International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 18 (2015) pp 38671-38678

Quality Management in Information System Development Units – An Empirical Study

Dr. Murugavel R
Associate Professor,VIT Business School,VIT University, Vellore – 632 014.rmurugavel@vit.ac.in

Abstract
This study attempts to identify the prevailing Quality Management Systems (QMS) in IS industries and proposes means and ways by which it could be analysed, characterised, evaluated and improved to ensure the quality expected in software development activities. The study began with a proposition that the quality can be defined in several ways and the concept is dynamic because of very frequent changes in the factors that determined quality. Therefore, it was argued that the quality must be defined in terms of attributes that were satisfying the expectations of the customers. Though, the concept of quality was difficult to define precisely but there are guidelines to help develop quantitative measure of a quality. This was attempted in this study by preparing two hypothetical models one for the customers and another for the IS units and putting them to empirical test. The result showed that this approach yielded a reliable measure of quality.

The IS units used the attributes identified by the customers for the quality of the products as the goal to be achieved with the help of QMS either singly or in combination. The result showed that the use of QMS was definitely advantageous over non-use of it. When combinations were used they provided further improvement in the quality as compared to their use individually, especially when TQM and SSM were part of these combinations.

Thus, ISO, CMM and CMMI are seen to be in quality management through the attention to structural and process variables. TQM and SSM directly targeted the organizational goals to be realised to the highest level possible. Therefore, the combinations of TQM and SSM with other QMSs would take care of both structural and operational variables to enhance quality of products of Information System Development Units.

Keywords: Quality, Information Development Systems, Cost Effectiveness, Cutting Score, IV & V, Quality Index, Discriminant Function, Function Analysis

Introduction
Software industry is unique in many ways. New technologies, new application areas, new information-processing methods, and new programming environments create uncertainties and costly choices in software development decisions. Software products range from basic software to advanced mission critical software. Today’s software often fails "in unpredictable ways." Yet, it is the driving force for building quality in major sectors of the economy.

The software has proliferated into all sectors of the economy. Thus, its users vary from novice to experts, children to elders and students to scientists. All of them may be using the same product but their expectations in it are very different. And none of them will be in a position to state clearly what they want in it, in advance. More than their explicit requirements, there are implicit requirements that are often the basic cause of success of the product and they are to be understood and met. In this dynamic environment achieving quality in software is very difficult. Further, the market for software has no barrier to entry. Anyone who can develop a software programme can have immediate access to the market through the internet at a very little cost. Even without any hi-fi brand names, the products can be rolled out in the market as a trial version offering choice to the potential customers to try it before purchasing. By this process not only the product is accepted in the market as it is but also provides scope for its continuous development and customization. This makes the product life cycle much shorter than those of conventional products. Staying-in-tune with the customers is the only strategy to cope up with this situation.

Information Systems
Information processing is probably the most significant industry in the present world economy and in the foreseeable future [1]. As more companies go in for computerising their operations, the successful operation of their business depends on the satisfactory functioning of the software and computer systems. Hence, it is no wonder that of the different types of software, Information System (IS) accounts for a majority in terms of number of projects.

Again an information system could be as trivial as a transaction processing system with a billing or a point of sales software or it could be comprehensive data mining software working on terabytes of data. Within these types of information systems, the quality attributes vary enormously. The billing software requires high speed and accuracy in arithmetical calculations such as units and amount, whereas accepting spelling mistakes or any such mistakes in customer details such as addresses are not considered as a lapse on quality. On the other hand the prediction model in a data mining environment requires closer approximation rather than the most accurate estimate. Similarly the high end enterprise wide resource planning (ERP) packages require high stability and maintainability rather than speed and accuracy. The software industry is growing exponentially with this kind of complexities and variations. That is why it continues to be a challenge to the community of software developers to develop products satisfying the changing expectations of their customers.

Quality – What is it?

