*Advances in Science, Technology and Engineering Systems Journal Vol. 2, No. 1, 6-15 (2017)*

www.astesj.com

*Special Issue on Computer Systems, Information Technology, Electrical and Electronics Engineering*

**ASTESJ**

**ISSN: 2415-6698**

# System Testing Evaluation for Enterprise Resource Planning to Reduce Failure Rate

Samantha Mathara Arachchi*, Siong Choy Chong, Alik Kathabi

*Management and Science University (MSU), University Drive, Off Persiaran Olahraga, Section 13, 40100 Shah Alam, Selangor Darul Ehsan, Malaysia*

*Email: saman17lk@yahoo.com, eddyscchong@yahoo.com, alik@msu.edu.my*

ARTICLE INFO

*Article history:*
*Received: 26 November, 2016*
*Accepted: 23 December, 2016*
*Online: 28 January, 2017*

*Keywords:*
*ERP*
*Risk factors*
*Security risks*
*Failures and Evaluations*

ABSTRACT

*Enterprise Resource Planning (ERP) systems are widely used applications to manage resources, communication and data exchange between different departments and modules with the purpose of managing the overall business process of the organization using one integrated software system. Due to the large scale and the complexity nature of these systems, many ERP implementation projects have become failure. It is necessary to have a better test project management and test performance assessing system. To build a successful ERP system these processes are important. The purpose of the Test project management is verification and validation of the system. There was a separate stage to test the quality of software in the software development lifecycle and there is a separate independent Quality Assurance and testing team for a successful ERP development team. According to best practice testing principles it is necessary to, understand the requirements, test planning, test execution, identify and improve processes. Identify the necessary infrastructure; hardware and software are the major areas when developing test procedures. The aim of this survey is to identify ERP failures associated with the ERP projects, general and security within the Asian region, so that the parties responsible for the project can take necessary precautions to deal with those failures for a successful ERP implementation and bring down the ERP failure rate.*

## 1. ERP Testing Overview

System testing checklist is an excellent way to ensure the success of a software testing process during the software development life cycle as it helps to make proper planning for any particular software [1], based on its requirements and expected quality [2,3]. Software testing plays a major role in any software development life cycle or methodology. Therefore, it has to be done accurately by professional software testers. Although professional testers have a good knowledge and experience in software testing, there are situations where they also create a big mess with the test planning and process testing activities. So the best way to solve this problem is to prepare a system testing checklist in order to make an accurate and suitable plan to test each software [1]. When a QA team has checklists, they can easily process all the required test activities in proper order without missing anything [2, 3]. A system testing checklist is a document or may be a software tool that systematically plans and prepares software testing and helps to define a framework for test environment, testing approach, staffing, work plan, issues, test sets and test results summary. As mentioned earlier the software testing phase is vital for any implementation team because, if the developed system doesn't show the expected quality it would be a total waste of time and money since clients may not prefer to use the software in the future [2,3,4].

## 2. Literature Review

ERP software quality evaluation system developed on the bases of the testing [1]. Finally, the project manager must assess the system and also the factors used for the new system Therefore, it is important to maintain documents for each phase. In order to do this the allocated and used resources must be compared and it is also important to assess the test performances [5].

### *2.1. Requirements for Test Procedure*

The comprehensive evaluation and the comparison of different testing strategies for scenarios is a critical part in the

[*] Corresponding Author: Samantha Mathara Arachchi, Management and Science University (MSU), University Drive, Off Persiaran Olahraga, Section 13, 40100 Shah Alam, Selangor Darul Ehsan, Malaysia, Email: saman17lk@yahoo.com

https://dx.doi.org/10.25046/aj020102

current research on software testing [6]. Mutation analysis helps to determine the effectiveness of a test strategy and compare the different test strategies based on their effectiveness measures [7]. Test strategy which is reliable for all programs cannot be established [6]. Boolean and Relational Operator (BRO) and Boolean and Relational Expression (BRE) testing are two condition based testing strategies [8]. These are unlike the existing condition based upon the detection of errors such as Boolean errors and relational expression error in a condition [8]. In accordance with the empirical studies of the algorithm SBEMIN and SBEMINSEN along with theoretical properties of BRO and BRE it is confirmed that BRO and BRE testing are practical and effective for testing programmes with complicated conditioning [8].

Software based self-test strategy is well suited for a low cost embedded system, which does not require immediate detection of error [9].

Three software state of the practice testing strategies are: code reading by stepwise abstraction, functional testing using equivalence portioning and boundary value analysis, and finally structural testing using 100% statement coverage criteria [10]. Adaptive testing is an online testing strategy, adapted to reduce the variance in the results of the software reliability assessment [11]. Complex system test strategies are derived through the performance of the software, calculated based on the computer line of code. Test policies for the complex system are chosen based on the derived test strategy [12].

Optimally refined proportional sampling is a strategy which is simple and low in cost [13]. Empirical study was made through a sample programme with seeded error, and found that this strategy is better than random testing [13].

Test data generation strategy is used to generate the test data [7]. Test strategy can only be considered reliable for a particular programme, if and only when it produces a reliable bunch of test data for that programme [7].

