Determinants of software quality in offshore development – An empirical study of an Indian vendor

G. Kannabiran *, K. Sankaran

National Institute of Technology, Tiruchirappalli 620 015, India

Abstract

Context: Cost advantage has been one of the primary drivers of successful offshoring engagements of Indian software and services companies. However, the emphasis has shifted to the ability of the vendors to provide high quality over cost advantage in delivering software products and services. Meeting high quality requirements of the clients is a challenge due to the very nature of development and delivery of software through offshoring.

Objective: The objective of this research paper is to identify and evaluate the key determinants of quality in the case of software projects delivered through offshoring model.

Method: A detailed survey was conducted among project managers/project leaders (leads) of a leading midsize Indian IT services company to evaluate the relationship of the determinants on the attributes of quality.

Results: Out of six determinants, our research reveals requirements uncertainty has significant association with all the attributes of quality. While process maturity and trained personnel have moderate association, communication and control, knowledge transfer and integration and technical infrastructure have relatively low association on software quality attributes in the case of offshoring.

Conclusion: It is concluded that the complexities in offshoring necessitates proper capturing of requirements. In addition high level of process maturity and availability of trained personnel to the project will help vendors to achieve software quality. The paper provides a set of implications for practice and directions for further research.

1. Introduction

The Indian IT (Information Technology) industry has registered a sustained growth over the past decade and has emerged as a global destination for offshoring. Many Indian software companies have become an acclaimed “vendor” for offshore software development due to their delivery capabilities, characterized by quality and cost-effective software to the customers across the globe. Primarily driven by cost considerations, increasing number of software projects have been outsourced by organizations in the US and Europe to firms in countries such as India and China. In recent times, managers of both outsourcing and vendor firms have realized that cost considerations, generally assumed to be the primary reason for offshore outsourcing, need to be balanced with the focus on high quality. In the past few years, the Indian software industry has pursued the goal of acquiring the highest standards of quality and has thus created a strong value proposition in the software and services arena.

Researchers have indicated that in view of the decreasing cost advantages, Indian offshore vendors need to change focus from ‘cost’ to the “quality” related measures of the software services provided [12,16]. It is rather this shift of focus towards developing the ability to provide higher quality software services and thus moving up the value chain is a strategic necessity for the vendor [21]. Therefore, software quality is a very important aspect in evolving strategy for both offshore vendors and customers. Although, offshore software development had gained prominence over the past two decades, even today the subject of software quality has not been addressed adequately [62]. It is reported that there have been only few comprehensive studies on factors that impact software quality; but quantitative survey-based research is lacking on the subject [82]. Software quality research has focused on the technical and engineering aspects of quality control, while paying limited attention to the organizational dimensions [67]. These researchers also admit that there is inadequacy and insufficiency of empirical studies investigating the management and control of quality of software development. Therefore, this paper empirically evaluates the key factors that determine the quality in offshore software development, through a study of leading IT Services Company in India.
The present research study has taken into consideration the ‘software quality’ as defined by the ISO/IEC 9126 model. The research work was undertaken by operationalizing the software quality as the ‘dependent variable’ which in turn, consists of five key attributes of quality – Functionality, Reliability, Maintainability, Usability and Performance. The determinants are considered as ‘independent variables’ include requirements uncertainty (RU), technical infrastructure (TI), knowledge transfer and integration (KTI), process maturity (PM), trained personnel (TP) and communication and control (CC). The paper is organized as follows. The review of previous research is presented followed by research objectives and methodology. The next section covers the data analysis and interpretation, followed by the discussion on the relative importance of the determinants. The paper is concluded with implications for practice and directions for future research.

2. Previous research

Both practitioners and academicians claim that software quality as one of the critical issues of the decade [45,87,91]. The importance of software quality has been stressed by authors who observed that software quality can determine the success or failure of a software product in today’s competitive market [81,53]. Jalote [38] observed that software quality represents how well the product satisfies the customer. The importance of concentrating on quality efforts during the development process was highlighted and the benefits of process standardization through CMM (Capability Maturity Model) and similar initiatives of higher process maturity levels will lead to higher development quality [75]. The need for quality management in software projects becomes highly relevant in this context and quality of software is receiving a great deal of attention. Quality management is one of the methods for reducing the cost of production, by eliminating inefficiencies in the development processes. In an effort to keep costs within budget and simultaneously to meet the scheduled time of delivery, IT firms are adopting various quality management practices [32,84].

The relevant previous research relating to software quality and its determinants are reviewed and presented in the following section/sub-sections. Further, the literature review for each determinants of software quality is presented in the subsequent sections.

2.1. Software quality (SQ)

Assessing software quality in the early stages of design and development is crucial as it helps reduce effort, time and money [2]. Many definitions of software quality (SQ) are in use, which in general agree on what quality means and it can be enshrined by the phrase ‘satisfaction of customer requirements’ [90]. SQ is a multidimensional measure and it is essential to determine what aspects of quality are important to organizations. There are essentially two approaches that can be followed to ensure software product quality, one being assurance of the process by which the product is developed, and the other being the evaluation of the quality of the end product. The state of the art in software technology does not yet present a well-validated and widely accepted description scheme for assessing the quality of software products. Much work has been done by a number of individuals to define a SQ model which acts as a framework for the evaluation of attributes of SQ [8].

