# FRAMEWORK TO ASSESS SOFTWARE QUALITY IN ERP SYSTEMS

***C. Otieno, W. Mwangi and S. Kimani***
*Institute of Computer Science and Information Technology, Jomo Kenyatta University of Agriculture and Technology, Nairobi, Kenya*
*E-mail:otienocalvins22@gmail.com*

**Abstract**
Doing business in African continent requires customer satisfaction and efficiency of service. To do these every company must invest in Enterprise Resource Planning system software that can support business goals and growth. One major challenge that most enterprises face in the country is to acquire correct and reliable ERP systems that can be used in their Enterprises. The major objective of this research paper is to propose a framework tool that enterprise IT systems manager or leaders can use to assess the software systems to determine their quality before approval and acquisition by their organizations. The methodology involved in this research will be literature review, data collection, analysis and design and proposal of the framework. From the data collected and analyzed it was discovered that for the software quality index of 0-1.7 indicates low quality, 1.8-2.4 indicates average quality and 2.5-3 indicates high quality. The major contributing factors to high quality is that the attributes are built and considered into the system then the production process is being improved and therefore quality checks and control are being constructed into the system resulting in better quality. From the assessment results it was noted that the framework model managed to achieve 90% success and therefore it can be concluded that it can be adopted for use in assessing quality of software system.

**Key words:** Software quality index, framework model, ERP, quality

## 1 Introduction

Framework development is a multifaceted endeavor undertaken to promote reuse of software within a family of related applications (Kruchten, 2007). Traditional approaches involve either the evolution or the systematic design of the needed generic structure. Frameworks can improve developer productivity and improve the quality, reliability and robustness of new software. (D. Garvin, 2010). Kruchten (2007) has proposes a four factor modified framework to address software safety. The four factors relating to software safety in his framework which are part of the original McCall framework are: Correctness, efficiency, reliability, testability. To these four quality factors, a new factor-responsiveness was introduced to account for the real time performance. For each factor the corresponding criteria attributes from the developer point of view. It is argued that determination and application of specification, design, coding and testing methods in a project should be based on the metrics derived from the criteria in order to "ensure" software safety. (Kruchten, 2007) expands FURPS into FURPS+, where the '+' indicates such requirements as design constraints, implementation requirements, interface requirements and physical requirements (Jacobson *et al.*, 2008).

### 1.2 Problem Statement

In Kenya rapid development of software that are poorly designed has made major organizations such as universities, banks, health and Saccos to experience operational challenges especially when their operations are brought to a halt(Daily Nation, pg56, 2008). Huizinga Dorota (2007) describes software defects as main source of errors introduced during the design process. The estimation for losses in organization is estimated to be of equivalent of $26billion annually (Huizinga Dorota, 2007).

The IEEE international conference and on IEEE Annals of the History of Computing, Vol 22 Issue 1, 2000 has discussed on potential losses and defects on operations caused by software bugs to various manufactures. For example; According NASA, space shuttle a crash on space shuttle was as a result of poor design on conversion of distances from kilometers per hour to miles per hour due to poor design conversion procedures this was caused because of poor design which didn't capture distance conversion elements to the programming logic implementation. (IEEE, 2004).

In November 2000 at the National Cancer Institute, Panama City. In a series of accidents, therapy planning software created by Multidata Systems International, a U.S. firm, miscalculates the proper dosage of radiation for patients undergoing radiation therapy. At least eight patients die, while another 20 receive overdoses likely to cause significant health problems(John. D, 2006). The physicians, who were legally required to double-check the computer's calculations by hand, are indicted for murder (Huizinga *at al*., 2007).

According to Wide Open Source Conference on Open source system report, It was discovered that there is no standard way or formula for evaluating and measuring of the relevance of design to the user requirements and overall goal on the functionality of the software system in the industry. The overall cost of these damages has greatly hampered and added cost to maintainability processes to computers. The cost and time period associated with maintainability has been of high net worth value to the service consumers.ISO 2000 quality framework describes software design quality as the set of attributes that bear on the effort needed to make specified.

### 1.3 Proposed Solution

To overcome the above industrial problem, this research work proposes to improve the Dromey software quality framework by developing an improved framework that implements quantitative frameworks of quality index and thresholds for every quality factor analyzed to present aspect of

software quality. The proposed framework will test specific attributes of the software quality factor to evaluate its presence in the software system and to estimate the relevance of the design to user needs and requirements stated to the overall goal of the system.

