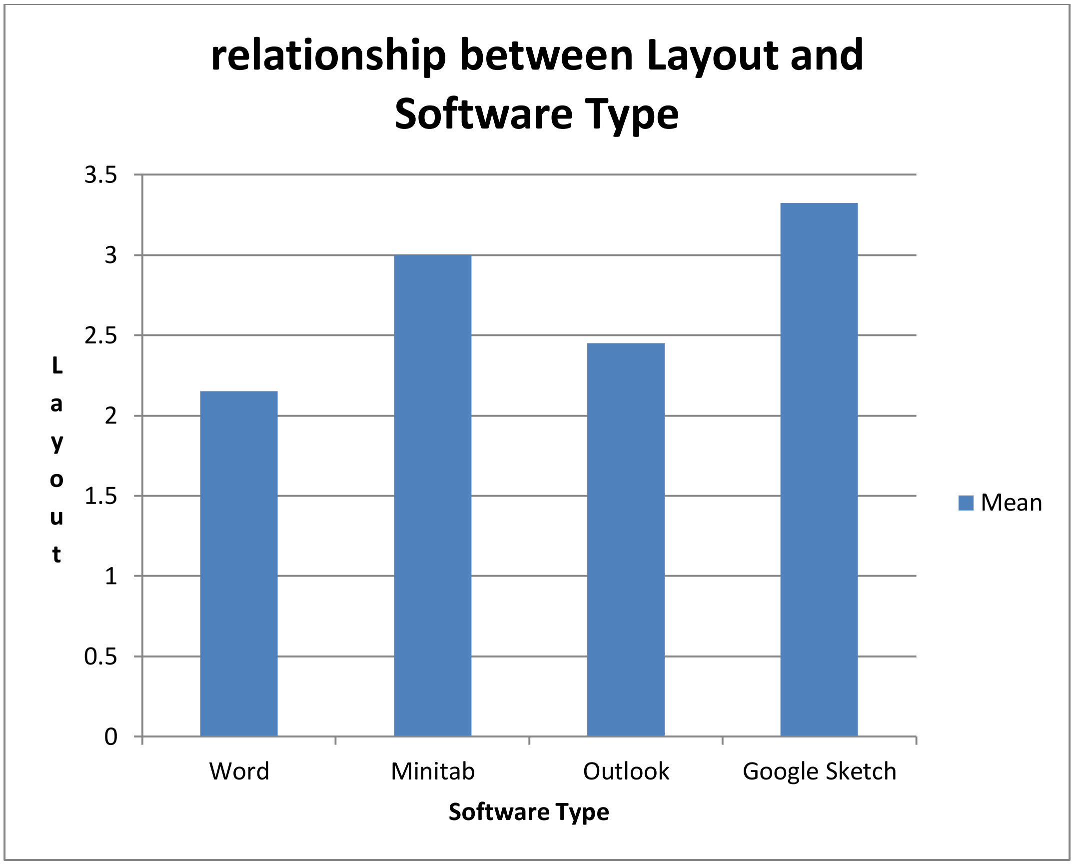


Figure 5-3: Relationship between Layout and Software Type

Figure 5-3 has mean values plotted for the *Layout* attribute. The Tukey test result shows that there is a significant difference between means of these four softwares. MS Word is significantly different from Google Sketch and Minitab while MS Outlook is significantly different from Google Sketch.