Traditionally, SQ has been defined to be composed of correctness, reliability, usability, and maintainability [20]. Quality models have been developed in the past, the most widely known of which being McCall’s and Boehm’s, and more recently the ISO9126. In 1991, International Organization for Standardization introduced a standard named ISO/IEC 9126 software product evaluation-quality characteristics and guidelines for use [43]. McCall’s model is an early, product-oriented quality model, which shows how external quality factors, such as correctness, reliability, efficiency, etc., relate to product quality criteria, such as completeness, accuracy, error tolerance, etc. These product criteria can be measured and thus external quality factors may be assessed [56]. Boehm et al. [10] proposed the same hierarchical structure as McCall’s model, but also put emphasis on users’ expectations and hardware performance. The complexity of Boehm’s model is equal to that of McCall’s, that is, the quality criteria are related to a variety of quality attributes with relations sharing common attributes. Another model for IS success was proposed by incorporating six independent measures viz., system quality, information quality, user satisfaction, individual impact and organizational impact [18].

Finally, the ISO 9126 model follows the same approach as the other two aforementioned models, with the main difference being that the level of complexity is lower here as the hierarchy is stricter that each quality characteristic is related only to exactly one attribute, with no common attributes between characteristics.

2.1.1. Software quality attributes

As per ISO 9126 standard, quality is defined as “the totality of features and characteristics of a product or service that bears on its ability to satisfy given needs” [11]. Yang [91] identified that quality of the software product could be estimated using its attributes, such as reliability, maintainability, functionality, extensibility, and usability. SQ has also been defined [9] in terms of two types of product characteristics: (i) external quality (how the product works in its environment), like usability and reliability, and (ii) internal quality (how the product was developed), such as, software structure and complexity [27].

External SQ attributes are being emphasized as they have been already validated through a study carried out in the recent past [40]. A recent study by Gorla and Lin [26] on determinants of SQ through a survey of Information system project managers has found that organizational factors are more important than technical factors in impacting SQ. According to ISO 9126 model, there are six major characteristics (also sometimes called quality factors), namely, functionality, reliability, usability, efficiency, maintainability and portability, along with their associated sub-characteristics [4]. However, many researchers have customized these generic quality attributes suit- ing to their context. For example the attribute ‘portability’ is considered applicable only in software products that need to be implemented on multiple platforms [23]. The desirable SQ attributes for web applications are reliability, usability, security, availability, scalability, maintainability, and time to market [8].

Similarly, efficiency attribute is viewed as an internal quality attribute as it deals with time behavior and computing resources consumed [26]. The following quality attributes are adapted from the ISO 9126 model for measurement.

2.1.1.1. Functionality. It is the capability of the software product to provide functions which meet stated and implied needs when it is used under specified conditions. It includes the following sub-characteristics: suitability, accuracy, interoperability, compliance and security. Functionality expresses the ability of a system to provide the required services and functions, when used under specified conditions [26,57,88].

2.1.1.2. Reliability. A set of attributes that bear on the capability of a software product to maintain its level of performance under stated conditions for a stated period of time. It includes the following sub-characteristics: maturity, fault tolerance and recoverability. Reliability is an indication of the confidence that the software will live up to the expectations [55,13].
2.1.3. Maintainability. A set of attributes that bear on the effort needed to make specified changes. It includes the following sub-characteristics: analyzability, changeability, stability and testability. Maintainability relates to the means provided by the software to be upgraded and customized. In this context, software process improvement (SPI) helps to enhance maintainability of the software [8,41].

2.1.4. Usability. It is the capability of a software product to be understood, learned, used and liked by the user, when used under specified conditions. It includes the following sub-characteristics: understandability, learnability and operability. Usability indicates the understandability of software as well as the easiness to learn and operate it [3,31,44].

2.1.5. Performance. A set of attributes that bear on the relationship between the level of performance of the software product and the amount of resources used under stated conditions. It includes the following sub-characteristics: time behavior and resource behavior. In our research, the attribute efficiency is substituted by ‘performance’ of the software in our chosen list of software quality attributes [4,48].

2.2. Requirements uncertainty [RU]

According to Parnas [61], requirements uncertainty (RU) means that within a software system, requirements are not known until it is practically used. Requirements uncertainty emerges from inability to apply a standardized process to convert requirements into functional specifications [66]. Previous studies [7,59,32,85,30] posit RU to be an important source of poor quality in software development. This uncertainty results from software requirements creating an ill-defined problem. According to Pressman [66], as requirements keep changing in the project life cycle, subsequent changes in the design and rework needs create disruptions in managing requirements and provide scope for defects. Unstable requirements lead to sub-optimal design during the initial phases of the project [66].

Ebert and De Man [14] observed that no single technique will be applicable but a combination of techniques need to be applied for managing the RU in software projects. Further, when there is user diversity in requirements, the vendor needs to address and reconcile the differences [59]. Apart from not meeting the quality requirements, RU leads to projects delays and cost overrun. While most studies have hypothesized the effects of RU in in-house projects, its effects are magnified in the offshore domain due to barriers of geographical and organizational boundaries between the client and vendor [25]. Therefore, it may be considered that RU will affect adversely the quality of software in projects. The variable RU was measured based on the construct used by Nidumolu [59] with the additional items arrived through discussion with industry experts.