Path analysis testing strategy is a method used to analyze the reliability of the path testing. In this strategy data are generated which enables the different paths of the system to be executed [14] and [13].

Debugging strategy based on the requirement of testing is focused on the situation where the selected testing requirement does not indicate the fault site but apparently provides helpful information for fault localization [15].

Path pre fix testing strategy is a user interactive. Adaptive testing strategy [16] and Test path which are used previously will be used to select the subsequent paths for testing [16].

### 2.2. Instruction for Test Procedures

It is necessary to have a better test project management and test performance assessing system. To build a successful ERP system these processes are important. Test project management is different from the other project management roles [3]. The purpose of the Test project management is verification and validation of the system.

There was a separate stage to test the quality of software in the software development lifecycle and there is a separate independent Quality Assurance and testing team for a successful ERP development team. According to best practice testing principles it is necessary to, understand the requirements, test planning, test execution, identify and improve processes (Process Improvement, Defect Analysis, requirements review and risk mitigation) [17].

A Development Model's Implications for testing is based on: Review the user interface early, Start writing the test plan as early as possible, Start testing when the work is on the critical path, Plan to staff the project very early, Plan waves of usability tests as the project grows more complex, Plan to write the test plan and Plan to do the most powerful testing as early as possible [18,19]. Identify the necessary infrastructure; hardware and software are the major areas when developing test procedures.

### 2.3. Get Ready for Test Preparation

Emphasis is on the typical test description or the schedule and milestones for each of the tasks. Therefore, the project manager has complete control over the project. The development timeline will be shown as; Product Design, Fragments coded: first functionality, Almost alpha, Alpha software, Pre-beta, Beta, User Interface Freeze, Pre-final and Final Integrity Test Release [18-21,3].

### 2.4. Create Test Plan

As soon as the test plan is created incorporating risk management, communication management, test people management and test cases, it has to be reviewed by an independent group and approved by an authority that is not influenced by the project manager responsible for the testing [4].

Test plans help to achieve the test goal in an organized and planned manner. It says, build test plan, define metric objective and finally review and approve the plan as a step involved in the development of test plan [18]. Test plan is an ongoing document as it changes when the system changes, and it is very true in the spiral environment.

Test plans are the main factors which lead to the success of the system testing [4]. Testing is done at different levels and the master test plan is considered to be the first level in order to follow the methodology of testing.

In addition to the master level test plan, level specific test plan according to the phases such as acceptance test, system test, integration test and unit test plans will be created. The main goal of test plan is to address issues such as test strategy and resource utilization.

### 2.5. Create Test Cases

Software failures in a variety of domain have major implications for testing and further emphasize on exhaustive testing of computer software where all errors in the system are identified by a combination of fewer parameters and testing the n-tuples of parameters, which enable effective testing [22,23].

"Proportional Sampling Strategy" which facilitate partition testing, i.e. testing in each partition has higher probability in identifying faults compared to random testing. The amount of test cases used in each partition will be proportional to the size of the partition [24]. Therefore, identifying and naming each test case is significant.

A good set of test cases would have to develop and most of the developers have similar experiences in developing software products rather than responding to a good test set [25]. Since software programmes have many branches, paths or sub-domain, testing every part is not practical. Test cases for special functions and cases are important [26].

### *2.6. Testing Methods*

There are several testing methods that incorporate to test the ERP application. These methods can be used in a different stage as well as for different purposes.

Structural Testing: the software entity is viewed as a "white box". The selection of test cases is based on the implementation of the software entity. The expected results are evaluated on a set of coverage criteria. Structural testing emphasizes the internal structure of the software entity; [27,28]. Grey-box testing, is defined as testing software while already having some knowledge of its underlying code or logic [28]. The testers are only aware of what the software is supposed to do, not how it does it [29, 28]. Black-box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, state transition tables, decision table testing, fuzz testing, model-based testing, use case testing, exploratory testing and specification-based testing [30]. Validation is basically done by the testers during the testing. While validating the product if some deviation is found in the actual result as against the expected result then a bug is reported or an incident is raised. Hence, validation helps in unfolding the exact functionality of the features and helps the testers to understand the product in a much better way. It helps in making the product more user friendly [18, 28].

### *2.7. Testing Levels*

Testing Levels have been used to test the application in a different level. There are different testing levels also to make sure of the accuracy of the Applications.

Unit testing is the basic level of testing that focuses separately on the smaller building blocks of a program or system. It is a process of executing each module to confirm that each performs its assigned function. It permits the testing and debugging of small units. Therefore, it provides a better way of integrating the units into larger units [3]. Integration testing is testing for whole functionality and also for the acceptance of them. It also verifies some nonfunctional characteristics. Some examples of system testing include usability testing, stress testing, performance testing, compatibility testing, conversion testing, and document testing and so on, as described by Lewis [3]. System testing is tested as a whole for functionality and fitness of use based on the system test plan. Systems are fully tested in the computer operating environment before acceptance testing carried out. The source of the system tests are the quality attributes specified in the software quality assurance plan. System testing is a set of tests to verify quality attributes [3].

