

Effectiveness of software quality assurance in offshore development enterprises i...



EFFECTIVENESS OF SOFTWARE QUALITY ASSURANCE IN OFFSHORE

DEVELOPMENT ENTERPRISES IN SRI LANKA Malinda Sirisena, Department of

Computer Science & Engineering, University of Moratuwa. ABSTRACT The aim of the research described in this thesis is to evaluate the effectiveness of software quality assurance approaches of Sri Lankan offshore software development organizations, and to propose a framework which could be used across all offshore software development organizations. An empirical study was conducted using derived framework from popular software quality evaluation models.

The research instrument employed was a questionnaire survey among thirty seven Sri Lankan registered offshore software development organizations.

The findings demonstrate a positive view of Effectiveness of Software Quality Assurance - the stronger predictors of Stability, Installability, Correctness, Testability and Changeability. The present study's recommendations indicate a need for much emphasis on software quality assurance for the Sri Lankan offshore software development organizations. Keywords: Software Quality Assurance (SQA), Offshore Software Development, Quality Assurance Evaluation Models, Effectiveness of Quality Assurance. BACKGROUND

INFORMATION Software Quality Assurance (QA) plays a major role in successful implementation and maintenance of a software project. In many organizations, QA has been simply traded-off to project cost [1]. The motivation of this research is to highlight the value of Software Quality Assurance against the economic cost. The IEEE standard ANSI/IEEE 730-2002 defines software quality assurance as " a planned and systematic pattern of

all actions necessary to provide adequate confidence that the software conforms to established technical requirements”[2].

QA is not only holding a direct relationship of meeting customer satisfaction, but it has a very high impact on project schedules and cost. Failing to pay attention is often resulted in budget overruns and schedule delays [3].

Software Quality Assurance has paid back in many industries such as telecommunication, health, travel, law, hospital, government and schools in many American organizations. • A system of teaching hospitals conservatively estimates \$17. 8 million saved on an investment of \$2. 5 million in quality management over a five-year time period. The University of Pennsylvania saved more than \$60, 000 a year from one project focused on reducing mailing cost. • The U. S. Bureau of Labor Statistics reduced the time needed to produce the monthly Consumer Price Index (CPI), compiled by 650 people in five departments, by 33 percent with no loss in accuracy.

[4] Even in Sri Lankan software engineering companies, have been recognized QA as an important element. In 2005, Affno (www. affno. lk) has won the National Best Quality Software Gold Award for their product - eTender, which developed for Sri Lanka Telecom to automate their tendering process [5].

2 THEORETICAL BASE OF THE STUDY
2. WHAT IS SOFTWARE QUALITY
The IEEE standard ANSI/IEEE 730-2002 defines software quality assurance as “ a planned and systematic pattern of all actions necessary to provide adequate confidence that the software conforms to established technical requirements”[2]. By going down the path of IEEE definition, there are two major camps when defining software quality[6]: 1. Conformance to specification: quality defines in terms of the level which the product or

service meets its' written specifications. 2. Meeting customer needs: meeting customer's explicit or implicit needs, irrespective of any measurable product or service characteristics.

Currently software quality assurance is measured in two ways: from technical perspective and from user perspective[7]. In the technical perspective of measuring software quality is based on specifications.

Developers measure quality and ensure specifications in terms of errors in code through testing process and through other mechanisms such as formal specifications, structured programming[8]. End-user perspective of software quality is measured through user experience to denote how well software meets user expectations. User dissatisfactions do not necessarily be resulting from failure to meet specifications or coding errors. . 2 SOFTWARE

QUALITY MANAGEMENT PHILOSOPHIES This section of the literature presents different philosophies of quality from view points of quality management gurus. These quality management philosophies could be a good alternative to formalized quality models which the research is going to based on. Quality management requires customer satisfaction, prefers prevention to inspection, and recognizes management responsibility for quality[9]. 2. 2. 1 DEMING AND FOURTEEN POINTS FOR MANAGEMENT Walter Edward Deming defines quality in terms of customer satisfaction[10].