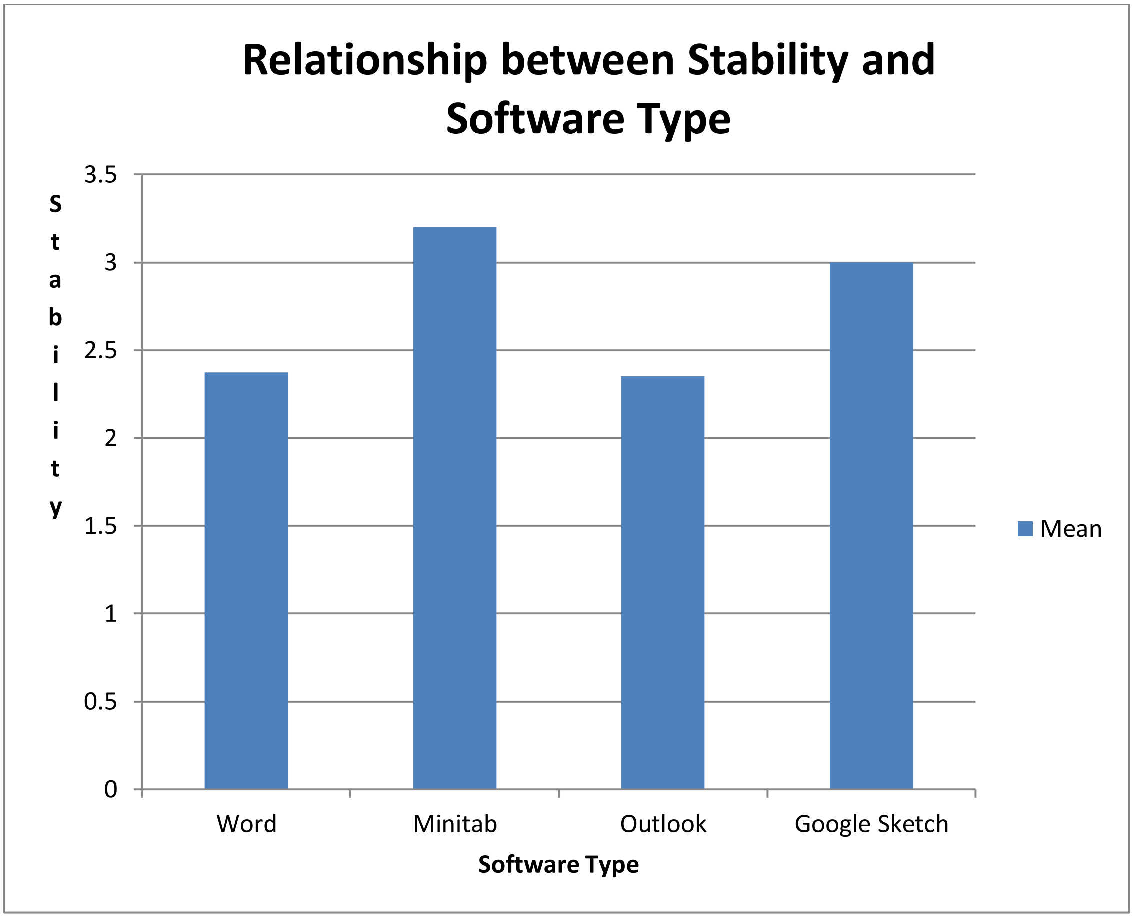


Figure 5-4: Relationship between Stability mean and Software Type

Figure 5-4 is a plot of mean values of *Stability* against the software type. Stability was an attribute that was significant with the software type. From the Tukey test, it was found that there is a significant difference between the means of Minitab and MS Outlook. The means of MS Word and MS Outlook are not statistically significant. There is no significant difference between the means of Minitab and Google Sketch.

Main effect Group has significant impact on four independent SQ dimensions of Consistency, Maintainability, Security, usability, UI Aesthetic. The mean values of these dimensions are plotted against two levels of users, namely Novice and Expert. The graphs are shown in figure 5-6, 5-7, 5- 8 respectively.

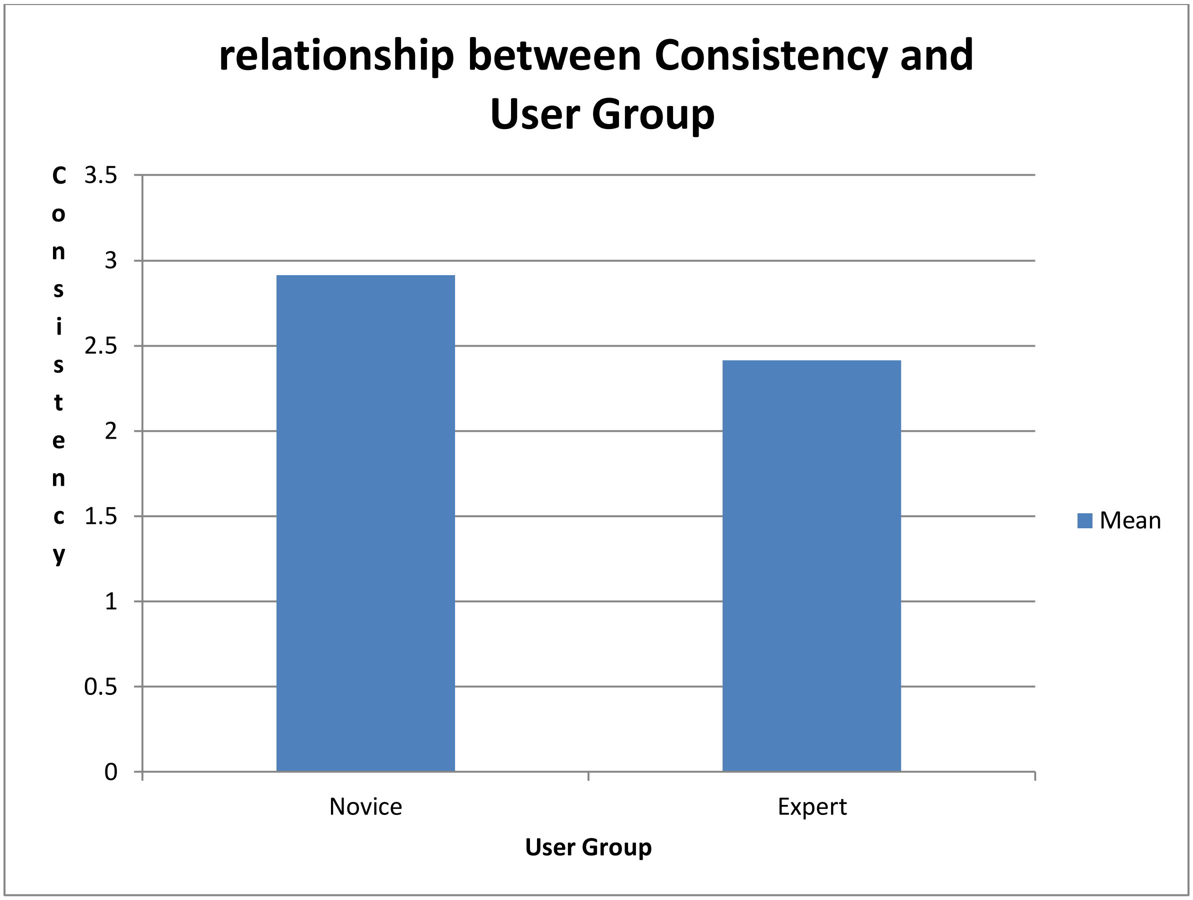


Figure 5-5: relationship between Consistency and User Group

Figure 5-5 is a plot of mean value of *Consistency* across user level, i.e. Novice and Expert. When comparing Expert users and Novice users' agreement on consistency of the software, Novice users feel that softwares were less consistent whereas Expert users agree that software were consistent.