Definitions of quality are many. Users are unlikely to be aware of them; however they know very well when quality is not available in products and services. The meaning of the term “quality” is like “beauty”; it is in the eyes of the beholder [2]. As Weinberg states, "quality is relative, what is quality to one person may even be lack of quality to another [3]."

Not only the concept of quality is varying extensively, the approaches to meet it also vary significantly. Implementation of ISO standards and Total Quality Management (TQM) are a few examples in this direction. Wheeler and Duggins (1998) see no absolute formula that can be used to improve software quality, but there are many guidelines and approaches that have been provided by the quality experts and industry professionals [4].

Changing Paradigm

The recent developments in the field of quality have resulted in a paradigm shift in computer application. The cost of quality for any product is shifted from ‘liability’ to ‘asset’ as the modern tools and techniques of quality fairly aim at controlling the cost more than costing the product, process or service. The new paradigm in quality emphasizes prevention of defects at their root.

Quality Management

The best quality software is one that performs best its intended function "first time and every time", makes efficient use of the target hardware capabilities, and is easy to modify or enhance. Though this sounds simple, with the complexity of software for today's and tomorrow's applications, it is of a very high order.

As the software becomes more complex, it is increasingly difficult to test all the possible paths it takes. In 1960’s and 1970’s there were rudimentary software development methodologies and also little thought of (a) how to design and develop software that could be modified later, and (b) of using the computer hardware efficiently. When it was found that software was not performing as it was supposed to, a method for analyzing the developed code and to locate faulty code was formulated; thus, the Independent Verification and Validation (IV&V) method was born. IV&V attempted to find and document deficiencies in the software and correct those deficiencies, thus ensuring development of more reliable software.

However, IV&V was very costly and the results or benefits were not appealing. IV&V was also reactive in nature, responding to software that had already been designed and coded. i.e., it treated the symptoms and not the cause. Yet, IV&V plays a crucial role in the life-cycle of software.

The causes of poor quality software are numerous; they are broadly grouped into three categories: 1) software development and design methodologies, 2) the need for ensuring that requirements are well defined and are traceable throughout design and development, and 3) the human element. Of course adequate software documentation is integral to all the three categories. This raised the question: will standardisation help?

Standardisation

The attempts to force standardization and enforceable controls on the software development process have met with varying degrees of success. The managements of the software firms are constantly looking for ways and means to adopt the best practices and institutionalise the same in their organizations. The quality concepts, philosophies or systems that worked fairly well in manufacturing sectors are a source of inspiration for these software units to emulate. ISO Certification, TQM (Total Quality Management), and Six Sigma philosophies are efforts to serve the purpose.

Further progress came in customization of these principles suiting the sector, developed across different countries. BOOTSTRAP, TickIT, CMM (Software Engineering Institute’s Capability Maturity Model), CMMI (Integrated Capability Maturity Model), P-CMM (People-Capability Maturity Model), Trillium, STD (Software Technology Diagnostic), and SPICE (Software Process Improvement and Capability Determination) are some typical examples in this direction. These standards have been developed to evaluate software providers’ capabilities in developing software for government services.

As software development and software design processes are evolving, the quality assurance process also has undergone its own evolution in response to the new methodologies and increasing complexity and criticality of the software.

Attributes of Software Quality

Many researchers have identified the desirable attributes of software quality. Boehm et.al. (1996) identify various dimensions of quality and relate them in a logical sequence [5]. However, the degree of their importance varies considerably among the different types of software. The metric used to measure is also a subject on its own. There is no consensus on the utility as well as the ways and means by which quality can be measured. Thus, where quality cannot be exactly defined and cannot be measured; it can’t be controlled. But, attempts are made to define software quality at least contextually. The developers, users and researchers use the following attributes in one or different nomenclature. Some have these metrics for guidance; but they are not absolute techniques for how to achieve quality.

A brief description is presented in Chart 1.