Customer satisfaction is beyond conformance to specifications. According to Deming, the judge of quality should be the end user or the customer.

Deming argues that management system should implement in a way that everyone in the organization to be responsible for quality of their output to the internal stakeholders. He introduced fourteen points for management for <https://assignbuster.com/effectiveness-of-software-quality-assurance-in-offshore-development-enterprises-in-sri-lanka/>

people to understand and implement necessary quality transformation[10]:

1. Create constancy of purpose for improvement of product and service: Stay in business and provide jobs through innovation, research, constant improvement and maintenance. 2.

Adopt the new philosophy: For the new economic age, management needs to take leadership for change into a learning organization. 3. Cease

dependence on mass inspection: Eliminate the need for mass inspection by building quality into the product. 4. End awarding business on price: Aim at

minimum total cost and move towards single suppliers. 5. Improve

constantly and forever the system of production and service: Improvement is not a one-time effort. Management is obligated to continually look for ways

to reduce waste and improve quality. 6. Institute training: Workers should be trained properly on their jobs. . Institute leadership: Leading shall consist of

helping people to do a better job and to learn by objective methods. 8. Drive out fear: To assure better quality and productivity, people feel secure. 9.

Break down barriers between departments: Team work culture across

departments. 10. Eliminate slogans, exhortations and numerical targets: Let workers formulate their own slogans. Then they will be committed to the

contents. 11. Eliminate numerical quotas or work standards: Quotas take into account only numbers, not quality or methods. They are usually a guarantee of inefficiency and high cost.

A person, in order to hold a job, will try to meet a quota at any cost,

including doing damage to the company. 12. Remove barriers to taking pride in workmanship: People are eager to do a good job and distressed when they

cannot. 13. Institute a vigorous programme of education: Both management
<https://assignbuster.com/effectiveness-of-software-quality-assurance-in-offshore-development-enterprises-in-sri-lanka/>

and the work force will have to be educated in the new knowledge and understanding, including teamwork and statistical techniques. 14. Take action to accomplish the transformation: It will require a special top management team with a plan of action to carry out the quality mission.

A critical mass of people in the company must understand the 14 points. 2.

2. 2 JURAN AND THE IMPORTANCE OF TOP MANAGEMENT COMMITMENT TO

QUALITY

Joseph M Juran proposes two meanings to quality[11]: 1. Quality consists of those product features which meet the need of customers and thereby provide product satisfaction. 2. Quality consists of freedom from deficiencies. In the handbook Juran propose quality as “ fitness for use” rather than “ meeting customer needs” he argues that it is not a feasible task to meet customer need. His view is much closer to the thought - “ conformance to specifications”.

Juran proposes three fundamental managerial processes for the task of managing quality. The three elements of the Juran Trilogy are[11]: 1. Quality planning: A process that identifies the customers, their requirements, the product and service features that customers expect, and the processes that will deliver those products and services with the correct attributes and then facilitates the transfer of this knowledge to the producing arm of the organization. 2. Quality control: A process in which the product is examined and evaluated against the original requirements expressed by the customer. Problems detected are then corrected. . Quality improvement: A process in which the sustaining mechanisms are put in place so that quality can be achieved on a continuous basis. This includes allocating resources, assigning people to pursue quality projects, training those involved in pursuing

<https://assignbuster.com/effectiveness-of-software-quality-assurance-in-offshore-development-enterprises-in-sri-lanka/>

projects, and in general establishing a permanent structure to pursue quality and maintain the gains secured. 2. 2. 3 CROSBY AND STRIVING FOR ZERO DEFECTS Philip B Crosby is a “conformance to specification” adherer. Crosby summarizes his perspective on quality in fourteen steps that is built around four fundamental “absolutes” of quality management[12]: 1.