2.3. Technical infrastructure (TI)

TI refers to interconnecting hardware and software and other devices that are interconnected to support the flow and processing of information [6]. It becomes very vital in the case of software industries, where the technological advancement is experienced at a very rapid pace and the adaptation of technological advancement is mandatory for the very survival of software organizations. SQ depends on availability of good tools, good materials, good methods and management techniques, and latest technological developments [52]. Jones [42] identified that the improvement of infrastructure (support facilities) was one of the essential elements of successful development. Many researchers have found obsolescence of technologies as the major risk faced by software companies [7,38,71].

 Exposure to latest technology and availability of facilities, including latest hardware and software, play a crucial role with respect to SQ. Several authors have expressed the need for telecommunication, software tools as part of the infrastructural facilities available for software development [15,17,63,77,39]. Rao [70] also claimed that the facilities and work environment would add to the motivational value of the employees. The purpose of the work environment is to establish and maintain physical working conditions that allow individuals to perform their tasks efficiently and to concentrate on their tasks without unnecessary distractions. Bahrani and Evans [6] argued that providing the facilities and stable physical environment could improve the development environment. The variable TI was measured using the construct developed by Rajendran et al. [68].

2.4. Trained personnel (TP)

Availability of TP is a key factor of software quality. Individual capability was found to be the most significant determinant of performance among the software developers. Curtis et al. [37] advocated that trained individuals are those who have superior application knowledge, communication skills, high levels of motivation, team spirit and dependability are ‘essential’ for the success of a software project. Prior research has highlighted the importance of personnel competence in developing high-quality software. This effect should be particularly true in the offshore development since there is always a supply–demand mismatch with respect to trained professionals [78]. Pressman [66] observed that trained programmers use better design techniques, are more aware of the link between requirements and systems parameters and are able to write better code. Experienced personnel are also more capable in testing and defect prevention activities, thereby resulting in better quality and more productive use of resources [49].

In offshoring projects, the availability/non-availability of trained personnel is further affected by the continuity in the staffing of projects. Frequent staffing changes affect quality because different personnel, at different points in time, make changes to the project that may have little relation to earlier choices and specifications. More particularly, it leads to poor design and development of the project, leading to poor quality. Thus, the lack of trained personnel impacts quality of offshore projects in two ways: first, by the staffing of inexperienced and untrained personnel and second, by frequently shifting personnel between projects as managers seek to manage multiple projects with insufficient numbers of trained personnel [25]. The variable TP is measured by the extent to which trained staff were deployed in the project and adopted from Lacity & Hirshheim [64].

2.5. Knowledge transfer and integration (KTI)

Knowledge transfer and integration is defined as the process of absorbing knowledge from external sources and blending it with the technical and business skills, know-how, and expertise that reside in the business and IS units of a firm [29,60]. The role and importance of knowledge in software development is widely recognized by previous researchers [19,86]. Rus and Lindvall [76] observed two types of knowledge central to the software development process-technical knowledge that is used to develop a system and knowledge about the business application domain. Technical knowledge refers to knowledge about design (e.g. design patterns, heuristics best practices, technical constraints, and estimation models), programming (e.g. programming languages and development tools), and software processes (e.g. methodology,
code testing and debugging procedures). Business application domain knowledge refers to knowledge about the customer’s business processes, business rules, activities, stakeholder needs, and the customer’s business objectives for the software.

Internal knowledge integration refers to the extent to which the development team builds on the knowledge of the stakeholders during the development process. External integration refers to the knowledge relating to market needs, regulatory constraints, external environment and business and technical developments that may affect development project [65]. Adopting the development project as the focal point, both external and internal integration are considered to be important in achieving quality [54,83]. This variable was measured based on the construct used by Tiwana et al. [92] with the additional items derived through discussions with industry experts.

2.6. Communication and control (CC)

Communication and control (CC) is broadly defined as the proactive formal and informal sharing or exchange of meaningful and timely information between firms [5]. This definition focuses on the efficacy of information exchange, rather than quantity. Inherent in this definition is the notion that communication is a two-way process; the information flow is not unidirectional, and should go beyond day-to-day exchanges of routine operational information to include open exchanges of desires, needs, and resources that will affect the future of the relationship [33,47]. A control mode is a formal arrangement practiced by both vendors and customers in software projects facilitating the interaction among the teams. Communication and control mechanisms in offshore development reduce project uncertainty and thereby improve quality [24].

Previous researchers [46,58] have suggested the importance of communication in inter-organizational relationship. Communication helps to provide better control of processes, which in turn helps to improve quality [11]. A study of large-scale software development in a major U.S. firm showed the importance of communication in the development process [50]. Similarly, Grover et al. [28] found that communication is a main determinant for the success of offshoring. Communication was also found to correlate positively with offshoring partnership quality [51]. The variable CC was measured based on the construct used by Lee and Kim [51] with the additional items derived through discussions with industry experts.