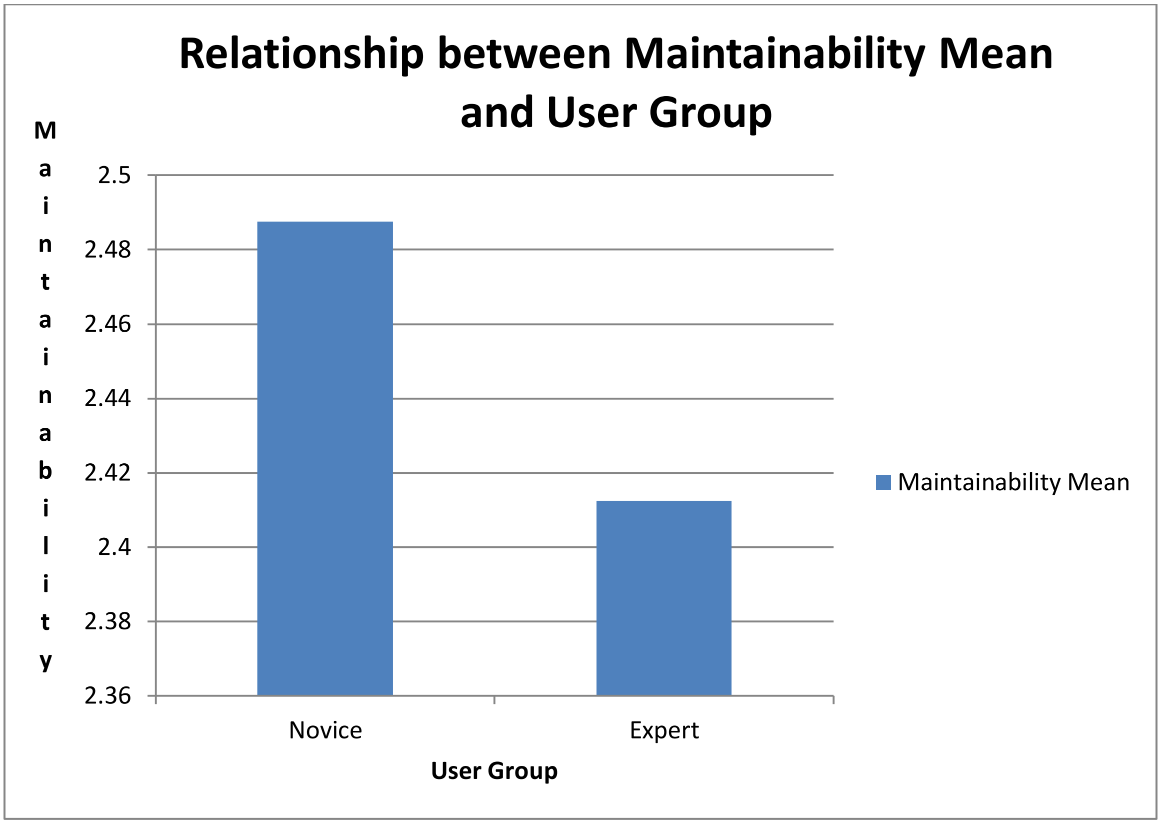


Figure 5-6: Relationship between Maintainability and User Group

Figure 5-6 is a plot of mean values of Maintainability across different user groups. Expert users agree that softwares are maintainable and can be updated while Novice users feel that softwares have less maintainability.

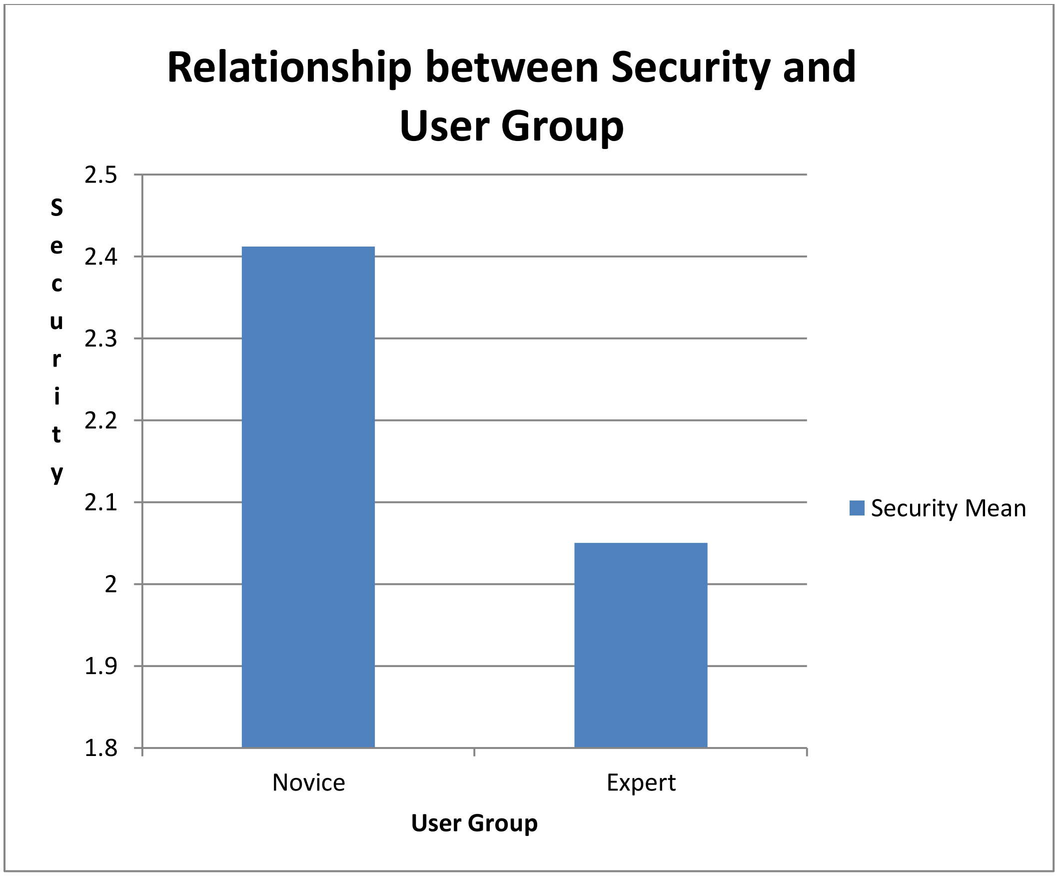


Figure 5-7: Relationship between Security and User Group

Figure 5-7 is a plot of mean values of Security across different user groups. Expert users agree that softwares are secure enough while Novice users feel that softwares are less secure.