Quality is defined as conformance to requirements, not as “goodness” or “elegance” 2. The system for causing quality is prevention, not appraisal. That is, the quality system for suppliers attempting to meet customers’ requirements is to do it right the first time. Crosby is a strong advocate of prevention, not inspection. In a Crosby oriented quality organization everyone has the responsibility for his or her own work. There is no one else to catch errors. 3. The performance standard must be Zero Defects, not “that’s close enough”. Crosby has advocated the notion that zero errors can and should be a target. . The measurement of quality is the cost of quality. Costs of imperfection, if corrected, have an immediate beneficial effect on bottom-line performance as well as on customer relations. 2. 2. 4 ISHIKAWA AND FISHBONE DIAGRAM Kaoru Ishikawa defines quality as “meeting customer needs”[13]. He further argues that no specific quality standard could ever define and following them does not meet the expected quality levels. According to Ishikawa, quality is a very broad concept which goes beyond product, service, process, information quality, etc.

He introduced quality circles through Fishbone diagrams. 2. 2. 5

FEIGENBAUM AND TOTAL QUALITY CONTROL Armand Vallin Feigenbaum built his thought around “total quality control”[14]. Feigenbaum states that quality is a dynamic factor which must be defined in terms of customer <https://assignbuster.com/effectiveness-of-software-quality-assurance-in-offshore-development-enterprises-in-sri-lanka/>

experiences. He further states that quality should satisfy customers' explicit and implicit needs[14].

2.3 SOFTWARE QUALITY MODELS

Previous section focus on different view points of quality management gurus. These points will be helpful in solving common quality management problems in Sri Lankan, offshore enterprises.

Quality management philosophies presented in the previous section represent flexible and qualitative view of quality; this section will present a rigid and quantitative[15] quality structure, which will be a roadmap of identifying independent variables for current study.

2.3.1 MCCALL'S QUALITY MODEL

Jim McCall's quality model is primarily aimed towards the system developers and development process, however he has tried to bridge the gap between users and developers by focusing on number of quality factors, considering both user's and developer's priorities[16, 17].

The quality model is organized around three quality characteristics[16]:

Figure 1: McCall's quality model organized around three types of quality characteristics

McCall's model furthermore elaborated with a hierarchy of factors, criteria and metrics around the three types of major perspectives.

Figure 2: McCall's quality model

Eleven factors on the left-hand side of the model represent the external view of quality as viewed by end users. These eleven factors attribute to twenty three quality criteria, which describe the internal view of software. The evaluation is done by answering each quality criteria with "yes" and "no".

Finally the quality level is derived as a percentage based on the responses received as "yes".

2.3.2 BOEHM'S QUALITY MODEL

Barry W Boehm's model

has similarities to McCall's model. His qualitative approach of defining quality stems from three levels in the hierarchy, which ends with primitive characteristics[18]. These primitive characteristics individually contribute to the overall quality level. Figure 3: Boehm's software quality characteristics tree[19]. Quality measurement is carried out through extent or degree to which the product or service achieves each characteristic[19].

2. 3. 3 ISO 9126

Among the ISO 9000 series of quality standards, ISO has released the ISO 9126: Software Product Evaluation[20]. Figure 4: The ISO 9126 quality model [20]. ISO further proposes quality characteristics/guidelines to evaluate the above six areas of importance. Figure 5: ISO 9126 quality attributes Each quality factor/ six areas of importance is represented by sub-factors as depicted in the above diagram. Details of each selected attribute will be discussed in the next chapter.

3 CONCEPTUAL FRAMEWORK

This chapter elaborates how the conceptual framework for the study has been derived through the existing work identified in the literature review.

1 EXISTING WORK

Since the study is on evaluating software quality from software developing organization's view, it is necessary to filter down the quality attributes discovered in the literature, only to represent developer view of software quality. Therefore it has been decided to take the union of developer related quality attributes from all three popular models referred in the previous chapter. It is not an easy task to differentiate developer oriented quality attributes from user oriented attributes as quality classifications are different from each model and some attributes are subjective to their multiple definitions.

