*J. Appl. Environ. Biol. Sci.*, 4(9S)403-406, 2014



ISSN: 2090-4274
Journal of Applied Environmental and Biological Sciences
*www.textroad.com*

# Software Quality Measurement: (A New Statistical Approach)

**Uzma Sattar, Kashif Razzaq, Salman Afsar Awan**

Department of Computer Science, University of Agriculture, Faisalabad

*Received: September 12, 2014*
*Accepted: November 23, 2014*

**ABSTRACT**

Software Quality Measurement is a term to quantify that how much a system or software can fulfill the requirements and desired characteristics. We can check it to apply quantitative and qualitative research or can also apply both at the same time. The paper shows two types of measurement theories; these are discussed just to show that how to measure the quality of software. This measurement is based on the representation. These theories lead to develop a powerful tool which is used to measure values and operations. Here three types of assumptions also discussed in the paper.

**KEYWORDS**: Software Quality, Measurement, Quantitative, Qualitative, Measurement Theories, Assumptions, Statistical Approaches, Prediction.

## 1 INTRODUCTION

When there is need to define software engineering in Business context,[7] software quality must be measured into two notions. Functional quality and structural quality.

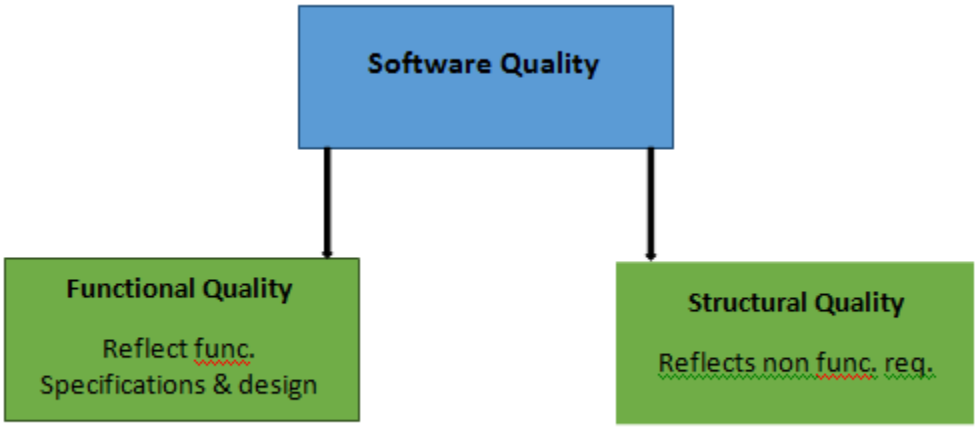


FIG: 1 Software Quality in Business Context

Software Quality can define into two different but almost related ways in the context of Software Engineering,[6] which defined in the point of business. These two ways are functional point of view and structural point of view. Software functional quality is based on functional requirements and specifications of software. If these are measured and gathered through a proper process and criteria then the functional point of view is covered. Any software quality is basically depends heavily on the functional requirements[2]. In the case of structural quality means how well the nonfunctional requirements are achieved and delivered. This phase to gather and specify the nonfunctional requirements in a qualitative format is most critical part of any software development. It means nonfunctional requirements are more critical than the functional requirements[3]. To achieve the maximum quality of any type of software, it needs to develop a matured system. There are two reasons are involved in measuring the quality of a software. These are risk management and cost management. in risk management if functional and nonfunctional requirements are not gathered in a quality format then it leads towards increment in system failure. These failures are occurred before and after the delivery of the system which also annoyed the user.

* **Corresponding Author:** Uzma Sattar, Department of Computer Science, University of Agriculture, Faisalabad. uzmasattar@uaf.edu.pk

This risk also leads towards cost increment and in this situation handling the cost management becomes more critical.