### *2.8. Testing Types*

ERP type of testing is a continuous task performed before implementation of the system and even after being implemented in order to ensure the quality of the ERP system. ERP systems are very critical to a company's operation as all its functions are integrated into one holistic system. According to William E. Lewis in 2005 and 1983, apart from the traditional testing techniques, various new techniques necessitated by the complicated business and development logic were realized to make software testing more meaningful and purposeful [3]. Software testing is an important means of assessing the software to determine its quality [17]. Since testing typically consumes 40 ~ 50% of development efforts and consumes more effort for systems requiring higher reliability it is an essential part of software engineering [27]. Various techniques reveal different quality aspects of a software system, and there are two main categories of techniques, functional and structural [27].

Alpha testing of an application is performed when development is nearing completion; minor design changes may still be made as a result of such testing. Typically, the alpha testing is done by end-user or others, not by programmers or testers [3]. Deciding on the entry criteria of a product for beta testing and deciding the timing of a beta test poses several conflicting choices to be made. Therefore, the success of a beta programme depends heavily on the willingness of the beta customers to exercise the precut in various ways, being well aware that there may be defects [3, 4].

The load testing is an application tested under heavy loads. Such testing of ERP applications run through the internet under a range of loads to determine at what point the system's response time degrades or fails [3].

Volume test checks are applied to test whether there are any problems when running the system under test with realistic amounts of data, or even a maximum. Typical problems are full or nearly full disks, databases, files, buffers, counters that may lead to overflow. Maximal data amounts in communications may also be a concern [31]. Acceptance Testing is applied to test whether the software products meet the requirements of the users or contract. [32]. This ensures the overall acceptability of the system.

Automated acceptance testing has been recently added to testing in agile Software development process [33]. The concern is on good communication and greater collaboration which enable testing in each stage to decide whether all functions meet the requirements and push the process to arrive at an acceptable software product [34].

The Compatibility testing is done to check how well or not the software performs in a particular hardware, software, operating system, network environment and other infrastructure compatible [3]. Conformance testing, also known as compliance testing, is a methodology used in engineering to ensure that a product, process, computer programme or system meets a defined set of standards.

Manual trial and error testing approach has made the ERP system complicated. There is a growing demand for test automation due to a shortfall in manual testing such as, high cost and resource utilization, problem in detecting errors, and timeline issues in the manual testing [12]. Not all the testing should be automated, only up to 40% -60% should be automated [31,35].

Software testing is an important, but time consuming and costly process in the software development life cycle [36]. Testing process will be efficient only if the testing tool would enable to organize and present it in an easily understandable manner in addition to executing the test [37].

Factors that support the use of test automation are, low human involvement in testing and steady underlying technology reusability [38]. In Industry, companies are trying to achieve efficiency in testing through automating the software testing for both hardware and software components [36]. Currently, testing tools have enhanced to a level where one can automate the interaction with software systems at the GUI level [39]. Even though there is such advancement, the industry currently faces a lack of knowledge on the usability and applicability of testing tools [40].

Preparing application for the automated testing, organizing the automation team members, designing automation test plan, defining criteria to measure the success of automation, developing appropriate test cases, appropriate selection of test tools and finally, implementation of the automated testing are considered test automation best practices to be followed [1,41]. Software test automation framework (STAF) is a multi-platform, multi-language approach based on the underlying principle of reusable service [36]. STAF is used to automate major activities of the testing process and automate resource intensive test suit [36].

The use of test automation started in the mid-1980s after the invention of the automated capture or replay tool [17]. Now we have very sophisticated tools used throughout the testing process such as Regression testing tool, Test design tools, Load/performance tools Test management tools Unit testing tools, Test implementation tools, Test Evaluation tools, Static test analyzer Defect Management tool, Application performance monitoring /tuning tool and Run time analysis testing tool [17,27].

System testing or software testing is a major area in the software industry. In traditional waterfall development life cycle the system testing / software testing phase plays a big role. There are several definitions available and used for system testing, and according to the IEEE definition, system testing is "the process of analyzing a software item to detect the differences between existing and required conditions (that is, bugs) and to evaluate the features of the software item". Basically, this means that the main intention of system testing is to identify whether there is a gap between the developed functions or system with the expected outcomes which are defined according to the previous and the current customer requirements. "Validation" and "Verification" are very important processes in system testing. There are various types of techniques and methodologies used in the software development industry to test and debug the software system and these techniques are different from one another according to the type of system or software or the organization perspective. System testing is an overall activity which started from the requirement analysis phase and goes through each and every phase in the software/ system development life cycle.

### *2.9. Verification Tests*

Verification is intended to check that a product, service, or system meets a set of design specifications. In the development phase, verification procedures involve performing special tests to model or simulate a portion, or the entirety of a product, service or system. The verification procedures involve regularly repeating tests devised specifically to ensure that the product, service, or system continues to meet the initial design requirements, specifications, and regulations as time progresses [42].