For a consistent interpretation of the quality attributes, the definitions of attributes have been used according to Software Engineering Institute's (SEI) Software Technology Roadmap glossary[23] and ISO 9126[24] definitions.

3. 1. 1 DEVELOPER ORIENTED ATTRIBUTES FROM MCCALL'S MODEL McCall's model mainly goes hand in hand with external quality factors. Following are the quality attributes extracted from McCall model, which are related to developer related quality based on SEI definitions. Selected Attribute Maintainability SEI Definition[23] " The ease with which a software system or component can be modified to correct faults, improve performance, or other attributes, or adapt to a changed environment. " " The degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met. " " The ease with which a system or component can be modified for use in applications or environments other than those for which it was specifically designed. " " The ease with which a system or component can be transferred from one hardware or software environment to another. " The degree to which a software module or other work product can be used in more than one computing program or software system. " " The ability of two or more systems or components to exchange information and to use the information that has been exchanged. " Testability Flexibility Portability Reusability Interoperability

Table 1: Developer related quality attributes from McCall's model

3. 1. 2 ADDITIONAL ATTRIBUTES FROM BOEHM'S MODEL Boehm's model, which has put the utility perspective in terms of quality, is much similar to McCall's model.

After evaluating definitions, following two attributes were added to the list.

Selected Attribute Understandability Modifiability SEI Definition[23] “ The degree to which the purpose of the system or component is clear to the evaluator. ” “ The degree to which a system or component facilitates the incorporation of changes, once the nature of the desired change has been determined. ” Table 2: Additional developer related quality attributes from Boehm’s model 3. 1. 3 ADDITIONAL ATTRIBUTES FROM ISO 9126 Following are sub-attributes taken from the ISO 9126 definitions.

Selected Attribute Analyzability ISO Definition[24] “ The capability of the software product to be diagnosed for deficiencies or causes of failures in the software, or for the parts to be modified to be identified. ” “ The capability of the software product to enable a specified modification to be implemented. ” “ The capability of the software product to avoid unexpected effects from modifications of the software. ” “ The capability of the software product to be adapted for different specified environments without applying actions or means other than those provided for this purpose for the software considered. ” “ The capability of the software product to be installed in a specified environment. ” “ The capability of the software product to co-exist with other independent software in a common environment sharing common resources. ” Changeability Stability Adaptability Installability Co-existence Replaceability “ The capability of the software product to be used in place of another specified software product for the same purpose in the same environment. ” Table 3: Additional developer related quality attributes from ISO 9126 model 3. 1. 4 FINAL ATTRIBUTE LIST

After analyzing the above mentioned attribute lists and completing the preliminary studies, the list could filter down to the following for the current study. 1. Correctness 2. Testability 3. Changeability 4. Stability 5.

Installability In the following sections, each of above attribute will be discussed in terms of their quality characteristics. 3. 1. 4. 1 CORRECTNESS SEI defines correctness as “ The degree to which a system or component is free from faults in its specification, design, and implementation”[23]. McCall attributes correctness through[16]: • • • Traceability Completeness Consistency

Through traceability, it makes possible to know the relationships of each module or component and thereby higher confidence states correctness. Completeness assures that there are no parts left in terms in executing a function of a system or a procedure; thereby 100% completeness ratio guarantees correctness. Inconsistent systems or functions will lead to higher error probability; therefore it is a part of correctness. Through the initial discussions with some key personnel, it was revealed that these characteristics are equally hard to reach to achieve Correctness. . 1. 4. 2 TESTABILITY SEI defines testability as “ The degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met”[23]. Both McCall and Boehm have attributed testability to quality assurance on following characteristics[16, 18]: • • • • • • • Simplicity Instrumentation Self-descriptiveness Modularity and structuredness Accountability Accessibility Communicativeness. Simplicity of applications will make easier in testing comparatively to complex applications.