2.7. Process maturity (PM)

Process maturity (PM) is defined as the indication of how close a developing process is to being complete, and capable of continuous improvement through quantitative measure and feedback [79]. Software development process is viewed as vital for getting new products of good quality, quickly into the market [74]. Process improvement enables the same amount of software to be built in less time, with less effort and fewer defects [38]. The efficiency of the software development process is an important issue facing the researchers and practitioners [89]. Hence, to improve the product quality, the processes need to be improved continuously [11,35,52]. Huyink [36] observed that the documentation of the processes would help to provide a better control over the processes, which would lead to quality improvement.

Harter et al. [32] observed that quality improvement; cycle time and effort reduction can be simultaneously achieved by fewer defects and rework in software development. Maturity implies a potential for growth in capability and indicates both the richness and consistency with which software processes are applied across the organization [59]. Many researchers have endorsed the view that process improvements are vital for improving software quality [11,38,52]. Hevner [34] observed that development of high-quality software is an essential business activity and improving the quality requires the effective use of a software development process. Higher level of process maturity, invariably, results in the enhancement of SQ [80]. The variable PM was measured based on the construct used by Ravichandran and Rai [67] by including few more items derived through discussions with industry experts.

3. Research objectives and methodology

Software quality research has focused on the technical and engineering aspects of quality control, while paying limited attention to the organizational dimensions [67]. However, current challenges facing software development and quality improvement are largely organizational and not technical in nature [68,25]. It is also found that theories and principles were being drawn from other areas; the empirical research is still in its early stage. Based on the detailed review of the previous research on the factors impacting the SQ attributes, in the context of offshore software development, the objectives of the research are to identify and evaluate the impact of determinants on the specific SQ attributes in the offshoring context. The conceptual research model covering the dependent variable, viz., attributes of software quality (SQ) and the independent variables, viz., the determinants listed above, is presented in Fig. 1. The factors are observed to influence the software development process, and thereby, the SQ.

The study was carried out in one of the leading midsize IT Services organizations in India, referred as Helix. The organization is a provider of IT and BPO (Business Process Outsourcing) services and consulting. Founded in 1990, the organization today has seven state-of-the-art development centers - four in India and one each in Germany, USA and Mexico, and offices in North America, Europe and Asia Pacific, and employs around 5200 workers globally. With 159 active clients, Helix has achieved leadership position in industries such as Healthcare and Life sciences, Manufacturing, Travel, Transportation, Hospitality and Logistics, Banking, Finance, Insurance, Leasing and in Domains such as HR and Business Analytics. Helix has achieved the rare distinction of having SEI CMMI Level 5, ISO 9001, BS 7799 and SAS 70 Type 1 certifications. We choose and used a two-stage methodology to meet the objectives of the research.

In the first stage, given the paucity of empirical knowledge on the SQ in the context of offshoring, and the exploratory nature of the study, we carried out a qualitative case-study to validate the variables [93]. Open and semi-structured interviews were conducted with few senior executives, heads of Quality and Practice. These interviews, conducted through few site-visits, covered all aspects of SQ in the organization, with specific reference to QM planning and implementation for projects. The case study findings are used to validate the six important drivers of quality in the context of offshore software development. Some of the items were used by previous researchers and were also found to be relevant to our research [25,67]. The objectives of the research and the conceptual research model are translated into six sets of hypotheses as detailed below:

Set-1: Hypotheses related to requirements uncertainty:

\( H_{1.1} \): Higher the RU lower will be the functionality.
\( H_{1.2} \): Higher the RU lower will be the reliability.
\( H_{1.3} \): Higher the RU lower will be the maintainability.
\( H_{1.4} \): Higher the RU lower will be the usability.

were identified as respondents of the survey. However, most of the responses were collected through e-mails. Out of about 500 projects that were executed during the year, 100 projects were randomly chosen. However, responses from 70 PMs/PLs were received. The data was collected during May–June 2010. The summarized list of questionnaire items is furnished in Appendix A.

4. Data analysis and interpretation

Using the data collected through the survey, a reliability analysis of the measure of each variable was performed using Cronbach’s alpha. Thereafter, a Confirmatory Factor Analysis was carried out for all constructs for checking unidimensionality and convergent validity of the constructs. The results of the analysis relating to validity and reliability are presented in Table 1. The reliability was tested using and Cronbach’s alpha. It is found that value of Cronbach’s alpha for all the variables is above 0.75, thus verifying the reliability of data collected for the study. Confirmatory Factor Analysis (CFA) ensured unidimensionality and convergent validity. The results shown in Table 1 indicate that the Comparative Fit Index (CFI) > 0.90 for all variables, supporting unidimensionality. Similarly, convergent validity is checked by the (NFI = Normed Fit Index (Bentler-Bonnet Index) > 0.90) for all the variables of the study.