Backward/forward compatibility, consistency, Layout and stability varies from software to software which means that though they are considered important measures of SQ, according to end users, their presence is different among these softwares.

The dimensions which are not significant are shown in table 5-3. Though these dimensions are not significant from the end users' perspective, the presence of these dimensions in the software is important to build the high quality software.

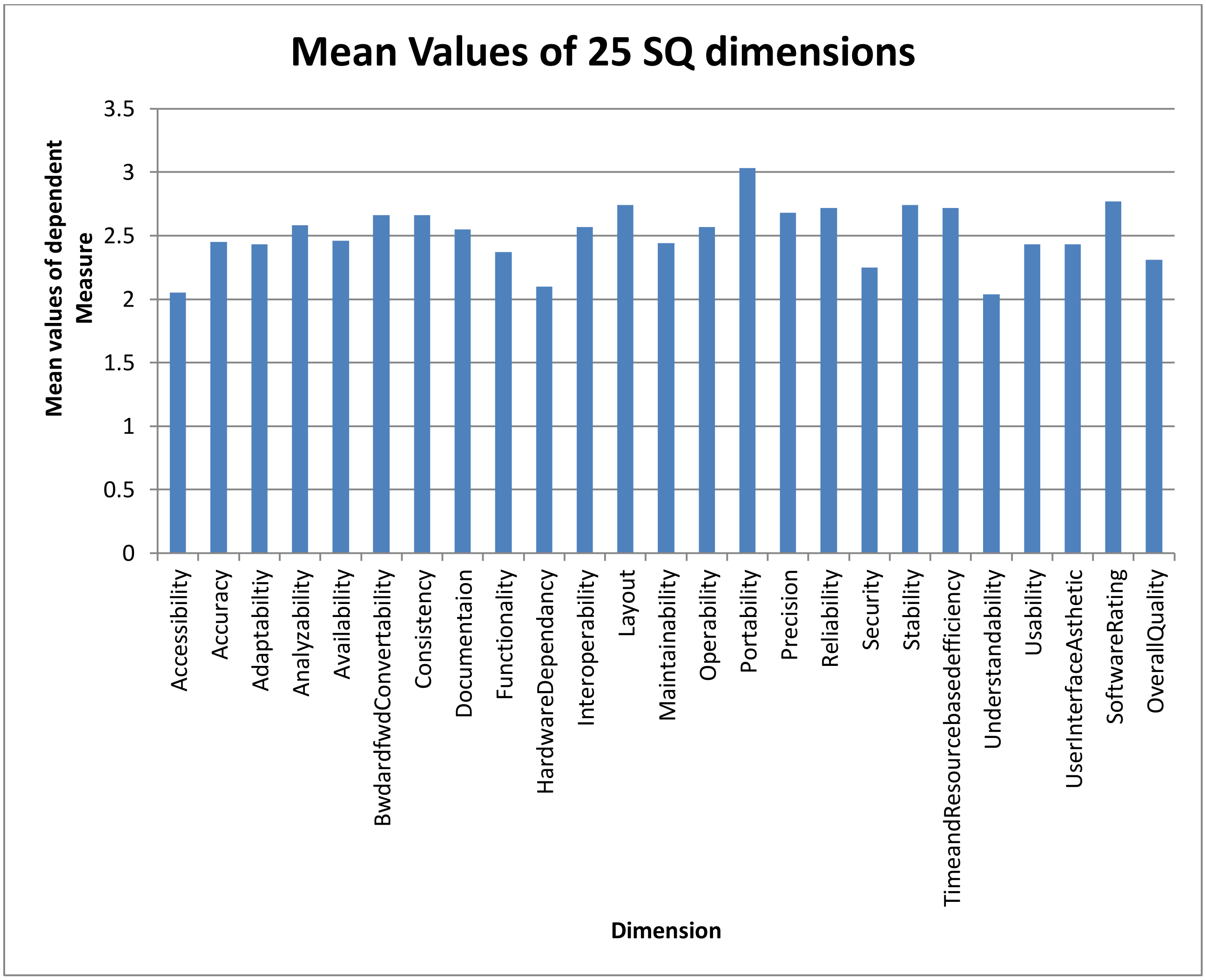


Figure 5-8: Plot of mean values of all SQ dimensions.

Mean values of the SQ dimensions in figure 5-8 indicate that all the 25 dimensions' mean value is above 2.0. The mean value is above 2.0 which convey that on average the users ‘Agree’ to the questions in the survey. And thirteen dimensions have mean value above 2.5. This survey questionnaire had response value ranging from 1-6. The mean value is 3. Mean values above 3 indicates that response/perception of users towards disagreement and below indicates that response/perception of users towards agreement. This will highlight the dimensions, which are present and which are not present, according to users. This will help in evaluating the four softwares and build a model for these softwares.

5.2 Regression Analysis:

In the total of 27 questions, two questions which represents “overall software rating (SR)” and “Overall software quality (OSQ)” respectively were considered as the dependent variables. The regression analysis was performed using qualitative question attributes as the predictors of the responses to the two SQ outcomes (dependent variables). The stepwise and normal regression analysis was done for each type of software separately.

The regression analysis was performed to study the relationship between Overall software quality (dependent), Software rating (dependent) and rest of the dimensions (independent) variables. The stepwise regression was achieved by successively removing the variables. This gave the fine-tuned model which would consider only significant SQ dimensions and their coefficients to generate the model. The models for the output of regression analysis with regression coefficients can be written as mentioned below.