### *2.10. Vulnerability Testing*

Vulnerability analysis is a process that defines, identifies, and classifies the security holes (vulnerabilities) in a computer or in an application such as ERP, network, or communications infrastructure. In addition, vulnerability analysis can forecast the effectiveness of proposed countermeasures and evaluate their actual effectiveness after they are put into use [3].

## 3. Research Method

This research employs the descriptive method. This method is used in any fact-finding study that involves adequate and accurate interpretation of findings. Relatively, the method is appropriate to the study since it aims to describe the present condition of ERP failure analysis [43].This method describes the nature of a condition as it takes place during the time of the study and explore the system or systems of a particular condition at each and every phase of the SDLC.

Specifically, direct-data survey using questionnaire was used in the study. This is a reliable source of first-hand information because, since the researcher directly interacts with the respondents, rational and sound conclusions and recommendations can be derived at. The respondents were given ample time to assess the failures in the ERP testing faced by the software development companies in Sri Lanka. Their own experiences at the testing phase of software development are necessary in identifying their strengths and limitations.

The primary data are derived at from the responses provided by the respondents through the self-administered questionnaires prepared by the researcher. The constructs are based on recent literatures related to reducing the failure rate of ERP in the software development industry in Sri Lanka as well as the challenges and the concepts cited by respondents during the pre-survey. In terms of approach, the study employed the quantitative approach which focused on obtaining numerical findings.

## 4. Variables and Hypotheses

Variables are anything that can take on differing of varying values [44]. Based on the literature, the following variables have been identified as shown in the research framework. Independents variables will usually affect the expected output, that is, the dependent variable. The System Testing is the independent variables in this study. A dependent variable is a measure based on the independent variable. In other words, the dependent variable responds to the independent variable. ERP failure has been considered as the dependent variable in this study. A mediating variable can influence the relationship between an independent variable and the dependent variable. In other words, it determines whether the indirect effect of the independent variable on the dependent variable is significant [45]. Furthermore a variable may be considered a mediator to the extent to which it carries the influence of a given independent variable to a given dependent variable. In this research, the testing method mediators which is tested through the following hypotheses:

H1 There is a negative relationship between System Testing and ERP Failure

H2 Many Testing Methods do mediates the positive relationship between System Testing and ERP Failure

## 5. Study Setting

The researcher has critically examined the ERP development companies under the Companies Registration Act and found that such development work is undertaken by software development companies registered under the Sri Lanka BOI and the Public Limited Companies (PLC) Act of Sri Lanka. The software development companies registered under the PLC Act have not been considered in this study due to their lack of investment capacity and involvement of ERP application development [46]. In addition, the PLCs do not have enough capital and resources as well as bank guarantee when signing vendor agreements [47]. Hence, they are not engaged in developing total solutions for ERP applications. Further, there are some companies which registered under the BOI but having less than twenty employees, making little investments and having less capital. Such PLCs have not been considered as well. The same justification has been applied for the BOI companies to select the appropriate ERP development companies for data collection.

The direct-data survey aims at collecting pertinent data to achieve the research objectives. Accordingly, direct-data survey is used to reveal the status of some phenomenon amongst the people identified who are engaged in developing ERP applications at software testing phase of the SDLC in the software development industry in Sri Lanka.

## 6. Unite of Analysis

The unit of analysis consists of employees engaged in testing ERP applications at the phase of the SDLC in software development industries in Sri Lanka. All of the respondents were selected using stratified sampling method by considering staff members with similar educational qualifications and working experience. Under this sampling method, each member of a population has an equal opportunity to become part of the sample based on the software testing phase of SDLC prescribed. As all the members have an equal chance of becoming research participants, this sampling method is said to be the most efficient sampling procedure [44]. Using this sampling strategy, the researcher first defined the population, that is, 48 companies registered under the BOI, and then listed all the members and selected members to determine the sample based on the testing phase of the SDLC.

### *6.1. The Survey Questionnaire*

There were 400 participants for the questionnaire survey in soft testing stage of the SDLC. Questionnaire was given to each participant individually or the head of divisions in each and every software development company.

### *6.2. Content Analysis*

The survey items in the study were developed as a result of analyses of previous studies, discussions with practitioners in the field as well as a review of relevant literature which also takes into consideration the research framework and hypotheses developed.

Reliability and validity are important aspects of questionnaire design. According to Suskie (1996), a perfectly reliable questionnaire elicits consistent responses [48]. Robson (1993) indicates that a highly reliable response is obtainable by providing all respondents with the same set of questions. Validity is inherently difficult to establish using a single statistical method. If a questionnaire is perfectly valid, these inferences drawn from it will also be accurate. In addition, Suskie (1996) reports that reliability and validity are enhanced when the researcher takes certain precautionary steps. Accordingly, it is important to have people with diverse backgrounds and viewpoints to view the survey before it is administered to determine if: (1) each item is clear and easily understood; (2) they interpret each item in the intended way; (3) the items have an intuitive relationship to the topic and objectives of the study; and (4) the intent behind each item is clear to colleagues knowledgeable about the subject [48].