Spearman and Pearson’s 2-tailed correlation analysis was to check the inter-relationships between the constructs. The bi-variate correlation coefficient factors are presented in Table 2. It can be observed from the results of the bi-variate correlation analysis that most of the correlation coefficients, except few are relatively

<table>
<thead>
<tr>
<th>Factor/construct No. of Items</th>
<th>Cronbach alpha</th>
<th>CFI*</th>
<th>NFI**</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functionality</td>
<td>4</td>
<td>0.84</td>
<td>0.99</td>
</tr>
<tr>
<td>Reliability</td>
<td>4</td>
<td>0.80</td>
<td>0.99</td>
</tr>
<tr>
<td>Maintainability</td>
<td>4</td>
<td>0.76</td>
<td>0.93</td>
</tr>
<tr>
<td>Usability</td>
<td>4</td>
<td>0.75</td>
<td>0.92</td>
</tr>
<tr>
<td>Performance</td>
<td>4</td>
<td>0.82</td>
<td>0.99</td>
</tr>
<tr>
<td>Requirements uncertainty</td>
<td>7</td>
<td>0.89</td>
<td>0.96</td>
</tr>
<tr>
<td>Technical infrastructure</td>
<td>4</td>
<td>0.74</td>
<td>0.99</td>
</tr>
<tr>
<td>Knowledge transfer and integration</td>
<td>7</td>
<td>0.75</td>
<td>0.94</td>
</tr>
<tr>
<td>Process maturity</td>
<td>4</td>
<td>0.84</td>
<td>0.93</td>
</tr>
<tr>
<td>Trained personnel</td>
<td>4</td>
<td>0.88</td>
<td>0.96</td>
</tr>
<tr>
<td>Communication and control</td>
<td>5</td>
<td>0.73</td>
<td>0.98</td>
</tr>
</tbody>
</table>

* CFI = Comparative Fit Index > 0.90 indicates unidimensionality.
** NFI = Normed Fit Index (Bentler-Bonnet Index) > 0.90 indicates convergent validity.
15 hypotheses were supported. Moreover, it is found that out of 30 hypotheses regressed on each dependent variable, the quality attributes using the technical infrastructure, knowledge transfer and integration, process maturity, trained personnel and communication and control are significantly associated with the reliability attribute of the SQ. Similarly, we observed that hypotheses \( H_{4.1} \), \( H_{4.2} \) and \( H_{4.5} \) are supported and therefore trained manpower has significant relationship with the functionality, reliability and performance of the SQ. We have significant support for the hypothesis \( H_{5.1} \) which confirms that knowledge transfer and integration is related to functionality attribute of SQ. Our results show the support for hypotheses \( H_{5.2} \) and \( H_{5.4} \) which lead us to the finding that communication and control is associated with reliability and usability. Finally, we observed significant support for hypotheses \( H_{6.1} \), \( H_{6.2} \) and \( H_{6.3} \) showing that process maturity significantly affected the quality attributes functionality, reliability and maintainability of the SQ. In the offshore development, although RU affects all quality attributes, it can be inferred from the \( \beta \), \( t \) and \( p \) values in Table 3 that it is highly related with maintainability and moderate on the usability, performance and functionality while its association on reliability is relatively low. In the same manner, we infer that the process maturity has a higher level of association with functionality and reliability while its association is relatively low on maintainability.

### 5. Results and discussion

We found hypotheses \( H_{1.1} \), \( H_{1.2} \), \( H_{1.3} \), \( H_{1.4} \) and \( H_{1.5} \) are supported and therefore it is found that requirements uncertainty (RU) is associated with all attributes, viz., functionality, reliability, maintainability, usability and performance of SQ. We found support for hypothesis \( H_{2.2} \), showing that the technical infrastructure and communication and control are significantly associated with the reliability attribute of the SQ.

### Table 2

Bi-variate correlation among the constructs of software quality and determinants.

<table>
<thead>
<tr>
<th>RU</th>
<th>TI</th>
<th>KTI</th>
<th>PM</th>
<th>TP</th>
<th>CC</th>
<th>Funct</th>
<th>Relia</th>
<th>Maint</th>
<th>Usabi</th>
<th>Perfo</th>
</tr>
</thead>
<tbody>
<tr>
<td>RU</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TI</td>
<td>0.14</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KTI</td>
<td>0.01</td>
<td>0.08</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PM</td>
<td>0.20</td>
<td>0.283</td>
<td>0.484</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TP</td>
<td>0.270</td>
<td>-0.01</td>
<td>-0.13</td>
<td>0.21</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CC</td>
<td>0.282</td>
<td>0.286</td>
<td>0.418</td>
<td>0.599</td>
<td>0.03</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Funct</td>
<td>0.23</td>
<td>0.19</td>
<td>0.08</td>
<td>0.315</td>
<td>0.04</td>
<td>0.20</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relia</td>
<td>-0.09</td>
<td>0.18</td>
<td>0.05</td>
<td>0.13</td>
<td>-0.20</td>
<td>0.22</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maint</td>
<td>0.306</td>
<td>0.18</td>
<td>-0.02</td>
<td>0.18</td>
<td>0.05</td>
<td>0.02</td>
<td>0.579</td>
<td>0.21</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>Usabi</td>
<td>0.249</td>
<td>0.07</td>
<td>0.04</td>
<td>0.13</td>
<td>0.05</td>
<td>0.06</td>
<td>0.549</td>
<td>0.18</td>
<td>0.600</td>
<td>1.00</td>
</tr>
<tr>
<td>Perfo</td>
<td>0.298</td>
<td>0.19</td>
<td>0.04</td>
<td>0.15</td>
<td>0.12</td>
<td>0.08</td>
<td>0.451</td>
<td>0.16</td>
<td>0.456</td>
<td>0.397</td>
</tr>
</tbody>
</table>

* Significance at a level of \( p < 0.05 \).
** Significance at a level of \( p < 0.01 \).