Instrumentation makes possible to put probes in the system in order to deduce test data. Self-descriptive systems have inbuilt help or system documentation which will be sufficient to understand the system by going through. Modularity helps in isolating system tests which structuredness denotes consistent organization of the system. Accountability on system for which it is possible to measure the usage of the code[19]. Such measurements are typically covered by debugging tools, which exist specifically for programming languages. Accessibility of a system allows usage of its parts in a selective manner[19].

This allows in creating flexible test scenarios. Through communicativeness, systems make easier to understand inputs and output, which makes easier to compose test cases. 3. 1. 4. 3 CHANGEABILITY ISO defines changeability as “ The capability of the software product to enable a specified modification to be implemented”[24]. Changeability is an attribute defined in ISO 9126 and lacks supporting characteristic definitions. However changeability could be achieved through:

- Aiming simple solution rather than complicated systems as by nature simple applications are easier to change. Low coupling of individual modules of a system as lower interactions make easier to change individual components.
- Designing the systems change in mind from the beginning while keeping application evolution.

3. 1. 4. 4 STABILITY ISO defines stability as “ The capability of the software product to avoid unexpected effects from modifications of the software”[24]. Therefore stability in this context does not denote the ability of the system to show stable behavior when used. However, if modification often results in unexpected behavior, there will be a high impact on stability.

Stability is directly influenced by Changeability. Low changeability is likely to show low stability. This will depict the fact that, trying to change a low changeable system will lead to a greater risk of instability. 3. 1. 4. 5

INSTALLABILITY ISO defines Installability as “ The capability of the software product to be installed in a specified environment”[24]. Installability requirements are generally specified in the form of an installation process. The target environment in this case will have to be known at the development time.

Installability is measured as a percentage exercised of the total specified Installability requirements. In the Sri Lankan context, Installability is commonly referred as Deployability. 3. 1. 5 RELATIONSHIPS OF VARIABLES Having identified the variables and attributes, it had been decided to limit the study to following variables, after interviewing key quality assurance personnel in target organizations. Based on their arguments, on applicability to offshore organizations, the best suited variables have been selected for the study. Dependent Variable: Effectiveness of Software Quality Assurance Independent Variables: . Correctness a. Completeness b. Consistency 2. Testability a. Simplicity b. Modularity c. Structuredness 3. Changeability a. Simplicity b. Coupling 4. Stability a. Changeability 5. Installability Having identified the variables, following relationships have been derived based on the reviewed literature in the previous section. Correctness Testability Effectiveness of Software Quality Assurance Changeability Stability Installability Independent Variables Figure 6: Schematic diagram for conceptual framework Dependent Variable 3. 2 HYPOTHESES FORMULATED

In order to statistically test the derived conceptual framework, following hypotheses have been formulated. Since the study is targeted to test each independent variable separately, hypotheses also have been formulated independently to each independent variable. H01: there is no relationship between the Correctness of software developed and released to QA team), on the effectiveness of software quality assurance approach. HA1: the greater the Correctness of software developed and delivered to QA team, the higher the effectiveness of software quality assurance approach.

H02: there is no relationship between the Testability of software developed and released to QA team, on the effectiveness of software quality assurance approach. HA2: the greater the Testability of software developed and delivered to QA team, the higher the effectiveness of software quality assurance approach. H03: there is no relationship between the Changeability of software developed and released to QA team, on the effectiveness of software quality assurance approach. HA3: the greater the Changeability of software developed and delivered to QA team, the higher the effectiveness of software quality assurance approach.