The testing stage of the SDLC was used to test the overall ERP application and to detect whether there were any bugs. The questionnaire developed included48 key points as items clustered under seven factors. In addition, there is another set of questionnaire which was used to test the quality or failure of ERP applications.

### *6.3. The Sample Design*

According to the statistics of BOI, there are 66 companies registered for BOI software development projects [49]. After applying the justifications mentioned under the study setting, only 48 companies fulfilled the conditions and are capable of developing ERP applications for local and foreign industries. The company name and sensitive details such as job hierarchy were not disclosed due to taxation and policies as per government rules and regulations.

In the study, the representative samples were selected using stratified sampling technique to select the employees (groups/strata), followed by applying the random sampling approach to distribute the questionnaires. The stratified random sampling is an appropriate methodology to make the samples proportionate so that meaningful comparisons between the sub-groups in the population are possible.

Accordingly, everyone in the sampling frame is divided into ‘strata’ (groups or categories). There are four strata as follows:

1. Employees engaged in Business Requirement Analysis (BA)

2. Employees engaged in System Design (SD)
3. Employees engaged in System Implementation (SI)
4. Employees engaged in Quality Assurance for Software Testing(ST)

Within above stratum, a simple random sample is selected to the Quality Assurance for Software Testing (ST) group. Since the employees engaged in the same job role, possessing the same academic qualifications and experience in the same field within the strata (group), the random sampling method was used to distribute the questionnaires to within the above employee group.

To ensure that a sample of 400 from a group of 7745 employees in the four strata (groups) that Software Testing employees in same proportions as in the population (i.e. the group of 7745), the researcher has identified the following sample size who engaged in Quality Assurance for Software Testing(ST) for this research.

No. of ST in sample = (400 / 7745) x 1413 = 73

On the basis of this calculation, the researcher has determined the number of respondents for software testers and for this research 73 sample size has been identified as software testers.

## 7. Methods of Analysis

Data analysis generally begins with the calculation of a number of descriptive statistics such as the mean, median, standard deviation scores and the like. The aim at this stage is to describe the general distributional properties of the data, to identify any unusual observations (outliers) or any unusual patterns of observations that may cause problems for later analyses carried out on the data [50]. Taking the cue from Kristopher (2010), severalstatistical methods were used to formulate the output of data analysis using Statistical Package for the Social Sciences (SPSS) and Sobel test as described in the following sub-sections.

## 8. Hypotheses Testing for System Testing

The researcher has tested the hypotheses of H4: There is a negative relationship between System Testing and ERP Failure. It signifies that if the System Testing does not do a proper job it could be a case of failure for the ERP application. If they do a better job according to the factors that the researcher has identified as per the literature they might be able to achieve success of developing ERP application by doing a proper System Testing.

### *8.1. Overall Descriptive Analysis for Mean in each Factor in System Testing*

The overall Descriptive Analysis for mean in each factor is given as follows as shown below table 1;

Table -1: Overall Descriptive Analysis for mean in each factor in System Testing

**Descriptive Statistics**

| | Mean | Std. Deviation | N |
|---|---|---|---|
| TOTALST | 3.2566 | .26071 | 73 |
| RD | 3.4188 | .56960 | 73 |
| RTP | 3.3973 | .71720 | 73 |
| ITP | 3.6438 | .60369 | 73 |
| RDTP | 3.6545 | .35744 | 73 |
| TM | 3.4452 | .57168 | 73 |
| TT | 3.5845 | .43706 | 73 |

Based on the 73 samples of respondents Required Documents (RD), Requirements for Test Procedure (RTP), Instruction for Test Procedures (ITP), Require Document for Test Preparation (RDTP), Testing Methods (TM), Testing Types (TT) are also close to the very important scale. The Requirements for Test Procedure (RTP) has the biggest variance of points (0.71720) and Require Document for Test Preparation (RDTP) has the lowest variance of points (0.35744) as shown in the above table 1.

### *8.2. Descriptive Statistics of System Testing (ST)*

The descriptive statistics of system testing is given bellow table in 2;

Table 2 : Descriptive statistics of system testing

**Descriptives**

| | | | Statistic | Std. Error |
|---|---|---|---|---|
| TOTALST | Mean | | 3.2566 | .03051 |
| | 95% Confidence Interval for Mean | Lower Bound | 3.1957 | |
| | | Upper Bound | 3.3174 | |
| | 5% Trimmed Mean | | 3.2555 | |
| | Median | | 3.1875 | |
| | Variance | | .068 | |
| | Std. Deviation | | .26071 | |
| | Minimum | | 2.73 | |
| | Maximum | | 3.85 | |
| | Range | | 1.13 | |
| | Interquartile Range | | .40 | |
| | Skewness | | .267 | .281 |
| | Kurtosis | | -.611 | .555 |

The mean system Implementation for ERP applications for the 72 samples is 3.2 with the standard deviation of 0.26071. The maximum and minimum ranks are 3.85 and 2.73. The median is 3.2, indicating at least 50% of the developers ranked more than 3.2. The mode value, obtained using the frequency procedure, is 3.2 Thus the most frequent rank among the developers was 3. According to this, the Factors in the system implementation questionnaire for ERP system testing is moderately important.