### Table 3

Summary of multiple regression results.

<table>
<thead>
<tr>
<th>Independent Variable</th>
<th>Beta</th>
<th>( t )</th>
<th>Sig[p]</th>
<th>Dependent Variable</th>
<th>Functionality</th>
<th>Reliability</th>
<th>Maintainability</th>
<th>Usability</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements uncertainty</td>
<td>-0.26</td>
<td>-2.27</td>
<td>0.03</td>
<td>RU</td>
<td>-0.20</td>
<td>-2.78</td>
<td>0.04</td>
<td>-0.31</td>
<td>-2.39</td>
</tr>
<tr>
<td>Technical infrastructure</td>
<td>0.09</td>
<td>0.76</td>
<td>0.45</td>
<td>KTI</td>
<td>0.26</td>
<td>2.11</td>
<td>0.04</td>
<td>0.14</td>
<td>1.10</td>
</tr>
<tr>
<td>Knowledge transfer and integration</td>
<td>0.29</td>
<td>2.44</td>
<td>0.04</td>
<td>PM</td>
<td>0.14</td>
<td>1.06</td>
<td>0.29</td>
<td>-0.05</td>
<td>-0.38</td>
</tr>
<tr>
<td>Process maturity</td>
<td>0.33</td>
<td>2.02</td>
<td>0.05</td>
<td>TP</td>
<td>0.32</td>
<td>2.99</td>
<td>0.03</td>
<td>0.25</td>
<td>2.54</td>
</tr>
<tr>
<td>Trained personnel</td>
<td>0.26</td>
<td>2.46</td>
<td>0.01</td>
<td>CC</td>
<td>0.28</td>
<td>2.57</td>
<td>0.02</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>Communication and control</td>
<td>-0.04</td>
<td>-0.27</td>
<td>0.79</td>
<td>RU</td>
<td>0.40</td>
<td>2.61</td>
<td>0.01</td>
<td>-0.23</td>
<td>-1.51</td>
</tr>
</tbody>
</table>

* Significance at a level of \( p < 0.05 \).
** Significance at a level of \( p < 0.01 \).

### Table 4

Summary of hypotheses supported versus rejected.

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Variable</th>
<th>Functionality</th>
<th>Reliability</th>
<th>Maintainability</th>
<th>Usability</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Requirements uncertainty</td>
<td>( H_{1.1} )</td>
<td>( H_{1.2} )</td>
<td>( H_{1.3} )</td>
<td>( H_{1.4} )</td>
<td>( H_{1.5} )</td>
</tr>
<tr>
<td>2</td>
<td>Technical infrastructure</td>
<td>( H_{2.1} )</td>
<td>( H_{2.2} )</td>
<td>( H_{2.3} )</td>
<td>( H_{2.4} )</td>
<td>( H_{2.5} )</td>
</tr>
<tr>
<td>3</td>
<td>Trained personnel</td>
<td>( H_{3.1} )</td>
<td>( H_{3.2} )</td>
<td>( H_{3.3} )</td>
<td>( H_{3.4} )</td>
<td>( H_{3.5} )</td>
</tr>
<tr>
<td>4</td>
<td>Knowledge transfer and integration</td>
<td>( H_{4.1} )</td>
<td>( H_{4.2} )</td>
<td>( H_{4.3} )</td>
<td>( H_{4.4} )</td>
<td>( H_{4.5} )</td>
</tr>
<tr>
<td>5</td>
<td>Communication and control</td>
<td>( H_{5.1} )</td>
<td>( H_{5.2} )</td>
<td>( H_{5.3} )</td>
<td>( H_{5.4} )</td>
<td>( H_{5.5} )</td>
</tr>
<tr>
<td>6</td>
<td>Process maturity</td>
<td>( H_{6.1} )</td>
<td>( H_{6.2} )</td>
<td>( H_{6.3} )</td>
<td>( H_{6.4} )</td>
<td>( H_{6.5} )</td>
</tr>
</tbody>
</table>

\( ^{a} \) Supported hypotheses.
Requirements uncertainty has been identified as a key factor which poses lot of challenges in the project execution, particularly in the case of software projects [7,59,32]. As requirements keep changing throughout the life cycle of the project, subsequent changes to the design and the need for rework creates disruptions in the organization's processes for managing requirements resulting in significant scope for defects [66]. Our research further shows that RU relates to all the attributes of quality in the case of offshoring. It was due to the fact that requirements are captured by an on-site team, which are typically in the vendor organization. Understanding of business context, processes and end-user expectations is likely to get affected by the very approach of gathering requirements and therefore relates to all the quality attributes. Requirements may keep changing and offshoring models limit the possibility of further validations. Moreover, high association of RU on maintainability may be attributed to the limitation of offshoring to understand the future requirements.