Regression Analysis	Software Type	Regression Model	Significant	R2 value
Stepwise	MS WORD	1.035 + (0.72)ACC -(0.86)5 + (0.54)IN TR + (0.26)U - (0.29)STBL	Yes	0.59
Enter	MS WORD	0.51 +(0.702)ACC + (0.335)ACCR + (0.358)ADA- (0.032)ANLY + (0.197)AVL -(0. 043)BFC + (0.126)CON - (0.298)DOC - (.207)FUN + (0.328)HD + (0.827)INTR + (0.365)1- (0.225)M - (0.404)0 - (0.482)P + (0.105)PRC-(0.366)R -(1.02) S-(0.271)STBL + (0.383)TRBE +(0.082)UNDR + (0.174)U + (0.072)U IA	No	0.72
Stepwise	MINITAB	(-0.02804) + (0.371) U + (0.143)L + (0.27)C- (0.246) R -(0.19)UNDR + (0.20)A	Yes	0.72
Enter	MINITAB	0.035 +(0.568)ACC + (0.261)ACCR - (0.029)ADA - (0.296)ANLY +(0.0727)AVL + (0.203)BFC + (0.008)CON -(0.097) DOC- (0.085)FUN -(0.003)HD + (0.257) INTR -(0.09)L-(1.428)M + (0.0932)O + (0.117)P+ (0.063)PRC-(0.178)R +(0.205)S - (0.092)STBL - (0.084)TRBE+(0.264)UNDR + (0.294)U-(0.097)UIA	No	0.87
Stepwise	MS OUTLOOK	(-.9522)+(0.47)0+ (0.28)U1A+(0.33)FUN+(0.29)M	Yes	0.56
Enter	MS OUTLOOK	(-0.88) +(0.801)ACC + (0.435)ACCR- (0.310)ADA - (0.344 AN LY+ (0.283)AVL -(0.328)BFC + (0.097)CON - (0.346)DOC-(0.084)FUN - (0.628)HD + (0.381)IN TR - (0.207)1 +(0.315)M +(0. 585)0 -(0.242)P + (0.209)PRC - (0.166) R + (0.356)S -(0.022)5181- (0.134)TRBE + (0.547)UNDR + (0.015)U+(0.614)UIA	No	0.77
Stepwise	Google Sketch	(-.2761) + 0.79 ACC + 0.46 M -0.31 8FC + (0.31)STBL -(0.221)O +0.16 PRO	Yes	0.74
Enter	Google Sketch	(-0.56) + 0.786ACC + 0.027 ACCR - 0.251 ADA- 0.049 ANLY - 0.183 AVL - 0.463 BFC - 0.010 CON - 0.139 DOC + 0.105 FUN +0.235 HD - 0.044 INTR - 0.030 L+ 0.474 M - 0.2610 + 0.241P +0.368 PRO + 0.126 R - 0.268 S + 0.281 STBL + 0.232 TRH -0.088 U+ 0.008 U IA	No	0.82

Table 5-4 Regression analysis for Overall Software Quality (OSQ)

Regression Analysis	Software Type	Regression Model	Significant	R^2 Value
Stepwise	MS WORD	2.6 – 0.152 (TRBE)-0.205(U) + 0.175(INTR) -0 .18(HD)	YES	0.34
Enter	MS WORD	2.76 –(0.154)ACC + (0.189)ACCR + (0.174)ADA + (0.196)ANLY - (0.204)AVL –(0.298)BFC– (0.124)CON + (0.257)DOC – (.126)FUN + (0.014)HD + (0.145)INTR – (0.087)L + (0.262)M - (0.299)O – (0.062)P + (0.0444)PRC –(0.152)R + (0.243)S – (0.126)STBL – (0.019)TRBE – (0.110)UNDR –(0.346)U + (0.194)UIA	NO	0.66
Stepwise	MINITAB	2.171 – (0.256) O+ (0.294)P – (0.207) STBL + (0.163) CON	YES	0.4
Enter	MINITAB	2.34 +(0.016)ACC - (0.037)ACCR + (0.230)ADA - (0.172)ANA + (0.249)AVL –(0.210)BFC + (0.232)CON - (0.058)DOC – (0.025)FUN - (0.192)HD + (0.272)INTR– (0.116)L - (0.242)M - (0.325)O + (0.445)P- (0.012)PRC –(0.193)R+ (0.009)S –(0.222) STBL – (0.199) TRBE – (0.149)UNDR+(0.032)U + (0.218)UIA	NO	0.8
Stepwise	MS OUTLOOK	(-.70444) +(0 .389)STBL+(0.39)AVL +(0.260)TRBE + (0.241)PRC – (0.22)DOC+(.141)R	YES	0.73
Enter	MS OUTLOOK	(-0.88) +(0.801)ACC + (0.435)ACCR - (0.310)ADA - (0.344)ANLY + (0.283)AVL-(0.328)BFC + (0.097)CON - (0.346)DOC – (0.084)FUN - (0.628)HD + (0.381)INTR – (0.207)L +(0.315)M + (0.585)O -(0.242)P + (0.209)PRC–(0.166)R + (0.356) S – (0.022)STBL – (0.134)TRBE + (0.547)UNDR+(0.015)U+(0.614) UIA	NO	0.85
Stepwise	GOOGLE SKETCH	(-2.05) + (0.30)S + (0.46)TRBE + (.93)HD+(0.42)U + (0.357)PRC– (0.32)AVL	YES	0.7
Enter	GOOGLE SKETCH	(- 1.98) + 0.259ACC + 0.430ACCR - 0.437ADA - 0.155 ANLY - 0.324 AVL - 0.610 BFC+ 0.229 CON + 0.031 DOC + 0.232FUN + 0.982 HD - 0.074 INTR - 0.094 L - 0.245 M - 0.156 O+ 0.116 P + 0.583 PRC + 0.282 R + 0.145 S + 0.165 STBL + 0.526 TRBE + 0.573 U- 0.336 UIA	NO	0.83

Table 5-5 Regression analysis for Software Rating (SR)

Table 5-4 and Table 5-5 shows the stepwise method and enter method regression models generated for each of the softwares MS Word, Minitab, MS Outlook and Google Sketch. Table 5-4 and Table 5-5 shows the regression models for OSQ and SR for fours softwares. In these Tables, the model is considered significant by looking at the significance. The enter method regression model includes the dimensions which are significant and not significant whereas stepwise model consists of the dimensions which are significant.