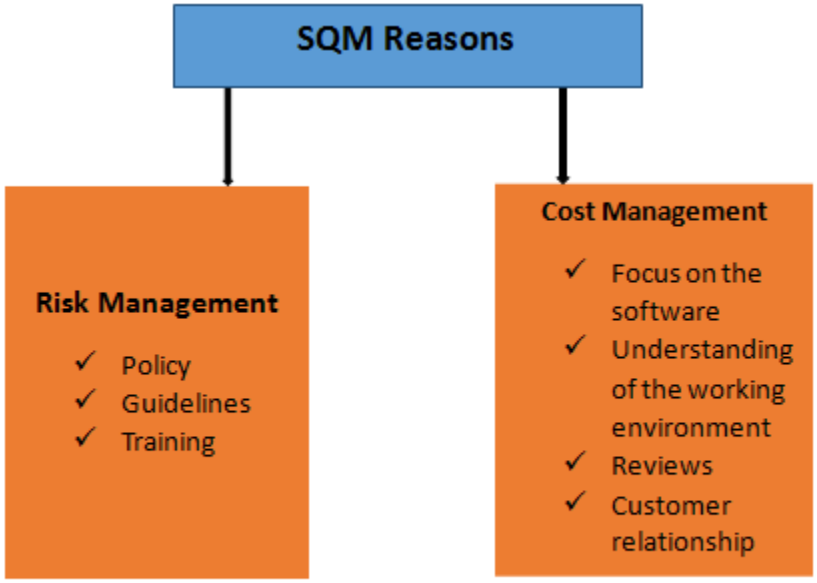


FIG: 2 Software Quality Measurement Reasons

The assumption one is refers that in software quality measurement the related persons should be involved nearly the same experimental relational system.

The second is the standard of measurement should be widely accepted for software quality. Although not any system can be exactly same with that standard or measurement scale.

The third one is any complex system can be subdivided into less complex system or sub modules and then the whole complex one would be solved or measured by the help of less complex modules.

Maturity of science depends on the measuring tools uses to measure the quality of any system or software scientifically. This measuring maturity is mandatory to measure the quality due to two reasons that shows in FIG 1.

**1.1. Risk Management:** software errors are due to human faults[9]. This may be poor interface and programming error. An example of a programming error is wrong logic etc. Requirements gathering phase is also one of the most important phase.

**1.2. Cost Management:** nonfunctional requirements are as important as functional because application with good structural software quality always costs less. It also provides batter approach to maintain and is ease of understanding.

## 2. BRIEF INTRODUCTION TO MEASUREMENT THEORY:

In statistical analysis, a measure on a set is a organized way to assign a number to each suitable division of that set, naturally interpreted as its volume. In this logic, a measure is a simplification of the concepts of extent, area, and degree.

In mathematical analysis applied mathematics has a branch called measurement theory uses to analysis of data but as discussed earlier that measurement results are not exactly same according to expectations and measurement scale [8]. So if any conclusion needed to be drawn of attribute being measured, compromise needed between attribute and measurement. This theory helps us to keep away from meaningless statements.

## 3. MEASUREMENT SCALES:

Special scales of measurement are use having standard properties[10]. These properties can distinguish these scales to one another. Some commonly discussed properties are.

**3.1. Identity:** On the scale every value has a unique meaning identity and have no inherently relationship to the attribute or variable to which that value assigned.
**3.2. Magnitude:** On the measurement scale every value has a relationship to other values. That relation can b in any order, like some values are larger than others and some are smaller.
**3.4. Equal Intervals:** On the scale, units are equal to one another. This means that the difference between 1 and 2 is equal to the difference between 19 and 20.
**3.5. A minimum value of zero:_** It must be ensure that on scale minimum value or starting value must be zero. With the context of quality checking all the modules should be test at same level with minium and maximum variations.