## 2 Research Methodology

Adoption of research process was necessary to carry out this project. Data as regards quality factors deemed important by users was collected from the industry, Business organizations were the major respondent in this case since they are the largest consumers of software in the country. The information they provided was the primary data for this project.

### 2.1 Sample Population:

The sample population for this research included software and interviewed up to a maximum of 50 per group the sample population are majorly software developers and anlyst around nairobi. The groups were selected randomly to allow for wider probability of picking any organization in the success category and any application developed for business.

### 2.2 Data Collection Methods

The proposed method for data collection in this project was a questionnaire**.** They are the only feasible way to reach a number of reviewers large enough to allow statistically analysis of the results.

### 2.3 Data Collection Instrument

The proposed data collection instrument for this research was a design likert scale that allowed me to assign numbers 1-5 to collect both qualitative and quantitative data about the business systems from the sample population. Likert scale was chosen because of its straight forward nature, ease of analysis of data. An open ended questionnaire also accompanied the likert scale to allow for collection of qualitative data on the general feelings of the sample population about the quality factors of the systems in use today.

### 2.4 Data Analysis Method

The method of data analysis to adopted in the research process was correlation statistics and multivariate analysis of variables.This was because it was noted that various data variables were collected en and therefore analysis of them was necessary.

### 2.**5** Hypothesis

The proposed hypothesis that guided this study were

a. The current framework to determine the software quality factors in design process of software engineering are not effective and are not effectively being used.
b. System analyst do not engage in evaluation of quality factors in the software system they develop
c. Software systems developed in Kenya do not meet the requirements stated in the requirements collection face
d. There is no match between the requirements collected, the design process and product quality of software systems

## 3 Literature Review

In the past years the scientific and industrial communities have proposed many QA standards and frameworks. According to Andres Sousa-Poza (2009) "there are more than 300 frameworks developed and maintained by more than 50 different organizations." Most popular framework is ISO/IEC 9126

which specify requirements for a quality management system within an organization. Some of the previously developed frameworks are:-

### 3.1 ISO/IEC TR 9126-2

The frameworks listed in ISO/IEC TR 9126-2 are not intended to be an exhaustive set. Users can
Select or modify and apply framework and measures from ISO/IEC TR 9126-2:2003 or may define application-specific framework for their individual application domain. Software frameworks are the only mechanized tools for assessing the value of internal attributes (B. Kitchenham et all, 2003). Software frameworks are defined as "standard measurements, used to judge the attributes of something being measured, such as quality or complexity, in an objective manner" (ISO2009).

### 3.2 ISO/IEC 9000:2005

ISO/IEC 9000:2005 provides guidance for the use of the series of International Standards, Named Software product Quality Requirements and Evaluation (SQuaRE). Software Quality in the Development Process (SQUID) allows the specification, planning, evaluation and control of software quality through the software development process. ( Georgiadou, 2003) SQUID uses external and internal quality measures defined in ISO 9126. Although the existence of documentation is a key requirement of a functional ISO 9001 Quality Management System (QMS), it is not in itself sufficient. To develop and implement a fully functional ISO 9001 QMS, it is essential that a small/medium-sized enterprises correctly identifies the initial state of its QMS and the path it will follow to achieve the desired state (. Georgiadou, 2003).

### 3.3. Capability Maturity Framework

Capability Maturity Framework (CMM) proposed by Software Engineering Institute (SEI) provides a framework for continuous software process improvement ( Georgiadou, 2003). The key notion is that CMM provides guidelines for conducting audits, testing activities, and for process improvement. The CMM approach classifies the maturity of the software organization and practices into five levels describing an evolutionary process from chaos to disciplines Initial (chaotic), Repeatable (project management), Defined (institutionalized), Managed (quantified), Optimizing (process improvement)( Suryn,2003).

### 3.4 Mccall's Framework

McCall's framework ( Suryn,2003).) of software quality incorporates 11 criteria encompassing three main perspectives for characterizing the quality attributes of a software product. These perspectives are Product revision (ability to change), Product transition (adaptability to new environments), and Product operations (basic operational characteristics) ( Suryn,2003).