Since the mean and median values are very close to each other, perhaps the data could be symmetrical. The skewness value is 0.267. If it shows long tailed data to the right it is said to be positive skewed or skewed to the right The Kurtosis value is -611, which is within ±1. Hence, the data can be assumed to be platykurtik.

### *8.3. Correlation Analysis for Overall System Testing for ERP Failure*

The correlation analysis for overall system testing for ERP is as follows according to the table 3;

According to the correlation analysis in the above table the Factors Required Documents (RD), Requirements for Test Procedure (RTP), Require Document for Test Preparation (RDTP), Testing Methods (TM), Testing Types (TT) have correlation of 0.632, 0.383, 0.411, 0.439, and 0.886 respectively.

They have almost a linear association with the dependent variable ERP Failure (FA) in respect to system testing.

Table -3: correlation statistics of system testing

| Correlations | | TOTALST | RD | RTP | ITP | RDTP | TM | TL | TT |
|---|---|---|---|---|---|---|---|---|---|
| Pearson Correlation | TOTALST | 1.000 | .632 | .383 | .217 | .411 | .439 | .268 | .886 |
| | RD | .632 | 1.000 | .081 | .117 | .094 | .062 | .116 | .443 |
| | RTP | .383 | .081 | 1.000 | -.078 | .240 | .404 | .455 | .129 |
| | ITP | .217 | .117 | -.078 | 1.000 | .028 | -.286 | -.282 | .236 |
| | RDTP | .411 | .094 | .240 | .028 | 1.000 | -.117 | -.086 | .299 |
| | TM | .439 | .062 | .404 | -.286 | -.117 | 1.000 | .599 | .274 |
| | TT | .886 | .443 | .129 | .236 | .299 | .274 | .010 | 1.000 |
| Sig. (1-tailed) | TOTALST | . | .000 | .000 | .032 | .000 | .000 | .011 | .000 |
| | RD | .000 | . | .248 | .163 | .215 | .300 | .164 | .000 |
| | RTP | .000 | .248 | . | .257 | .020 | .000 | .000 | .138 |
| | ITP | .032 | .163 | .257 | . | .407 | .007 | .008 | .022 |
| | RDTP | .000 | .215 | .020 | .407 | . | .163 | .236 | .005 |
| | TM | .000 | .300 | .000 | .007 | .163 | . | .000 | .009 |
| | TT | .000 | .000 | .138 | .022 | .005 | .009 | .465 | . |

The correlation coefficient for Required Documents (RD) and Testing Types (TT), r= 0.443, which is more than 0.3 Thus, there is an association between Required Documents (RD) and Testing Types (TT), in System Testing.

The correlation coefficient for Requirements for Test Procedure (RTP) and Testing Methods (TM), r= 0.404,which is more than 0.3 Thus, there is an association between Test Procedure (RTP) and Testing Methods (TM).

### 8.4. Regression Analysis on Factors of System Testing for ERP Failure

The Regression analysis on Factors of System Testing for ERP Failure is show according to the given table 4 below;

Table -4 : Regression analysis on Factors of System Testing for ERP Failure

**Model Summary**[b]

| Model | R | R Square | Adjusted R Square | Std. Error of the Estimate | Durbin-Watson |
|---|---|---|---|---|---|
| 1 | .995[a] | .990 | .989 | .02780 | 1.688 |

a. Predictors: (Constant), TT, RDTP, ITP, RD, RTP, TM

b. Dependent Variable: TOTALST

According to the above table 4.42, $R^2$ = 99%of the variation in ERP failure is explained by System Testing to reduce failure rate.

Table -5: ANOVA analysis on Factors of System Testing for ERP Failure

**ANOVA**[a]

| Model | | Sum of Squares | df | Mean Square | F | Sig. |
|---|---|---|---|---|---|---|
| 1 | Regression | 4.844 | 7 | .692 | 895.143 | .000[b] |
| | Residual | .050 | 65 | .001 | | |
| | Total | 4.894 | 72 | | | |

a. Dependent Variable: TOTALST

b. Predictors: (Constant), TT, RDTP, ITP, RD, RTP, TM

The p-value from the ANOVA table 5, is less than 0.001 which is 0.000. It means System Testing can be used to predict ERP failure. In the above ANOVA table, the large F-value (895.143), indicated by a small p-value (<0.05) which is 0.000, implies good fit. It means that the least one of the independent variables can explain the outcome.