Several authors have expressed the need for telecommunication, software tools as part of the technical infrastructure available for software development [15,17,52,63,77,39]. It is found the technical infrastructure is significantly associated to the reliability attribute of the SQ in offshoring. This is due to the fact that most of the projects that are offshored are developed through extensive use of automation tools and reusable codes. Such approaches to software development would have direct impact on the reliability of the software. The availability and cost of connectivity may affect the scope for detailed discussions needed for capturing finer reliability requirements. However, it was shown that the infrastructure does not relate to functionality, maintainability, usability and the performance of the software in offshored projects.

Previous researchers have postulated that the availability of trained manpower is a major challenge experienced by all vendors of Indian IT Industry. Our research outcomes also supported this, as one of the most perennial problems faced by offshore vendors is the increasing difficulty in attracting and retaining trained personnel [25,37,64]. Experienced personnel are more capable in testing and defect prevention activities, thereby leading to better quality and more productive use of resources [49]. To combat the above, the research reveals that the organization has implemented need-based training in order to meet its manpower requirements for various projects. Hence, the presence of trained personnel positively enables the quality software development by the vendor organization as in case of software industry. Our research shows that 'trained personnel' is associated with functionality, reliability and performance of the software. In respect of maintainability and usability which are not associated with the determinant, it is felt that the users will have a greater role to play as these attributes address the post development needs of the software.

Process maturity is essential for executing the software projects, there by promoting SQ [11,38,52]. The organization was treating the quality processes with high priority as vendors need to adapt to different software development methodologies in line with the requirements of clients. When there are multiple customers from many international ventures, it is important that the software methodologies are flexible and adaptable to multiple customers [34,69]. Our research reinforces the previous research findings with regard to the role of process maturity in ensuring the SQ. Further, it is also shown that process maturity significantly associated with functionality, reliability and maintainability attributes of SQ. When the software is developed in offshoring mode with less interaction among the members of the development teams and the end-users, matured processes ensure achieving key attributes of quality.

Our research reveals that the organizations need to establish effective communication and control mechanism among the developers and the customers and facilitated development with a view to resolve issues, reduce project uncertainty and improve performance [5,24]. The study reveals the importance assigned to the communication processes among various teams functioning at both on-site and offshore locations as observed and by many researchers [46,58,11] in the software development projects, particularly, in the context of offshore projects. Thus, the research subscribes to the view that communication and control was a key organizational factor, found to have a significant relationship with SQ in the offshoring context. Further, it is found that communication and control is related to reliability and usability attributes of the quality. Better communication with end-users, especially in offshoring, would help to improve the usability of the software. Time zone differences and organizational culture including decision making and reporting styles affect communication.

Our research also shows that knowledge transfer and integration (KTI) as one of the main inputs for quality software development [22,73,86]. The research reveals that software development can be viewed as a process of integrating technical and business domain knowledge in developing a solution to a business problem [72,76]. Various project stakeholders including members of the client and vendors bring a variety of skills, knowledge, and viewpoints to the software development process. The research also suggests that the organization has successfully implemented processes to achieve integration of knowledge transfer, by blending both business and technical knowledge acquired by the teams and applied for software development [92]. Further, knowledge transfer and integration shows significant association with the functionality attribute of SQ in offshoring. The very approach of offshoring provides limited window of time for the teams to understand the technical and business complexities and therefore affects the functionality of the software.

6. Implications for practice

India has the largest number of companies in the world that are certified for quality standards According to NASSCOM (2006), over 440 Indian companies had acquired quality certifications with 90 companies certified at SEI CMM Level 5 – higher than any other country in the world. This demonstrates the maturity levels achieved in quality by the Indian software industry. This study has attempted to identify and evaluate the key determinants which influence the SQ during the software development by the vendor. In the light of increasing pressures on managers to improve software quality and the growing importance of quality management systems in vendor organizations this study is both timely and significant. Our research has many implications for practitioners in this field.

Firstly, all the determinants have significant relationship with SQ in the case of offshoring. It is found that requirements uncertainty (RU) is related to all attributes, viz., functionality, reliability, maintainability, usability and performance of SQ. It is evident that RU is the foremost factor that is associated with all the attributes of SQ. Practitioners have to ensure very low requirements uncertainty. Use of Agile methodologies and CASE for capturing requirements would help the managers to achieve high level of certainty in requirements. The association of RU with maintainability is high and moderate on the usability, performance and functionality. The very nature of offshoring has limited scope for repeated validations of the requirements, and therefore it affects all the attributes of quality. Further, practitioners have to concentrate on RU if the focus is on the long-term relevance and usefulness of the software.

It is found the technical infrastructure significantly associated with the reliability attribute of the SQ in the case of offshoring. It is also shown that the infrastructure does not relate to
functionality, maintainability, usability and the performance of the software in the offshoring as there are other determinants which may have better relationship with them. Therefore, practitioners may need to focus on TI if the primary requirements are to meet the reliability of the software. This can be achieved by using appropriate development platforms and tools. This is required because most of the projects that are offshored are developed through extensive use of tools for process automation and reusable codes. Such approaches to software development would have direct impact on the reliability of the software. Presence of trained personnel positively enables the software quality in the case of offshoring. It is also shown that ‘trained personnel’ is related to functionality, reliability and performance of the software. Practitioners need to ensure availability of well trained manpower throughout the development of software.