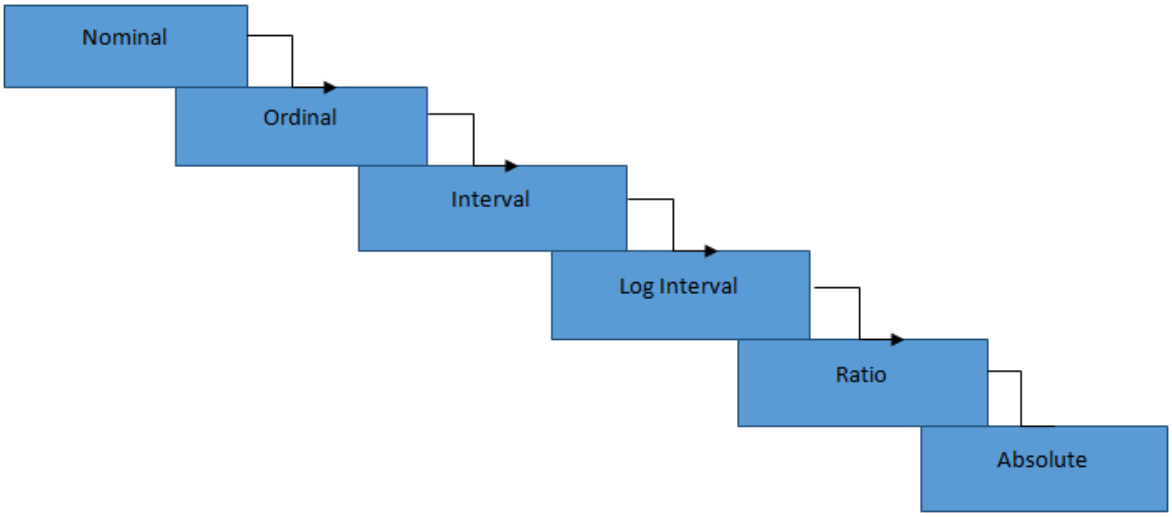


FIG: 3 Measurement levels

## 4.PRECONDITIONS FOR THE MEASUREMENT OF SOFTWARE QUALITY

Testing of software can be done at any stage. It is actually a process of validating and verifying the quality of any product in terms of computer science and other fields of life. etc., any manufacturing product [1]. It is also applicable at raw level of any product. Here we discuss three types of measurements on the bases of how we define software quality. Continuous measurement, anticipated measurement, Requirement wise measurement[4]. It is a sensitive issue to implement software testing with all constraints of quality even at initial stage. it is the main reason of defects and errors that turns into major causes at time of delivery and are difficult to catch during testing. Moreover non-functional magnitude of product's quality is highly measurable and cannot be overlooked at any stage.

## 5. THEORY OF MEASUREMENT AND QUALITY DEFINITIONS

Some software quality definitions discussed here: Quality is the main attribute of any product whom which no compromise could be made. In quality measurement three factors play lead role. These factors actually complete the definition of quality. Time, Cost, Performance. These are the major causes or factors due to which the quality can be accessed and fail too. Performance is the sensitive issue it is actually the reason of any product to be in market and business. While achieving the quality time and cost can be replace with each other for best performance. User satisfaction can be achieve when time, cost and performance are applied with best combination on any product specifically in software industry where end user does not know his or her exact demands, functional and non-functional requirements, business practices technically. It is a big challenge to satisfy user according to his or her requirements in the above scenerio. As we discuss above about risk. Organization must plane risk management policy with mission and vision statement. To start work on any product risk analysis should be done according to the risk management policy for achieving goals regarding quality. It also overcome errors and unexpected behavior.