The basic attributes in column 2 aggregate in different combinations and yield the secondary attributes (of col. 3 & 4) and these secondary attributes in turn aggregate and define the final software quality (col.5)

However, to build software with these quality attributes, the organizations need to adopt different tools and techniques depending on the type of software, because, contribution of these factors individually and collectively to the quality of the software varies depending on the criticality of the software and also the differing expectations of the customers who use it.

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 18 (2015) pp 38671-38678
© Research India Publications. http://www.ripublication.com

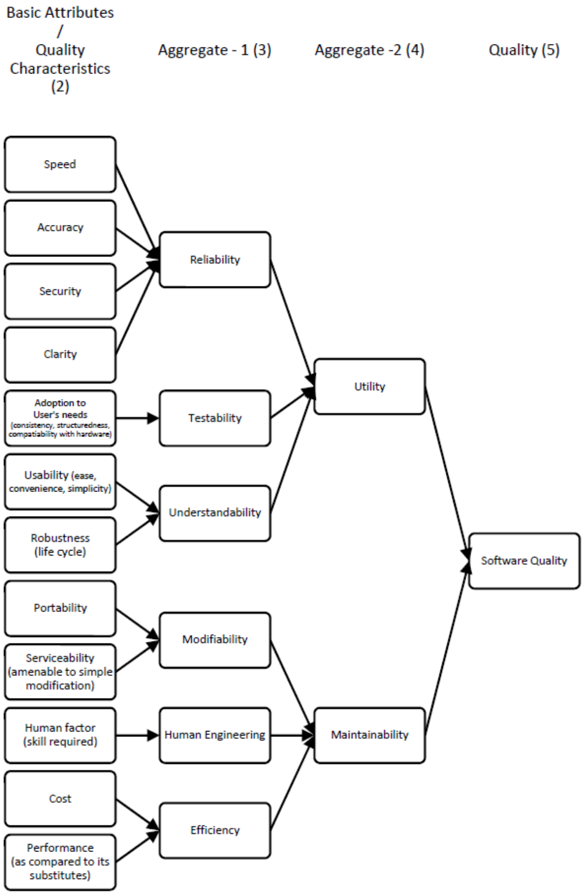

Chart 1. Dimensions of Software Quality

The Big Puzzle

In the attributes discussed above, there are conflicting challenges for the developers to work on it. For example, as security increases, performance decreases, because increased security consistently reduces performance. Thus, simultaneously increasing security and performance is difficult if not outright unrealistic and there is trade-off between these attributes. Similarly, as the fault tolerance increases, testability decreases, because the software can more easily reveal hidden defects.

As Hamid and Madnick (1991) observe, today's large software-intensive systems have critical quality requirements, limited budgets and tight development schedules [6]. To reduce the cost, and delivery time, if tight work schedules are drawn how could the quality be improved? Similarly, with the variation in the skill level and experience of development team, the quality level will also vary. In the case of customer driven development processes the quality of the software could only be as good as the input. Improving the Requirements Engineering (RE) process has been identified as a key issue for the development of effective, good quality systems that meet user expectations; that are delivered on time and developed within the budget. In order to improve the quality of the systems developed, there is a need to improve understanding of the RE process management.

In addition to these stated requirements or revealed requirements, customers have also expected requirements and exciting requirements (Kano's (1984)model) [7]. In the customer requirement acquisition, it is necessary to make sure that complete, consistent, non-ambiguous, non-redundant, and true customer requirements are identified and specified for all the three types of requirements.

Revealed requirements are typically those obtained by asking customers what they want. These requirements are satisfied in proportion to their presence in the product or service. Expected requirements are often so basic that customers may fail to mention them until products fail to perform to this need. They are expectations, without which the product or service may cease to be of value. Their absence is very dissatisfying. Exciting requirements are difficult to discover. They are beyond the customer's expectation. Their absence does not dissatisfy while their presence excites.