H04: there is no relationship between the Stability of software developed and released to QA team, on the effectiveness of software quality assurance approach. HA4: the greater the Stability of software developed and delivered to QA team, the higher the effectiveness of software quality assurance approach. H05: there is no relationship between the Installability of software developed and released to QA team, on the effectiveness of software quality assurance approach. HA5: the greater the Installability of software

developed and delivered to QA team, the higher the effectiveness of
<https://assignbuster.com/effectiveness-of-software-quality-assurance-in-offshore-development-enterprises-in-sri-lanka/>

software quality assurance approach. RESEARCH DESIGN Research design will outline the roadmap of achieving the research objectives through the identified variables and theoretical framework. Details of study Purpose of the study Type of investigation Extent of researcher interface Minimal: studying events as they normally occur and defining a framework Study setting Measurement Measurement and measures Effectiveness of Software Quality Assurance in Emerging Offshore Development Enterprises in Sri Lanka Descriptive: quality evaluation framework Hypothesis testing: to validate the evaluation framework

Correlation: study of correlations to effectiveness against evaluation factors

Noncontrived: study in real business environment Quality factors and their applicability through quality matrices and Likert scales Data analysis 1.

Classification of data 2. Goodness of data Unit of analysis Sampling design

Time horizon Data collection method 3. Hypotheses testing Individuals based on job categories in Offshoring organizations Judgmental sampling of

individual in the entire population of offshore enterprises Crosssectional

Interviews, Questionnaires, Observations Figure 7: The research design 4. 1

TYPE AND NATURE OF THE STUDY

The study was an empirical study through analysis of responses to the questionnaires which was formulated through the conceptual framework. 4.

2 DATA COLLECTION METHODS Since the study is on offshore software development organizations, it has been decided to collect data from all registered companies in Software Exporters Association Sri Lanka and seven other offshore software development organizations in Sri Lanka. There were

forty seven registered members as of first August, 2007. Questionnaires <https://assignbuster.com/effectiveness-of-software-quality-assurance-in-offshore-development-enterprises-in-sri-lanka/>

were distributed to the key quality assurance person or to the most senior quality assurance person in each organization. . 2. 1 QUESTIONNAIRE DESIGN A structured questionnaire was used to gather responses apart from the preliminary interviews. The questionnaire is divided in to four main sections. Section one has eleven questions, capturing organizational demographics of the responder. Section two has six questions, to capture responder's personal demographics. Section three is the main section of the questionnaire which captures organizations' software quality assurance, project specific demographics and responses to test the conceptual framework. Section four is targeted to capture additional information for the conceptual framework. RESULTS OF DATA ANALYSIS Responses received had been categorized to qualitative data and quantitative data. Qualitative data had been used to understand the responder's and company background. Quantitative responses, where the scale data is measured have been assigned scores as per following table for statistical analysis. Response Selected Strongly disagree Disagree Neutral Agree Strongly agree Score Assigned 1 2 3 4 5 Table 5: Rates given for questionnaire responses Each response was individually assessed to ensure data validity and integrity. Incomplete responses have been followed up with the responder with available contact information and have been able to complete in many instances. For the blank responses, score three was assigned in case the question is not applicable to the responder's organization. Following summary shows the statistics of the questionnaire distribution and responses received. Number of Organizations that Questionnaire had been sent 47 SEA

registered companies + 7 other offshore companies Total Responses

Received 39 Invalid / Unusable 2 Number of Valid Responses 37

Table 6: Statistics of questionnaire distribution responses received

5. 1 PILOT STUDY To test the primary data a pilot study was run among fourteen Quality Assurance Engineers at an offshore software development organization, using a draft questionnaire. On the scale of reliability in order to treat results with credibility[25] and the internal consistency of the draft questionnaire, was checked by using Cronbach's alpha coefficient. The alpha coefficient should be above . 7 for the scale to be reliable[26]. The overall Cronbach's alpha coefficient was . 81, thus the questionnaire was considered to have a good internal consistency and suitable for collecting the data for the main study. Details of Cronbach's alpha are discussed under Analysis of Reliability Section, below.

5. 2 PRELIMINARY ANALYSIS All thirty seven organizations selected as valid responses are exporting software. 89. 19% of the selected organizations are locally owned while 10. 81% of organizations which are in Sri Lankan operation are owned by foreign parties. 64. 86% of the target organizations are project based companies while 21. 2% of the organizations focus only on their own products. However 13. 51% of the organizations undertake client projects while they market their own products.