Table -6: Coefficient analysis on Factors of System Testing for ERP Failure

**Coefficients**[a]

| Model | | Unstandardized Coefficients | | Standardized Coefficients | t | Sig. | 95.0% Confidence Interval for B | | Collinearity Statistics | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | B | Std. Error | Beta | | | Lower Bound | Upper Bound | Tolerance | VIF |
| 1 | (Constant) | .081 | .054 | | 1.497 | .139 | -.027 | .190 | | |
| | RD | .139 | .007 | .304 | 21.107 | .000 | .126 | .153 | .759 | 1.318 |
| | RTP | .038 | .006 | .105 | 6.863 | .000 | .027 | .049 | .669 | 1.495 |
| | ITP | .061 | .006 | .140 | 9.863 | .000 | .048 | .073 | .780 | 1.282 |
| | RDTP | .157 | .011 | .216 | 14.823 | .000 | .136 | .179 | .746 | 1.340 |
| | TM | .101 | .008 | .222 | 11.898 | .000 | .084 | .118 | .455 | 2.196 |
| | TT | .345 | .010 | .578 | 33.952 | .000 | .325 | .365 | .544 | 1.838 |

a. Dependent Variable: TOTALST

The correlation coefficient for items in each factors are as shown in the above table and Required Documents (RD), Requirements for Test Procedure (RTP), Instruction for Test Procedures (ITP), Require Document for Test Preparation (RDTP), Testing Methods (TM), Testing Types (TT) represent p-values as 0.000 respectively, which are less than 0.05 Thus, there are significant predictors of ERP failures among these items in each factor when considering system testing to reduce failure rate as shown in the table 6.

The Equation: FAST = 0.081 +0.139 (RD) +0.038 (RTP) +0.061 (ITP) +0.157 (RDTP) + 0.101 (TM) + 0.345 (TT)

The Equation: FAST = 0.081 +0.139 (Required Documents) +0.038 (Requirements for Test Procedure) +0.061 (Instruction for Test Procedures) +0.157 (Require Document for Test Preparation) + 0.101 (Testing Methods) + 0.345 (Testing Types)

Thus for every unit increase in Required Documents (RD), System Testing for ERP Failure is expected to drop by 0.139. Every unit increase in Requirements for Test Procedure (RTP), System Testing for ERP Failure is expected to drop by 0.038 and for every unit increase in Instruction for Test Procedures (ITP), System Testing for ERP Failure is expected to drop by 0.061 and also for every unit increase in Require Document for Test Preparation (RDTP), System Testing for ERP Failure is expected to drop by 0.157. Furthermore, every unit increase in Testing Methods (TM), System Testing for ERP Failure is expected to drop by 0.055 and also for every unit increase in Testing Types (TT), System Testing for ERP Failure is expected to drop by 0.345.

The 95% Confidence Interval (CI) for Required Documents (RD) is [0.126, 0.153], Where the value of 0 does not fall within the interval, again indicating Required Documents (RD) is a significant predictor.

The 95% Confidence Interval (CI) for Requirements for Test Procedure (RTP) is [0.027, 0.049]. Where the value of 0 does not fall within the interval, again indicating Requirements for Test Procedure (RTP) is a significant predictor.

The 95% Confidence Interval (CI) for Instruction for Test Procedures (ITP) is [0.048, 0.073]. Where the value of 0 does not fall within the interval, again indicating Instruction for Test Procedures (ITP) is a significant predictor.

The 95% Confidence Interval (CI) for Require Document for Test Preparation (RDTP) is [0.136, 0.179]. Where the value of 0 does not fall within the interval, again indicating Require Document for Test Preparation (RDTP) is a significant predictor.

The 95% Confidence Interval (CI) for Testing Methods (TM) is [0.084, 0.118]. Where the value of 0 does not fall within the interval, again indicating Study of Testing Methods (TM) is a significant predictor.

The 95% Confidence Interval (CI) for Testing Types (TT) is [0.325, 0.365]. Where the value of 0 does not fall within the interval, again indicating Testing Types (TT) is a significant predictor.

### *8.5. Mulicollinearity for System Testing*

Required Documents (RD), Requirements for Test Procedure (RTP), Instruction for Test Procedures (ITP), Require Document for Test Preparation (RDTP), Testing Methods (TM), Testing Types (TT) are respectively 1.318, 1.495, 1.282, 1.340, 2.196, 1.890 and 1.838 and as they are below 5 the multicollinearity is not serious. Hence there is no problem of multicollinearity.

## 9. The Mediating effect of Testing Methods between System Testing and ERP Failure (H1)

The illustration of mediating effect of testing methods between system testing and ERP failure is given bellow (figure 1);

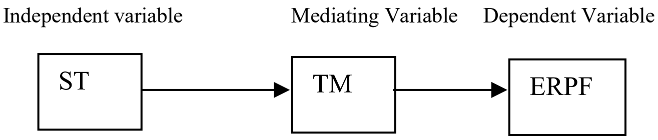

Figure -1: Mediating effect of Testing Methods between System Testing and ERP Failure

The Sobel test is the most appropriate test to measure the mediating effect of a variable on the relationship between the independent variable and dependent variable. According to the result given by the Sobel statistical, results of the Sobel test is 7.55930821, significant p-value is 0.000 which is less than 0.05. It implies that the Testing Methods (TM) is a significant mediator between System Testing (ST) and ERP Failure (ERPF)as shown in the below figure 2.