### 3.5 Boehm's Framework

Boehm's framework (Skelton, 2006) is based on a wider range of characteristics and incorporates 19 criteria. At the highest level of his framework, Boehm defined three primary uses (basic software requirements), these three primary uses are as-is utility, the extent to which the as-is software can be used (i.e. ease of use, reliability and efficiency), Maintainability, ease of identifying what needs to be changed as well as ease of modification and retesting, Portability, ease of changing software to accommodate anew environment (Khosravi, & Gueheneuc, 2004)

## 4 Data Collection and Analysis

The following data items were collected from the field. They were classified to generate matrix with regard to every quality factor collected.

**Maintainability Matrix**

| | | | | | | |
|---|---|---|---|---|---|---|
| *0* | 0 | 1 | 1 | 2 | 3 | 43 |
| *0* | 0 | 2 | 1 | 3 | 6 | 38 |
| 4 | 9 | 13 | 15 | 4 | 3 | 2 |
| 2 | 1 | 3 | 4 | 11 | 10 | 19 |

**Completeness Matrix**

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 2 | 1 | 3 | 10 | 15 | 9 |
| 1 | 1 | 3 | | 5 | 23 | 7 |
| 0 | 0 | 0 | 1 | 3 | 5 | 42 |
| 0 | 0 | 0 | 0 | 3 | 7 | 30 |

**Traceability Matrix**

| | | | | | | |
|---|---|---|---|---|---|---|
| 12 | 8 | 4 | 3 | 7 | 9 | 6 |
| 4 | 5 | 6 | 7 | 9 | 11 | 8 |
| 7 | 9 | 8 | 1 | 4 | 8 | 13 |
| 6 | 5 | 7 | 3 | 9 | 12 | 11 |

**Understability Matrix**

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | 1 | 5 | 5 | 15 | 24 | 49 |
| 2 | 12 | 15 | 19 | 31 | 36 | 35 |
| 0 | 1 | 2 | 1 | 14 | 14 | 18 |
| 1 | 5 | 6 | 1 | 7 | 8 | 12 |

**Usability Matrix**

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | 3 | 4 | 4 | 3 | 10 | 25 |
| 1 | 1 | 2 | 2 | 3 | 9 | 32 |
| 9 | 1 | 5 | 7 | 8 | 25 | 45 |
| 2 | 3 | 3 | 6 | 9 | 14 | 13 |

**Efficiency Matrix**

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 0 | 3 | 6 | 6 | 13 | 21 |
| 0 | 0 | 2 | 1 | 5 | 10 | 32 |
| 6 | 4 | 1 | 5 | 7 | 11 | 18 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 4 | 5 | 11 | 27 |

**Reliability Matrix**

| | | | | | | |
|---|---|---|---|---|---|---|
| 5 | 6 | 6 | 4 | 5 | 6 | 6 |
| 5 | 4 | 4 | 6 | 7 | 7 | 8 |
| 6 | 7 | 5 | 2 | 5 | 6 | 10 |
| 5 | 4 | 4 | 5 | 4 | 11 | 8 |

Considering the above population samples of factors represented in matrix form, we generate population mean of $\overline{x}$,

| P1 | P2 | P3 | P4 | P5 | P6 | P7 |
|---|---|---|---|---|---|---|
| 7.71 | 5.7 | 7 | 14.3 | 7.29 | 7.14 | 5.43 |
| 8.57 | 5.8 | 7.14 | 21.4 | 7.14 | 7.14 | 5.9 |
| 7.14 | 7 | 7.14 | 7.14 | 14.2 | 7.52 | 5.9 |
| 7.43 | 5.7 | 7.57 | 5.7 | 7.14 | 7.14 | 5.9 |

And an overall mean for every treatment values as

OVERAL MEAN $\overline{x}$

7.82

9.02

8.05

6.61

Obs=mean+ treatment effect +residual will generate the matrix shown below

obs

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 2 | 3 | 43 |
| 2 | 1 | 3 | 10 | 15 | 9 |
| 8 | 4 | 3 | 7 | 9 | 6 |
| 1 | 5 | 5 | 15 | 24 | 49 |
| 3 | 4 | 4 | 3 | 10 | 25 |
| 0 | 3 | 6 | 6 | 13 | 21 |
| 6 | 6 | 4 | 5 | 6 | 6 |

$=$

mean

| | | |
|---|---|---|
| 7.82 | ..... | ...7.82 |
| | | |
| | | |
| 7.82 | ..... | ....7.82 |

$+$

treatment

| | | |
|---|---|---|
| -0.11 | | -0.11 |
| -2.12 | ..... | -2.12 |
| -0.82 | ..... | -0.82 |
| -0.82 | ..... | -0.82 |
| -0.53 | ..... | -0.53 |
| -0.68 | ..... | -0.68 |
| -2.39 | ..... | -2.39 |