Problem Focus

It is well illustrated by various researchers and professionals that quality is a subjective attribute associated with products and services. Its value differs from person to person depending on one's perception. The provider needs to understand the customers in terms of their perceived value. Only when the factors contributing to quality are identified, well defined, analysed and trade-off amongst them are understood, it would be possible to overcome the difficulties in building quality in the products and services of Information System (IS) development units. The organizations do have a system of resources, methods, and materials to build the quality in their products. For example, Richardson (2002) proposes a model based on quality function deployment particularly suitable for small software development companies [8]. Fujimura and Moore (1997) propose a three step model to integrate often conflicting requirements of quality and schedule variance [9]. Dromey (2003) proposes a preventative approach to quality building by contrasting views expressed for curative approach to it [10]. However, the relative contribution to quality is not adequately documented in any of the studies.

It is in this context, the present research attempts to identify the prevailing Quality Management Systems (QMS) in IS industry and proposes ways and means by which it could be analysed, characterised, and improved to ensure quality expected in software development activities. More specifically, this study attempts to map the quality factors and contribution of standards and other resources in achieving expected quality. It proposes a theoretical model and provides for its empirical testing. This model needs to be validated over time.

Data and Sampling

The total population for this study was infinite as the number of users and developers of IS product is ever increasing. Hence, a simple random sampling technique was used to select the sample respondents. About 200 software companies that develop information systems for their customers were randomly selected from the companies listed in the NASSCOM website. Similarly, users or customers of IS products were listed with the help of selected IS units. Thus, 200 IS units and 400 customers were covered for the survey. 118 responses from software professionals and 134 responses from customers were received over a period of three months. Of the 118 responses received 12 were incomplete on many aspects and had to be dropped. For the same reason, 26 responses were dropped out of 134 customer responses. Therefore, the final sample consisted of 106 professionals (IS units) and 108 customers. It worked out to the response rate of 53 per cent of professionals and 27 per cent for customers.

Instrument for Data Collection

Two different questionnaires – one meant for (i) customers or users of IS software and (ii) the other for professionals and engineers engaged in the development of software were developed and used. The instrument was pilot tested with randomly selected professionals and persons in academia to test its validity and its usefulness to define quality. The content validity, and face validity of the instrument were assured in the initial stages of questionnaire development.

Analysis

The collected responses were analysed first separately for the customers and professionals, and then compared for the agreement among quality attributes required in software products. In the case of customers, 12 attributes were identified. For the professionals the criteria for building the customer preferred qualities in the product were identified. They included: human resource, organizational climate, cost of quality, customer focus, and the organizational methods to motivate and appraise the workforce to the desired level of quality. Particular attention was paid to the quality management systems that were in practice in the organizations studied. Specifically the five systems earlier listed were seen to be practiced either individually or in combination. Therefore, the extent of meeting the quality requirements of the customers was evaluated for each of the QMSs and also available combinations of them. This required a unique measure of quality that took into account all the 12 attributes preferred by the customers. For the purpose an aggregate quality index was constructed.

Empirical Measure

When IS units wanted to build quality in their products in response to the expectations of the customers, then management had customer focus as an important factor in their organizational goals. The result of factor analysis showed that customer focus was an important factor in the first component that largely described the structural characteristics of IS units. In the second component which represented the working strategy included priority to quality and cost control as important variable implying that the IS units wanted to build quality to satisfy their customers with care to control the cost of doing so. All other variables were organization specific decision variables to meet this goal of optimizing ‘quality-cost’ relationship.

The concept of customer focussed quality drew their attention to the attributes that defined quality in the perception of the customers. The results of factor analysis identified 12 of the 13 attributes to be important for the customers. However, human resource management and continuity of services were internal decision variables to the user organization. They required only some support for training employees and providing after sales services to the customers by the IS units. Therefore, in building quality in their product, the customer focus of the IS units would require specific attention to ten attributes. They were: (1) performance and (2) cost as influenced by (3) accuracy, (4) security, (5) speed, (6) clarity, (7) robustness, (8) serviceability, (9) portability and (10) usability (adaptable to the needs of the users).