The best fit model is decided based on the statistically significant variables as determined in stepwise regression. Therefore, for both the independent variables, the best fit model is stepwise model which is explained in detail as below.

1) Regression analysis of MS WORD software:

1.1) The results show that SR is affected by *Time and resource based efficiency, Usability, Interoperability and Hardware dependency.*

Therefore the model with regression coefficient can be written as

SR = 2.6 – 0.152 (TRBE)-0.205(U) + 0.175(INTR) -0 .18(HD)

It can be said that more the *Interoperability* of the MS WORD software, more the software rating and in turn high recommendation of the software to others. Whereas, lower the *Time and resource based efficiency, Usability, and Hardware dependency,* lower the software rating and in turn lower recommendation of software to others.

1.2) The OSQ was significantly affected by *Accessibility, Security, Interoperability, Usability and Stability.* Therefore the model with regression coefficient can be written as

OSQ = 1.035 + (0.72) ACC – (0.86) S + (0.54) INTR+ (0.26) U – (0.29) STBL

It can be said that more the *Interoperability, Accessibility, Usability* of the MS WORD software, more the software quality. Whereas, lower the *Security and stability*, lower the MS WORD software quality.

2) Regression analysis of MINITAB software:

2.1) In case of Minitab software, the results show that SR is affected by *Operability, Stability, Portability and Consistency*. Therefore the model with regression coefficient can be written as

SR = 2.171 – (0.256) O+ (0.294) P – (0.207) STBL + (0.163) CON

It can be said that more the *Portability and Consistency* of the Minitab software, more the software rating and in turn high recommendation of this software to others. Whereas, lower the *Operability, Stability*, lower the software rating and in turn lower recommendation of MINTAB software to others.

2.2) The OSQ was significantly affected by *Layout, Security, Interoperability, Usability and Stability*. Therefore the model with regression coefficient can be written as

OSQ = (-0.02804) + (0.371) U + (0.143) L + (0.27) C– (0.246) R + (0.19) UNDR + (0.20) ACC

It can be said that more the *Layout, Consistency, Accessibility, Usability* of the Minitab software, more the software quality. Whereas, lower the *Reliability*, lower the software quality**.**

3) Regression analysis of OUTLOOK Software:

3.1) For MS Outlook software, the results show that SR is affected by *Availability, Stability, Precision, TRBE, Documentation, and Reliability*. Therefore the model with regression coefficient can be written as

SR = (-.70444) + (0 .389) STBL+ (0.39) AVL + (0.260) TRBE + (0.241) PRC – (0.22) DOC+ (.141) R

It can be said that more the *Availability, Stability, Precision, TRBE and Reliability* of the Outlook software, more the software rating and in turn high recommendation of the software to others. Whereas, poor the Documentation, lower the software rating and in turn lower recommendation of software to others.

3.2) The OSQ was significantly affected by *Functionality, Operability, UIA, and Maintainability*. Therefore the model with regression coefficient can be written as

OSQ = (-.9522) + (0.47) O + (0.28) UIA + (0.33) FUN + (0.29) M

It can be said that more *Functionality, Operability, UIA, and Maintainability* of the Outlook software, more the software quality.

4) Regression analysis GOOGLE SKETCH Software:

4.1) For GOOGLE SKETCH software, the results show that SR is affected by *Security, Hardware Dependency, Usability, Precision, TRBE, and Availability*. Therefore the model with regression coefficient can be written as

SR = (-2.05) + (0.30) S + (0.46) TRBE + (.93) HD + (0.42) U + (0.357) PRC– (0.32) AVL

It can be said that more the *Security, Hardware Dependency, Usability, Precision, TRBE,* of the GOOGLE SKETCH software, more the software rating and in turn high recommendation of the software to others. Whereas, lesser the *Availability*, lower the software rating and in turn lower recommendation of software to others.

4.2) The OSQ was significantly affected by *Accessibility, Maintainability, BFC, Stability, Operability, and Precision*. Therefore the model with regression coefficient can be written as

$$OSQ = (-.2761) + 0.79\ ACC + 0.46\ M - 0.31\ BFC + 0.19 STBL - 0.221\ O + 0.16\ PRC$$

It can be said that higher *Accessibility, Maintainability, Stability and Precision* of the GOOGLE SKETCH software, higher the software quality of GOOGLE SKETCH. While poor the *Operability, and BFC,* poor the software quality of GOOGLE SKETCH.

Abbreviations:

SoftwareRating(SR),OverallQuality(OSQ),Accessibility(ACC.),Accuracy(ACCR.),Adaptability(ADA),Analyzability(ANLY),Availability(AVL),BFCompatibility(BFC),Consistency(CON),Documentation(DOC),Functionality(FUN),Hardwaredepenendency(HD),Interoperability(INTR),Layout(L),Maintainability(M), Operability(O),Portability(P),Precision(PRC),Reliability(R),Security(S),Stability(STBL), Time and resource based efficiency(TRBE),Usability(U),UI Aesthetic(UIA).

CHAPTER 6

DISCUSSION/RECOMMENDATION

This chapter further discusses the results discussed in the previous chapter. Based on these discussions, this chapter briefly summarizes the study, provides the conclusions of this research. In the end, it describes briefly the issues of limitation and future research.

6.1 Discussion of Results:

This research identified 25 major SQ dimensions which exist in SQ literature. Upon detailed review it became evident that there is a commonality among different SQ models proposed by different authors. Therefore a framework was developed based on the concept of commonality and it was fine tuned.