10 8 No. of Organizations 6 4 2 0 1. 00 2. 00 3. 00 4. 00 5. 00 6. 00 7. 00 8. 00 12. 00 14. 00 No. of years in Sri Lankan Operation

Figure 8: Analysis of organizations against number of years in operation According to the above graph, most of the Sri Lankan offshore organizations under the current study have started their operation two years before. 75. 68% of the responders were males and the balance 24. 32% were females.

The average age of responders was 30.11 years. On an average, they possess one year of experience in their current position in the respective organizations. The following chart represents the education level of responders.

Education Level	Count
Non IT Graduate	30
IT/Comp. Science Graduate	25
Post Graduate Diploma	20
MSc/MBA/Post Graduate Degree	15
Other Education Level	10

Figure 9: Education level of responders. Majority of quality assurance heads in the target organizations possess Information Technology or a Computer Science degree. 3.03% Little Early, 9.09% On Time, 24.24% Too Delayed, 3.64% Little Delayed.

Figure 10: Project completion against estimates. Responders were asked to select a completed project/product when they responded to part 3 of the questionnaire. The above pie chart highlights the project/product completion time against the estimates of the selected projects by the responders. From the selected projects/products, majority have been completed with a little delay from the estimates. Mean and the variance are calculated for each question under each independent variable and the dependent variable through the assigned scores as per Table 5.

Question No.	Question	Mean	Variance	Effectiveness of Software Quality Assurance
18	Software QA is a very important discipline in our organization	3.622	0.686	0.99
19	Without QA our products/services will not meet current level of customer satisfaction	4.081	0.99	0.99
20	Our Software QA approach/practice helps us in winning new businesses	3.811	0.686	0.99
21	Our organization has adequate number of QA Human Resources	3.919	0.686	0.99
22	Our organization has invested enough in Software QA tools	3.514	0.686	0.99
23	Our Software Development or any other Process has considered QA as a major practice	3.865	0.686	0.99

658 0. 465 0. 812 0. 842 Correctness 30 31 32 33 34 35 36 “ If the systems or components we deliver meet specifications to 100%”, we can say that it’s a high quality factor Systems or components we deliver, always met specifications Uniformity of functionality/operations/navigation of the designed system always contributed to high quality System maintained Uniformity of functionality/operations/navigation across individual functions If a function of a system, completes its execution without in between failures, we can say it is a high quality factor.

Our systems do not fail in executing a function or procedure to its completion Our QA team measures our systems, whether they meet specifications or not
3. 703 3. 568 3. 703 3. 324 3. 243 3. 243 4. 108 0. 604 1. 141 0. 715 1. 003
0. 745 0. 634 0. 544

Testability 37 If all functionality/operations/navigation of systems could be tested enough, then we can say it denotes high quality All the functionality/operations/navigation of our systems are properly being tested by our QA team Even the complex operations of our systems are represented by simple user interactions in order to make applications simple and user friendly Our applications are decomposed in to manageable modules in implementation in a practical manner Consistent organization of modules/code are evident in our applications Our QA team measures or put emphasis on testability (Simplicity, Modularity, structuredness) of applications during the QA cycle 4. 595 0. 303 38 4. 514 0. 312 39 4. 297 0. 270 40 3. 946 0. 330 41 3. 838 0. 417 42 4. 432 0. 308 Changeability 43 If a product allows a specified modification to be implemented without much

difficulty, then we can say it denotes a high quality factor Our systems do not need much effort to accommodate minor specification changes (i. e.

Adding a new field to a form) at implementation or quality assurance stage

Our systems maintain low interactions between individual modules,

therefore it is easier to change individual components without affecting

others Our QA team measures put much emphasis to test changeability and

stability of systems during the QA cycle 4. 000 0. 111 45 3. 946 0. 164 46 3.