Quality Index

For each item, representing a quality attribute desired by the customers they were asked to grade the attribute on a five point scale having values from 5 to strong agreement to 1 for strong disagreement. Lickert’s scaling technique was used to evaluate the statements. For each item, mean, variance, standard deviation, standard error and coefficient of variation of the scores of the respondents were estimated. The coefficient of variation indicated the importance of the item in defining quality, if it is smaller than 15 per cent. Any value large than 15 per cent indicated lesser precision and therefore also lesser importance of the item in determining aggregate quality index. The mean values were then expressed as percentage to maximum possible value of five and it represented the relative importance of the item (attribute) in the aggregate quality index. Their values were represented by qi. The standard error was ranked in the ascending order and the ranks were used as the weights (w_i). Then aggregate quality index (Q_p) was defined as

$$Q_p = \frac{\sum w_i q_i}{\sum w_i}. \quad (1)$$

It is the perceived measure of quality and might differ from actual empirical measure of quality denoted simply by Q_i for each IS unit i.

The success of IS units might be measured by the extent to which these attributes were met in the products. In order to evaluate how far these requirements were met, one product in each of 106 IS units was selected and the respondent in the unit was asked to say the percentage of quality attribute met in the product. These percentages were shown to the customers and their agreement with them was elicited on a five point Likert’s type of scale, with score values 5 to 1 from strong agreement to strong disagreement. The mean of the score values were then used as weights to find weighted average of percentage values assigned by respondents of IS units. That was an aggregate measure of quality attained by the sample units in their products studied. It was denoted Q_i for i = 1, 2 ... 106 of IS units. This quantitative measure of quality was the objective of this study besides identifying management practices that would further improve it.

This index was the target variable for the IS units to achieve so that their customers were fully satisfied because software producers were assumed to work for customer satisfaction. Each of the units would try to maximize the value of this Q subject to the resource constraints, especially strength and quality of the human resource. However, value of Q is not static and it is changing frequently due to the changes in the preferences of the customers many of whom were corporate bodies and also in technology and business environment.

The actual values of Q realized by the IS units for each system of quality management individually and also for combinations of them were estimated from the data collected from IS units. Therefore, there were 12 variables that determined achievement of the quality (Q) for each of the QMSs put to practice by them. Organizational climate, organizational culture (value system in IS), software development practices (software engineering), quality of the human resource, quality management systems, and customer focus in product quality were the common attributes influencing software quality for both customers and their providers.

The sample (106) for the study showed that the use of QMS was the strategy to enhance the quality of software products to meet the expectations of their customers. Six different QMS were in use and majority of IS units (66) used combination of two or more QMS's, while single QMS was used by 22 units. There were however, 18 units that used no QMS and they served as control units (CU). It was therefore possible to make a comparative study of IS units with and without QMS for their differences in quality management. The discriminant function analysis was used because the purpose was to identify the factors that influenced quality of the product significantly.

Discriminant Function

The discriminant function analysis (DFA) was useful to evaluate the relative importance of different variables that differentiated two groups to the maximum. In the study this analysis was used to differentiate IS units in four groups. The groups were:

Table 1 Groups Formation

Groups	*IS units with*	*No. of Units*
G1	No QMS	18
G2	One QMS	22
G3	Two QMS	33
G4	3 or more QMS	33
Total		**106**

The discriminant function analysis was done pair wise for (1) G1 and G2, (2) G2 and G3, (3) G3 and G4.

In constructing the function all variables which would differentiate the two groups maximally were first identified. Mahalanobi's Minimum D2 method was used to select the variables. Though nearly 14 variables were listed, only five variables were finally selected and they were:

1. Age of the IS units (Years) – AGE
2. Number of employees (No.) – EMP
3. Skill Level (Index) – SLI
4. QMS in Use (Index) – QMSI
5. Total Quality (Index) – QI

The variables 3, 4, 5 would require description. The skill level in the IS units was measured by the aggregate of number of years of education of all the employees beyond higher secondary level, including, graduate and post graduate education, special training attended and total number of years in software job in the present position and also elsewhere. After estimating this for the employees of each IS units, the smallest value was taken as 100 (base) and other values were expressed as a percentage to the base. It was Skill level index (SLI).