$+$

residual error

| | | | | | | |
|---|---|---|---|---|---|---|
| -7.71 | -7.71 | -5.71 | -5.71 | -3.7 | -4.71 | 35.29 |
| -5.7 | -3.7 | -4.7 | -2.7 | 4.3 | 9.3 | 3.3 |
| 5 | 2 | -3 | -4 | 0 | 2 | -1 |
| -12.3 | -13.3 | -9.3 | -9.3 | 0.7 | 9.7 | 34.7 |
| -5.29 | -4.29 | -3.29 | -3.29 | -4.29 | 2.71 | 17.71 |
| -6.14 | -7.74 | -4.14 | -1.14 | -1.14 | 5.86 | 13.86 |
| -0.043 | 0.57 | 0.57 | -1.43 | -0.43 | 0.57 | 0.57 |

Squaring the observation. the mean, the treatment and residual error the matrix below is obtained

square of observations

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 4 | 9 | 1849 |
| 4 | 1 | 9 | 100 | 225 | 81 |
| 64 | 16 | 9 | 49 | 81 | 36 |
| 1 | 25 | 25 | 225 | 576 | 2401 |
| 9 | 16 | 16 | 9 | 100 | 625 |
| 0 | 9 | 36 | 36 | 169 | 441 |
| 36 | 36 | 16 | 25 | 36 | 36 |

=

square of mean

| | |
|---|---|
| 61.1524 | 61.1524 |
| 0 | 0 |
| 0 | 0 |
| 0 | 0 |
| 0 | 0 |
| 61.1524 | 61.1524 |
| 0 | |

+

square of treament

| | |
|---|---|
| 0.0121 | 0.0121 |
| 4.4944 | 4.4944 |
| 0.6724 | 0.6724 |
| 0.6724 | 0.6724 |
| 0.2809 | 0.2809 |
| 0.4624 | 0.4624 |
| 5.7121 | 5.7121 |

square of residual

| | | | | | | |
|---|---|---|---|---|---|---|
| 59.4441 | 59.4441 | 32.6041 | 32.6041 | 13.69 | 22.1841 | 1245.384 |
| 32.49 | 13.69 | 22.09 | 7.29 | 18.49 | 86.49 | 10.89 |
| 25 | 4 | 9 | 16 | 0 | 4 | 1 |
| 151.29 | 176.89 | 86.49 | 86.49 | 0.49 | 94.09 | 1204.09 |
| 27.9841 | 18.4041 | 10.8241 | 10.8241 | 18.4041 | 7.3441 | 313.6441 |
| 37.6996 | 59.9076 | 17.1396 | 1.2996 | 1.2996 | 34.3396 | 192.0996 |
| 0.001849 | 0.3249 | 0.3249 | 2.0449 | 0.1849 | 0.3249 | 0.3249 |

SSob=0+0+…….36=7639
SSmean=61.15+61015+….61.15=2996.35
SStrt=0.0121+0.0121+……..5.712=375.3671
SSresidual=59.441+59.44+….0.325=4349.65
Where SSob is the sumof square of the observation, SSmean is the sum of the square of the population sample mean, SStrt is the sum of the square of the treatment effect and SS residual is the sum of the square of the residual effect. The same process uis repeated for all the sample population data, the population mean and overall mean of the sample population data. Since the sample covariance of the matrix for the Lth sample is Sl, we test the hypothesis of the number of treatment effects H1:To=T2=T3 against the alternate H2:T1≠T2……Tg where g is the number of the sample population. In this research g is 7.We reject the hypothesis H0:T1=T2=T3…..Tg if the ratio of the generalized variables is too small given by wicks lambda

$$\Lambda^{*} = \frac{/w/}{/B+W/}$$

Where the treatment is B, The residual is W and total cross product of the matrix is given by B+W.
From the evaluations in the matrices above the manova table becomes

$$B = \begin{bmatrix} 375.367 & 8475.59 \\ 324.2946 & 30.2127 \end{bmatrix}$$

$$w = \begin{bmatrix} 4349.65 & 3983.739 \\ 5186 & 1868.349 \end{bmatrix}$$

$$\Lambda^{*} = \frac{-12533998.6}{-4100813426}$$

$$\Lambda^{*} = 0.30564664$$

Suppose the test is tested of ά=0.01 level of significance ,we test the statistics to compute

$$2\left[\frac{1-\sqrt{\Lambda^{*}}}{\sqrt{\Lambda^{*}}}\right]\left[\left[\sum^{n-g-1}\right]\right]$$

$$g-1$$

Where g=7 the number of population in this study lambda is 0.03056466 We get that for any quality value below 1.8 we reject it to be very poor for a software system.