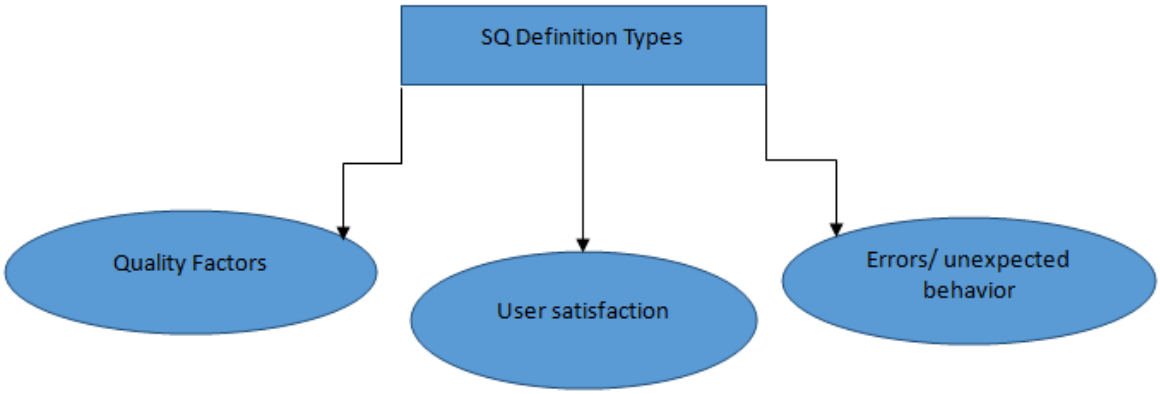


FIG 4: Software Quality Definitions

## 6. CONCLUSION AND RECOMMENDATIONS

However it is not possible to inaugurate experimental dealings among all the quality features and quality. The need is to design a better prediction system for measurement of quality and its related features like end user requirements, that must not be flouted and all quality lines should be in control and consistent. Previous study endorses us that it is not needed to develop a broadly used software quality environment. The focus must be on founding experimental interpersonal schemes and procedures having more scales of quality measurements connected with the characteristics of software. The measurement philosophy should practically applicable and in written document in edict to distinguish that what has to be measure and how to deduce the outcomes.

## REFERENCES

1. Honglei, T., Wei, S., &Yanan, Z. (2009). The Research on Software Metrics and Software Complexity Metrics. *International Forum on Computer Science-Technology and Applications* (Vol.1, pp. 131-136).
2. Issac, G., Rajendran, C. &Anatharaman, R, N. (2004). A Conceptual Framework for Total Quality Management in Software Organizations. *Total Quality Management* (Vol. 15, pp. 307-344).
3. Duewer, DL., Liu, H-K., & Reeder, DJ. (1999). Graphical Tools for RFLP Measurement Quality Assurance: Single-Locus Charts*. *Journal of Forensic Sciences* (Vol. 5, pp. 969-977)
4. Conger, S. (2013). Service Quality Measurement in a Service Desk Environment. *An itSMF USA ColdFusion '13 White Paper*. (pp. 1-8)
5. Schneidewind, NF. (2002). Body of knowledge for software quality measurement. *IEEE* (Vol. 35, pp. 77-83)
6. Genova, G., Fuentes, JM., Uorens, J., Hortadao, O., & Moreno, V. (2013). *Requirement Engineering*. A framework to measure and improve the quality of textual requirement. (Vol. 1, pp. 1-2)
7. BasiliVR., et al., (2010). Linking Software Development and Business Strategy Through Measurement. *IEEE* (Vol. 43, pp. 57-65)
8. Peng, Y., Kou, G., Wang G., Wu, W., & Shi, Y. (2011). Ensemble of Software Defect Predictors: An Ahp-Based Evaluation Method. *International Journal of Information Technology & Decision Making* (Vol. 10, pp.187-206)
9. Kriplean, T., Portner, C., Landay, J. (2011).Utility of human-computer interactions: toward a science of preference measurement. *CHI '11 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. (Vol. 1, pp. 2275-2284)
10. Villegas, NM., Muller, HA., Tamura, G., Duchien, L., Casallas, L. (2011). A framework for evaluating quality-driven self-adaptive software systems. *SEAMS '11 Proceedings of the 6th International Symposium on Software Engineering for Adaptive and Self-Managing Systems* (Vol. 1, pp. 80-89)