The IS units used ISO, CMM, CMMI, TQM, and Six Sigma (SSM) for quality management either singly or in combination. The users of ISO would gain experience and improve the system with necessary correction in the process of its application, especially guided by the quality audits that were mandatory. Therefore the number of years of ISO in use was taken as a measure of it. In CMM and CMMI, the maturity levels attained was important. The years spent in reaching the present status, were taken as a measure. TQM and SSM were used only in combination with other forms of QMS and only in few firms. Therefore, the number of years of their application was taken as a measure. Wherever combinations of QMS were in use, the years for all the QMS used by the IS units were aggregated. Finally, the lowest aggregate value was taken as 100 and an index of QMS application (QMSI) was constructed. When there was no application (control group – CU) a value of 10 was assumed to avoid the problem in using zero value.

Empirical Model

Total quality index was the weighted average of scores for variables that satisfied customer expectations as defined earlier. With the above mentioned variables the empirical model of the Discriminant Function (DF) was the following.

$$Z = a_0 + a_1 AGE + a_2 EMP + a_3 SLI + a_4 QMSI + a_5 QI \quad (2)$$

Where

a_0, a_1, a_2 ... a_5 were parameters (discriminant function coefficients) to be estimated.

Estimation

There are two methods of estimation of discriminant function (i) simultaneous method and (ii) stepwise method. As the independent (discriminating) variables were few, stepwise method had no special advantage and the simultaneous method was used. In this method, all the five independent variables were included.

Classification

To know the efficiency (predictive power) of the estimated discriminant functions, the classification matrix was used. The matrix showed actual and predicted group membership of the IS units. For the preparing classification matrix, the group of centroids (means), cutting score and also *a priori* probabilities of each group were calculated. Using the discriminant function (DF) the discriminant scores for each IS unit was

calculated by substituting the values of the discriminating variables from the data. Then mean scores (group centroid) were calculated for each group.
Using the number of IS units in each group and the centroids Cutting Scores (Z_c) were calculated as:

$$Z_c = \frac{N_0 Z_0 + N_1 Z_1}{N_0 + N_1} \quad (3)$$

Where
Z_0, Z_1 were centroids for groups compared
N_0, N_1 were number of IS units in the groups

If for any IS unit, its score was smaller than Z_c value, then it was assigned to group zero, otherwise to group one.
A priori probability for each group was calculated as the proportion of number of IS units in the group to the total in both groups i.e., N_0 / N_0+N_1 and N_1 / N_0+N_1
Using the probabilities, centroids and cutting score (Z_c) the classification matrix was formed. It showed how many of the IS units were correctly classified into the respective groups and also the overall correct classification percentage.

Table: 2. Estimated Canonical discriminant functions and R_j– Groupwise

Sl. No.	Variables	Notation &Coefficient	G-1 & G-2		G-2 & G-3		G3- G4	
			Estimates[@]	R_j	Estimates	R_j	Estimates	R_j
1.	Z - Constant	β_0	-9.217	-	-10.354	-	-26.328	-
2.	Age of Units	Age β_1	0.034	2.46	0.001	-0.008	0.010	-0.251
3.	No. of Employees	EMP β_2	-0.044*	8.02	0.033*	7.366	0.041*	16.242
4.	Skill Level Index	SLI β_3	-0.014**	-11.29	-0.015*	-6.8381	0.007*	6.321
5.	QMS used Index	QMSI β_4	0.257**	82.48	0.274*	99.702	0.206*	65.963
6.	Quality Index	QI β_5	1.113**	18.33	-0.014*	-0.221	0.088*	11.725
7.	Correct Classification	%	97.5	100.00[†]	100.00	100.00	100.00	100.00
8.	Cutting Score	Z_c	0.0002	-	0.000	-	0.000	-