## 5 Results Findings and Discussions

In performing the multivariate analysis in this research two hypothesis were tested that is H1:To=T2=T3 Ho represent the first hypothesis.To,T1,T2 represent the quality factors maintainability, portability, usability, traceability, reliability and under stability. This hypothesis was chosen to test if all factors have equal contribution to the overall software quality. This hypothesis is to be tested against the alternate H2: T1≠T2……Tg where g is the number of the sample population, The sample population represent the quality factors maintainability, Usability, Portability, Traceability, Reliability. The total number of sample population is 7.The hypothesis was chosen to with the assumption that none of the quality factors have direct contribution to the overall quality of the software system. In this research g is 7 representing the number of factors (sample population) to be tested and analyzed in the framework. We reject the hypothesis H1:T1=T2=T3…..Tg if the ratio of the generalized variables is too small given by wicks lambda $\Lambda^{*} = \frac{/W/}{/B+W/}$ , A low value implies that all factors usability, portability, traceability, reliability, maintainability contributes equally to the overall quality of the software system under development. This shows that low values for software quality are generated by the sample population and therefore low quality should not be accepted for the software system and therefore adopt the hypothesis H2:T1≠T2…Tg. This hypothesis was adopted since different factors contributed differently to the overall quality of the software system under study and consideration in this research. In the analysis for overall population sample the wicks lambda value was evaluated with ά=0.01 level of significance and g =7 and g is the number of population in this study, lambda was found to be 0.03056466 and using lambda value we evaluated the threshold value and we got the minimum value to be 2..taking α=0.005 the half value, we generate threshold value of 1.8 to translate that average quality can be between 1.8 to 2.49 .It can therefore be said for any value greater than 2.5 is an indication of high quality, for any value of evaluation between 1.8 to 2.49 indicates average quality and for any value less than 1.8 is an indication of low quality. A high score of 2.5 produced output of high quality since it can be observed that the data items under consideration were strongly linked that they directly contributed to overall software quality.

## 6 Conclusion

This research paper was majorly aimed at improving the Dromey quality framework. The Dromey framework viewed quality in four major perspectives and they are:

I. Portability
II. Reliability
III. Usability
IV. Security

In this research the software quality has been viewed in 8 different perspectives and they are Portability, Reliability, Maintainability, Usability, Efficiency, Understandability, Traceability and completeness. The quality factors that define software systems has been awarded quality index to determine their individual contribution to the overall software quality. This framework would improve assessment of software quality since it would assign weights to each of the attributes that define a factor and weight to each factor and overall weight to the quality index. The Dromey framework doesn't assign weight to the factors and neither does it assign more weights to the individual attributes that define a specific quality factor. This framework would be used also to identify the attributes that define quality of the software system. The framework improves the assessment of the software quality by ensuring that we can

actually identify the individual attributes which would affect the overall software quality. This attributes can then be reworked into the software system and ensure that the software system meets the intended quality threshold as required by both the developer and users of the system.For already existing system in operation the assessment framework can be used to assess the software to determine flows that exists in it so that they can be improved during the maintenances process to meet the actual quality levels.

**References**

http://www.yann_gael.gueheneuc.net/work/Tutoring/Documents/041021+Khosravi+Technic al+Report.doc.pdf, 2004.

Georgiadou E. (2003) "GEQUAMO– A Generic, Multilayered, Customizable, Software Quality Framework", International Journal of Cybernetics, **11**(4) , pp 313-323

Kitchenham B., S. L. Pfleeger(1996). "Software Quality: The Elusive Target", IEEE Software,**13**(1), pp.12-21

Sousa-Poza A., M. Altinkilinc, Cory Searcy, "Implementing a Functional ISO 9001

Gordon Skelton W.(2006), "Integrating total quality management with software engineering education", ACM SIGCSE Bulletin, **25**(2).

Suryn W. ,(2003) "Course notes SYS861", École de Technologie Supérieure, Montréal.

Garvin D. , "What Does 'Product Quality' Really Mean?" Sloan Management Review, Fall, pp 25-45, 2010