Data collected was analyzed using ANOVA model in SAS. Results show that Consistency, backward/forward compatibility, Layout, Portability and Time and resource based efficiency were the dimension impacted by the software category. While Consistency, Maintainability, reliability, Security, Usability, and UI Aesthetics were significantly influenced by the user group.

Also, thirteen dimensions form are the must dimension. They cannot be ignored while developing a high SQ software/application. In other words, these dimensions are critical or important across any software.

The means plots have identified that Experts users feel that softwares are maintainable, consistent and secure enough than Novice users.

Software Type	**Dependent Variable**	**Regression Model**
MS WORD	OSQ	1.035 + (0.72)ACC – (0.86)S + (0.54)INTR+ (0.26)U – (0.29)STBL
	SR	2.6 – 0.152 (TRBE)-0.205(U) + 0.175(INTR) -0 .18(HD)
MINITAB	OSQ	(-0.02804) + (0.371) U + (0.143)L + (0.27)C– (0.246) R + (0.19)UNDR + (0.20)A
	SR	2.171 – (0.256) O+ (0.294)P – (0.207) STBL + (0.163) CON
MS OUTLOOK	OSQ	(-.9522)+(0.47)O+ (0.28)UIA+(0.33)FUN+(0.29)M
	SR	(-.70444) +(0 .389)STBL+(0.39)AVL +(0.260)TRBE + (0.241)PRC – (0.22)DOC+(.141)R
GOOGLE SKETCH	OSQ	(-.2761) + 0.79 ACC + 0.46 M -0.31 BFC + 0.19STBL - 0.221 O +0.16 PRC
	SR	(-2.05) + (0.30)S + (0.46)TRBE + (.93)HD+(0.42)U + (0.357)PRC– (0.32)AVL

Table 6-1. Best fit regression model for MS Word, Minitab, MS Outlook and Google sketch.

The regression analysis on MS WORD software shows that *OSQ* is described by *Accessibility, Security, Interoperability, Usability and Stability* and SR is described by *Time and resource based efficiency, Usability, Interoperability and Hardware dependency.*

For Minitab software, the results show that *Layout, Security, Interoperability, Usability and Stability describes* OSQ and whereas *Operability, Stability, Portability and Consistency* describes SR.

In case of MS Outlook software, the results show that the OSQ was significantly affected by *Functionality, Operability, UIA, and Maintainability* and SR is affected by *Availability, Stability, Precision, TRBE, Documentation, and Reliability.*

For Google Sketch software, the results show that the OSQ was significantly affected by *Accessibility, Maintainability, BFC, Stability, Operability, and Precision* while the SR is affected by *Security, Hardware Dependency, Usability, Precision, TRBE, and Availability.*

6.2 Overall Discussion:

The objective for this study was as follows:

1. Identify the SQ dimensions that are relevant from the user perspective.

   After an extensive literature review, this study has found the most commonly occurring SQ dimensions. After identifying the most frequently occurring 21 SQ dimensions, this study perceived these dimensions from the end users' perspective. Keeping this perspective, two new sub characteristics 'Layout', and 'Interface Aesthetic' were proposed as new additions to the list. And finally, after the pilot study, the number of dimensions was increased to 25. This research identified two new software quality dimensions.

2. Do the SQ dimensions behave differently across different user level?

   The survey instrument was developed to collect the responses of end user which helped to understand the user perspective. As discussed in chapter 5, ANOVA was performed and between the subject design was used to identify the significant dimensions that affect the software quality. The mean values of these significant dimensions were plotted against the software type and user level. From the mean values, it was identified that the presence of SQ dimensions varies, with different user level.

3. What are the SQ dimensions relevant for MS WORD, MINITAB, MS OUTLOOK and GOOGLE SKETCH?

   The stepwise and enter method regression analysis was performed on the dimensions, considering dimensions as independent variables and OSQ and SR are considered to be dependent variable. The significant regression models considered for each software is mentioned in figure 5-5 and figure 5-6 respectively.

6.3 Recommendations:

The attributes which are significantly contributing in measuring the software quality are demonstrated using four different types of software. The software and their respective attributes are mentioned as following.

- For MS WORD software rating, *Time and resource based efficiency, Usability, Interoperability and Hardware dependency* are important while for improving the OSQ of MS WORD, *Accessibility, Security, Interoperability, Usability and Stability* are important.
- In case of MINITAB software rating, *Operability, Stability, Portability and Consistency and* MINITAB *OSQ Layout, Security, Interoperability, Usability and Stability* are important.
- For MS OUTLOOK software rating, *Availability, Stability, Precision, TRBE, Documentation, and Reliability* while for OSQ of MS OUTLOOK, *Functionality, Operability, UIA, and Maintainability* are important.

- In case of GOOGLE SKETCH software rating, *Security*, *Hardware Dependency*, *Usability, Precision, TRBE, and Availability* and to improve OSQ, *Accessibility*, *Maintainability*, *BFC, Stability, Operability, and Precision* are important.

References

Adnan, R., and Baseem, M. (2006). *A New Software Quality Model for COTS Components,* in the proceedings of Journal of Computer Science *2*(4), Pages 373-381, May, 2006.

Alvaro, A., Aleida, A., and Meira, S. (2005). *Quality attributes for a Component Quality model*, in the proceedings of 10th WCOP/19th ECCOP, Glasgow, Scotland, 2005.

Avadhesh, K., Grover, P., and Rajesh, K., (2009*)*. *A quantitative Evaluation of Aspect-Oriented Software Quality Model* in the proceedings of ACM SIGSOFT Software Engineering Notes, Volume 34, Number 5, Pages 1-9, New York, USA, September 2009.

Azuma, M.(1996). *Software Products Evaluation System: Quality Model, Metrics and Process* in the proceedings of International Standards and Japanese Practice, Inf. & Software Tech., Elsevier, Vol.38, No. 3, Pages 145-154, Tokyo, Japan.