Note:
Estimates @ - actual discriminating coefficient estimated
R_j – Relative discriminant Index (in %)
Z – Discriminant Index, β_0 – Constant
* - Significant at 5% level;
** - Significant at 1% level
No marking – not significant
% - percentage of cross-validated cases of groups correctly classified
† -total of R_j's (in %)

Discriminating Power
The relative discriminating power of the variables was obtained by creating relative discriminating index R_j by using the estimated values of discriminant function coefficients (a_j) and the group means of discriminant variables.

$$R_j = \left(I_j \Big/ \sum I_j \right) * 100 \quad (4)$$

where $I_j = a_j(\bar{X}_{j0} - \bar{X}_{j1})$ and $\bar{X}_{j0}, \bar{X}_{j1}$ are means for groups 0 and 1. R_j showed the present contribution of each variable to the discriminating power of the discriminant function.

Comparison
Finally a comparative study of R_j for the three groups of IS units viz., G1 and G2; G2 and G3, and G3 and G4 would bring out the contribution of QMS to the quality building in software products.

Economics of Quality Management
One of the dimensions of quality management in IS units is its economics. The IS units want to improve quality of their products in a cost effective way. Their cost of quality management includes: (i) the cost of investment in infrastructure and the human resource of appropriate skill and attitude to work, (ii) the quality management system that reduces errors in operation and ensures motivation and coordination among the employees and (iii) the cost of customer services in terms of installation, training to the users (clients) and after sales service for maintenance and warranty conditions. These costs will only increase with efforts to improve the quality, which increases the returns (prices) to the products / services of IS units. Therefore the cost effectiveness in quality management really means reducing unit cost. i.e. cost per rupee of value realized from sale of the products (marginal and average costs, which is the same as increasing returns per rupee of total cost incurred. The operating profit is usually taken as a measure of financial performance of any business firm, but it fails to take account of fixed costs in infrastructure, technology and human resources. Therefore net income after allowing for the fixed costs is the appropriate measure of cost effectiveness in quality management. The 'return over equity' (ROE) of the IS units is therefore taken as the appropriate variable to measure the cost effectiveness of IS units. It is measured as:

$$ROE\ (\%) = R1 \times R2 \times R3 \times 100 \quad (5)$$

Where,

$$R1 = \frac{Net\ Income}{Sales}$$

$$R2 = \frac{Sales}{Assets}$$

$$R3 = \frac{Assets}{Equity}$$

all variables measured in rupees in current values of the year. The measure ROE was used in this study for two reasons; first direct cost estimates were not available for IS units, second the estimates for the variables defining ROE were available in the published annual reports of IS units. The estimated ROE was taken as a measure of economics of quality management and it was expressed as a percentage to allow direct comparison of IS units and their groups.

Regression Analysis

A simple linear regression equation with ROE as the dependent variable and the quality index as the independent (explanatory) variable was specified by ordinary least square (OLS) method.

$$ROE = \beta_0 + \beta_1 QI + \varepsilon \qquad (6)$$

Where
ROE = as defined above (%)
QI = quality index
ε = random error term
β_0, β_1 = regression constant and coefficient to be estimated.
A priori expectation was $\beta_1 > 0$

Table: 3. Regression of ROE on QI

Groups	Regression Equations	F/R^2	d.f
G-1	$\widehat{ROE} = -10.825^{**} + 0.213\,QI^{**}$ $(-7.204)(10.565)$	111.61 0.91	10
G-2	$\widehat{ROE} = -37.453 + 0.528\,QI$ $(-9.657)(11.778)$	138.72 0.93	9
G-3	$\widehat{ROE} = -72.909^{**} + 0.940\,QI^{**}$ $(-10.127)(11.700)$	136.88 0.913	12
G-4	$\widehat{ROE} = -304.299^{**} + 3.494\,QI^{**}$ $(-9.824)(10.683)$	111.131 0.837	21

Note = figures within ( ) are t statistics ** significant at 1% level.