Our research reinforces the previous research findings with regard to the importance of process maturity in ensuring the SQ with respect to offshoring. Further, it is also shown that process maturity significantly associated with functionality, reliability and maintainability attributes of SQ. Indian vendors with high levels of process maturity are successful in the offshoring markets. However, continued success will depend on further fine-tuning of process to achieve quality expectations in the competitive environment. Similarly, it is found that communication and control was a key organizational factor, found to have a significant positive association with the reliability and usability attributes of the quality in the offshoring context. Better communication with end-users, especially in offshoring, would help to improve the usability of the software. It would also help to capture the requirements properly and there by ensure reliable software for the customers. Practitioners need to continuously improve the processes to accommodate the changing requirements in the case of offshoring.

The research also suggests that knowledge transfer and integration significantly associated with the functionality attribute of SQ in offshoring. Efforts to acquire necessary knowledge and integrated with internally developed knowledge would lead to development of software that meets the requirements of the customer. It will also help offshore vendors to dynamically acquire both business and technical knowledge from the customers in order to create new business opportunities. While there are many challenges in offshoring owing to the geographical distances, vendors need to identify and manage the determinants to achieve specific attributes of quality.

7. Conclusion

Our study has identified the key factors which influence the SQ when the software is developed through offshoring. Considering the fact that there are only very few empirical research initiatives in the past on SQ, linking the offshore development from vendor’s perspective, our research has explored and identified the key factors which need to be addressed for ensuring high quality software in the case of offshoring. Our research shows that requirements uncertainty has significantly related to all attributes of quality. While process maturity and trained personnel have moderate association, communication and control, knowledge transfer and integration and technical infrastructure have relatively low association on software quality attributes. Our research has been based on a survey carried out in a single vendor’s site. Nevertheless, further quantitative research may be undertaken to validate the variables based on survey of more organizations from IT Industry. Further, it may be interesting to extend and carry out future studies from different stakeholders’ perspective, especially the quality personnel in the vendor organizations.

Appendix A. Summary of questionnaire items

A.1. Software quality

Functionality:
The software developed was fulfilling the specifications of the user.
Response time of the software was meeting the user expectations.
The software developed provided accurate outputs for users
The software developed with interoperability features to interact with other systems.

Reliability:
The software was stable and unlikely to fail easily.
The software coped up well with environmental failures.
The software was able to recover data and restore optimal functioning during failures.
The software exhibited reliable results to the user under different conditions.

Maintainability:
The software was developed with features to accommodate changes suggested by the users.
The software was able to adapt easily to new specifications or operating environments.
The software required minimum efforts to make system changes.
The software was supported with adequate documentation.

Usability:
The software was easily understood by the Users and its usability was assured.
Users appreciated the software as it was understood well by human–computer interface Software was easily learnt by different Users to make use of the software The software enabled its ease of operation by the users.

Performance:
The software was easily installed by users.
The software incorporated adequate security features for its use.
The software had the ability to detect and trouble-shoot in case any of its component fails Response time of the software was rated to be good, meeting User expectations.

Requirements certainty:
Proper tools (like Rational Rose) were applied to convert the client needs to requirements specifications.
Most of the needs of the client was known and crystallized into requirements at the early stage itself.
Established processes were applied to develop software as per the client requirements Most of the requirements of the client happened in “UAT” stage only.
Most of the requirements of the client happened in “DESIGN” stage.
Most of the requirements of the client happened in “BUILD” stage.
Specific tools and Practices were in place to ensure requirements traceability.
Prototypes were created to understand the requirements better.
Requirements were signed off by relevant stakeholders on time.

Technical infrastructure:
Employees were provided access to production data and other resources at work Licenses relating to different technology were provided.
Access to development tools such as CASE was provided.
A corporate library with technical and domain specific books and other resources.

Knowledge transfer and integration:
Existing application/system knowledge impacted the project.
Technical knowledge transferred impacted the project.
Country specific needs impacted the project. Business domain knowledge transferred from the clients impacted the project.

Organization practice such as COE was adequate for the project. Many creative ideas emerged through the unique perspectives of members of project team at both offshore and on-site, leading to better knowledge transfer and integration.

Proper coordination structure with leadership helped to integrate the knowledge between offshore and on-site resources. Knowledge transfer documentation prepared, reviewed and approved for use.

**Process maturity:**

Software development/project management processes were defined to achieve product quality. Detailed metrics were used for the software development. Quantitative feedback enabled the software development process level of maturity. Higher levels of process maturity were set than expected by customers.

Processes for creating unique knowledge were set to make it available when required. Practices of other methodologies like Agile, Scrum were followed.

**Trained personnel:**

Trained personnel were available to execute the project. The required manpower resources for the project were deployed after cross training. Additional resources were recruited for taking up the project. The manpower resources deployed were not changed till the completion of the project.

**Communication and control:**

Software Tools were used for communication between the teams. Proper training of the team resources was undertaken for better communication and control. Specific governance model was implemented for better communication and control. Effective communication and control was achieved among the team members through regular reviews and interactions. Specific changes were made for the customer for better communication and control between the teams.

Usage of Web 2.0 was made for effective communication and collaboration.

**References**