Bertoa, M., Vallecillo, A.(2002). *Quality Attributes for COTS Components,* in the proceedings of the 6th International ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering, Pages 1-11, Malaga, Spain, October, 2002.

Boehm, B., Brown, J., and Lipow, M. (1976) *Quantitative evaluation of software quality*, in the Proceedings of ICSE '76 2nd International Conference on Software Engineering, Pages 592-605, California, USA, 1976.

Chang, C., Wu, C., and Lin, H., (2008) *Integrating Fuzzy Theory and Hierarchy Concepts to evaluate Software Quality*, in the proceedings of Software Quality Control, Vol. 16, Issue.2, Pages 263-276, June 2008.

Deutsch, M., and Wills, R.(1988). *Software Quality engineering: A total technical and management approach*, (Taylor, .J) Prentic- Hall, Inc., New Jersey, USA, 1988.

Dromey, R. (1995). *A model for software product quality*, in the proceedings of IEEE Transactions on software engineering, Vol. 21, Pages 146-162, Brisbane, Australia.

Fitzpatrick, R. (2003). *The Software Quality Star: A conceptual model for the software quality curriculum,* in the proceedings of Closing the Gaps: Software Engineering and Human Computer Interaction, INTERACT 2003: 9th IFIP TC 13 International Conference on Human-Computer Interaction, Pages 205-315, Zurich, Switzerland, January, 2004.

Fitzpatrick, R., Smith, P., and O'Shea, B.,(2004). *Software Quality Revisited*, in the proceedings of the Software Measurement European Forum, Istituto di Ricerca Internazionale S.r.l., Pages 307/315, Milan, Italy. 2004

Genichi, T., Elsayed, A. and Thomas, H., (1989). *Quality engineering in productions systems,* McGraw-Hill, New York, USA, P2/3, 1989.

Georgios, G., Vassilios, K., Konstantinos, S., Panagiotis, L., Vasileios, V. and Diomidis S. (2007). *Software quality assessment of open source software* in the Proceedings of the 11th PanHellenic Conference on Informatics, volume A, pages 303–315, Athens, May 2007.

Gill, N. (2005). *Factors Affecting Effective Software Quality Management Revisited*, ACM SIGSOFT Software Engineering Notes Volume 30, Issue 2, New York, USA, March 2012.

Gilmore, H. (1974,) *Product Conformance Cost*, Quality progress 7, no.5 Pages 16-19, USA, December, 1974.

Grady, R. and Caswell, D. (1987). *Software Metrics: Establishing a Companywide Program*, Englewood cliffs, NJ, USA, 1987.

IEEE.(2001). *Guide to the Software Engineering Body of Knowledge*, www.swebok.org. 2001.

International Standard, ISO/IEC 926-1 (2001), *Quality Model 2001: Part 1, 2, 3*. Institute of Electrical and Electronics Engineering, USA.

ISO, I., & FCD, I. (2004). *25000, Software Engineering-Software Product Quality Requirements and Evaluation (SQuaRE)-Guide to SQuaRE. International Organization for Standardization. Geneva, May 2004.*

Juran, J., (1989). *Juran on Leadership for Quality*. New York, USA. 1989

Kan, S.(2003). *Metrics and Models in Software Quality Engineering, 2/e*. Pearson Education India, 2003

Kandhari, J. (2010). *Information Quality on the World Wide Web: A user perspective*, a master's thesis at University of Nebraska, Lincoln, Nebraska, May 2010.

Kashayar, K. and Gueheneuc, Y.(2004). *A quality model for design patterns, Masters' thesis, Laboratory of Open Distributed systems and software engineering*, a thesis at University of Montreal, Montreal, Quebec. December 2004.

Krause P., Freimut B., and Suryn W.(2002) *New Directions in Measurement for Software Quality Control* in the proceedings of 10th International Workshop on Software Technology and Engineering Practice - STEP 2002, Page 129-143, Montréal (Canada), Guildford, UK, October, 2002

McCall, J., Richards, P. and Walters, G.(1977*)*. *Factors in software quality*, General Electric, National Technical Information Service., New York, USA. 1977.

Parasuraman, A., Zeithaml, V., Berry, L.(1988), *SERVQUAL: A Multiple-Item Scale for Measuring Consumer Perceptions of Service Quality.in the proceedings of* Journal of Retailing, Vol. 64 Number 1, Pages 141-157, January, 2004.

Powell, T., Jones, D., & Cutts, D. (1998). *Web site engineering: beyond Web page design*. Prentice-Hall, Inc. New Jersey, USA, 1998.

Saida, B., Khaled, E., Goyal, N., and Shesh, R. (2000). *Thresholds for object oriented measures,* in Proceedings of the 11th International Symposium on Software Reliability Engineering (ISSRE), Pages 24-38, California, USA, October, 2000.

Sharma, A., Rajesh, K., and Grover, P. (2008). *Estimation of quality for software components: an empirical approach*, in the proceedings of ACM, SIGSOFT Software Engineering Notes, Vol.33 Issue. 6, Pages 1-10, New York, USA, November 2008.

Suryn, W., Abran, A., Bourque, P., and Laporte, C., (2002). *Software Product Quality Practices, Quality Measurement and Evaluation using TL9000 and ISO/IEC 9126*, in the proceedings of 10th International Workshop on Software Technology and Engineering Practice - STEP 2002, Pages 156-160, Montreal, Canada, October 2002.

Tomar, A., and Thakare, V., (2011). *A systematic study of software quality models*, in the proceedings of International Journal of Software Engineering and Applications, Vol.2, Pages 61-70, No.4. October 2011.

Web Reference: http://www.qualitydigest.com/mar08/articles/03_article.shtml

Word, W., and Venkatramn, B. (1999). *Some Observations on Software Quality* in the proceedings of the 37th Annual Southeast regional conference, Page 2, New York, USA. April, 1999.

Appendix A:

https://www.surveymonkey.com/s/VJ2S6N3