Findings

- The study postulated that a quality of IS products must be in conformance with the expectations of their customers. The result showed that the customers (corporate end - users) had specific requirements expressed in 13 attributes that clustered into three broad categories viz., vital, essential and desirable attributes. These three aggregate factors would account for nearly 94 percent of variance in the quality required by customers.
- In the case of IS units, organizational climate, strategic planning and human resource were the three factors identified by the factor analysis to be the contributing factors of quality in IS products. Totally they would explain 90 percent of variance of the quality in the product.
- Customer focus emerged an important variable as an item in the factor of organizational climate to explain nearly 10 percent of the variance in the quality of the product showing that concern for the expectations of the customers is a part of quality management system in IS units.
- The quality of the product was estimated with a help of an aggregate quality index – first in a perspective model and then in its empirical verification. The two indices were Q_p and Q_i. Their validity was measured by their ability to explain the quality of the software in a quantitative measure. The result showed that Qp would explain 73.17 percent of the total quality, while Qi would explain 89.76 percent.
- Therefore, use of QMS had definitely contributed to improve the quality of the product, but a combination of more than two QMS would be advantages than just combining two.
- The IS units differed in the use of QMS and therefore they were classified into four groups. The canonical discriminant function analysis (CDF) was done by identifying five variables that would discriminate between pairs of the four groups. The variables were age of the units in years, number of professional employees, their skill level expressed as an index, quality management systems allowing for the level of maturity and number of years of use measured as in index and empirically estimated quality index(Q_i).
- The result showed that AGE of the unit was least influencing the quality. The quality management system index (QMSI) was the most discriminating variable as shown by its relative discriminating strength. It was followed by quality index and index of skill (SLI).

Conclusion

A critical validation of the hypothetical model showed that it fairly represented the quality management system with a specific attention to customer requirements. More specifically it showed strong agreement with the critical factors identified to have influence on the quality of the products. These factors included structural and strategic variables in IS units and vital and essential attributes of customer expectations.
The cost of quality management and the performance of the IS products were important variables for both IS units and their customers. Together, these two variables determined economics of software quality. The return over equity (ROE) was taken as a measure of economic status of IS units. ROE is seen to be significantly increased by QI and combinations of QMS. Thus, quality management in IS units has a high pay-off.

References

[1] Tiwari Ashish and Tandon Amit, Shaping software quality- the quantitative way", Proceedings of First International Conference on Software Testing, Reliability and Quality Assurance, IEEE Software, pp 84-94, 2004.

[2] Stockman Sinclair Guillaume, Todd Alan Richard, and Robinson Gerald Anthony. "A Framework for Software Quality Measurement", IEEE Journal on Selected Areas in Communications, 8(2): 225, 1990.

[3] Weinberg Gerald M., "Quality Software Management: Volume 1 Systems Thinking", Dorset House Publishing, New York, p 13, 1992.

[4] Sharon Wheeler and SheryDuggins, "Improving software quality", Association for Computing Machinery, p 307, 1998.

[5] Boehm, B. W., Brown, J. R., and Lipow, M., "Quantitative Evaluation of Software Quality", Proceedings of 2nd International Conference on Software Engineering, p 595, 1996.

[6] Tarek Abdel-Hamid, Stuart Madnick, "Software Project Dynamics: An Integrated Approach, Prentice Hall PTR, 1991.

[7] Kano, N., Seraku, N., Takahashi, F., "Attractive quality and must-be quality", The Journal of the Japanese Society for Quality Control, 14 (2): 39-48, 1984.

[8] Ita Richardson, "SPI Models: What Characteristics are required for Small Software Development Companies?" Software Quality Journal, 10: 101–114, 2002.

[9] Aki Fujimura and Frances Moore, "Quality on Time", Software Quality Journal, 7: 97–106, 1997.

[10] R. Geoff Dromey, "Software Quality-Prevention versus Cure?" Software Quality Journal, 11: 197–210, 2003.