838 0. 251 48 3. 919 0. 299 Stability 44 If the systems avoid unexpected

effects after modifications, it denotes a high quality or it's a high quality

factor After the design changes done to one module, our systems have very

few side effects to other modules Our QA team measures put much

emphasis to test changeability and stability of systems during the QA cycle

3. 595 . 359 47 3. 703 0. 437 48 3. 919 0. 299 Installability 49 If the system

could be installed in a specified environment without challenges, it denotes

high quality or it can be considered as a high quality factor Our systems do

not get challenged during the installation in the agreed/specified

environment Our QA team measures Installability of systems they test 3. 568

0. 863 50 3. 162 3. 541 0. 862 1. 311 51 Table 7: Means and variances of

questions Frequency distributions of responses to each of above questions

have been presented in Appendix 2. 5. 3 SECONDARY RESULTS ANALYSIS

Primary data is further analyzed to derive more meaningful results.

For statistical analysis, the ratings gathered through individual questions

were summed up to derive scores for individual independent variables.

Variable = sum of marks for relevant questions I. e. Correctness = Q30 +

Q31 + Q32 + Q33 + Q34 + Q35 + Q36 Sample Mean, where, n = sample

<https://assignbuster.com/effectiveness-of-software-quality-assurance-in-offshore-development-enterprises-in-sri-lanka/>

size, and = scores Sample Variance, Standard Deviation, Following table illustrates the statistics of independent variables, which denotes the effectiveness of quality assurance. Standard Deviation 0. 569 0. 552 0. 422 0. 327 0. 445 0. 752 Variable Effectiveness of QA Correctness Testability Changeability Stability Installability Mean 3. 802 3. 556 4. 270 3. 926 3. 739 3. 423 Variance 0. 324 0. 305 0. 178 0. 107 0. 198 0. 566

Table 8: Basic statistics of independent variables and the dependent variable

Following is the graphical illustration of above statistics. 4. 500 4. 000 3. 500 3. 000 2. 500 2. 000 1. 500 1. 000 0. 500 0. 000 Mean Variance Std. Div.

Figure 11: Basic statistics of independent variables According to the above illustration, Testability contributes to QA effectiveness most while

Changeability remains at the second position. Installability was rated as of least significant to the QA Effectiveness in the subject domain. 5. 3. 1

ANALYSIS OF RELIABILITY OF DATA Cronbach's alpha measure is used to determine how well the target independent variables measure single, unidimensional QA Effectiveness latent construct.

Cronbach's alpha can be written as a function of the number of test items AND the average inter-correlation among the items. N where, N = number of items and r = inter-item correlation among items. Cronbach's Alpha Based on Cronbach's Standardized Alpha (Items . 912 . 918 Table 9: Reliability statistics N of Items 28 Cronbach's alpha for all twenty eight questions is 0. 912, which denotes that the collected data is acceptable for the research. 5.

4 HYPOTHESES TESTING Analysis had been done to test each set of

hypothesis to find out whether there are relationships defined through the

hypotheses exist among independent variables and the dependent variable.

<https://assignbuster.com/effectiveness-of-software-quality-assurance-in-offshore-development-enterprises-in-sri-lanka/>

The correlations between the factors hypothesized to effectiveness of quality assurance shown in the following table: Set of Hypothesis/Independent Variable H1: Correctness H2: Testability H3: Changeability H4: Stability H5: Installability ** Correlation is significant at the 0. 01 level (2-tailed). Pearson Correlation/ Effectiveness of QA . 678** . 589** . 559** . 728** . 613** Sig. (2-tailed) . 000 . 000 . 000 . 000 . 000 Table 11: Correlations between hypotheses for quality assurance Hypothesis H1: According to Hypothesis H01, Correctness which is influenced by Consistency and Completeness has a positive relationship to effectiveness of software quality assurance approach. Since this hypothesis is supported by the data analysis (Sig. value was . 000, p