– a=.962sa =.234

– b=.007sb =.046

Standard error of ab is approximately square root of$b^2s_a^2 + a^2s_b^2$

(Sobel test)–ab $=\sqrt{(.962^2 * .046^2 + .007^2 * 0.234^2)} = .04421$

P<0.001

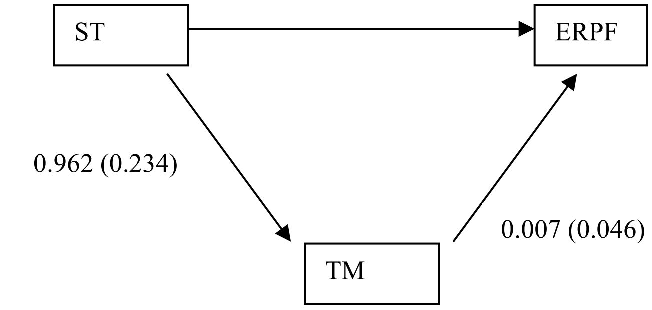

Figure -2: Mediating coefficient values for Testing Methods between System Testing and ERP Failure

Thus, Testing Methods (TM) mediate the relationship. Hence H1 can be accepted.

## 10. Full Path Analysis

The full path analysis is as follows for the conceptual framework.

FA=FABA+FASD+FASI+FAST+FATM+FABT+FAPT+FAPL +FASDM

FA= -0.121 +0.152 (CSI)+0.179(SES) -0.168(INBP)+0.144(SSS)-0.082(DNC) +0.142 (CI)+0.159 (IC)+0.000 +0.097 (ADA) +0.111(SUD) + 0.147(BUD) + 0.129(DD) + 0.101(DDBS) + 0.208 (UID) + 0.081 (CA)+0.000 +0.146 (ALC) +0.208 (BESI) +0.146 (POPE) +0.083 (VD) + 0.083 (MO) + 0.063 (SA) + 0.083 (CF)+ 0.188 (CS)+0.081 +0.139 (RD) +0.038 (RTP) +0.061 (ITP) +0.157 (RDTP) + 0.101 (TM) + 0.345 (TT) + 0.442(FATM) + 0.126(FABT) + 0.089(FAPT) + 0.000(FAPL) + 0.067(FASDM)

## 11. Conclusion

The p-value from the ANOVA table is less than 0.001 which is 0.000. It means System Testing can be used to predict ERP failure. In the above ANOVA table, the large F-value (895.143), indicated by a small p-value (<0.05) which is 0.000, implies good fit. It means that the least one of the independent variables can explain the outcome.

The correlation coefficient for items in each factors are as shown in the above table and Required Documents (RD), Requirements for Test Procedure (RTP), Instruction for Test Procedures (ITP), Require Document for Test Preparation (RDTP), Testing Methods (TM), Testing Types (TT) represent p-values as 0.000 respectively, which are less than 0.05 Thus, there are significant predictors of ERP failures among these items in each factor when considering system testing to reduce failure rate.

Thus for every unit increase in Required Documents (RD), System Testing for ERP Failure is expected to drop by 0.139. Every unit increase in Requirements for Test Procedure (RTP), System Testing for ERP Failure is expected to drop by 0.038 and for every unit increase in Instruction for Test Procedures (ITP), System Testing for ERP Failure is expected to drop by 0.061 and also for every unit increase in Require Document for Test

Preparation (RDTP), System Testing for ERP Failure is expected to drop by 0.157. Furthermore, every unit increase in Testing Methods (TM), System Testing for ERP Failure is expected to drop by 0.055 and also for every unit increase in Testing Types (TT), and System Testing for ERP Failure is expected to drop by 0.345.

### *11.1. 95% Confidence Interval (CI) for System Testing*

According to the regression Analysis the coefficient table says that 95% CI where the value 0 does not fall within the interval indicates it as a significant predictor. The following are the significant predictors that support the H1 hypotheses as shown in the below table 7.

Table -1 : 95% Confidence Interval (CI) for System Testing

| | Cluster name | 95% CI Values | Significant or Not |
|---|---|---|---|
| 01 | Required Documents (RD) | [0.126, 0.153] | Significant predictor |
| 02 | Requirements for Test Procedure (RTP) | [0.027, 0.049] | Significant predictor |
| 03 | Instruction for Test Procedures (ITP) | [0.048, 0.073] | Significant predictor |
| 04 | Require Document for Test Preparation (RDTP) | [0.136, 0.179] | Significant predictor. |
| 05 | Testing Methods (TM) | [0.084, 0.118] | Significant predictor |
| 06 | Testing Types (TT) | [0.325, 0.365] | Significant predictor |

All seven clusters are significant and therefore, there is a negative relationship between System Implementation and ERP Failure.

H1 There is a negative relationship between System Testing and ERP Failure -Accepted

H2 Many Testing Methods do mediates the positive relationship between System Testing and ERP Failure -Accepted

In accordance with the above discussion the above summary gives a clear picture. Based on the hypotheses the researcher has tested in order to reduce the ERP failure rate under the software testing phase. The findings in this study considered the H1 and H2 acceptable and aligned with software testing process in order to reduce the ERP failure rate.

## Conflict of Interest

The authors declare no conflict of interest.

## Acknowledgment

I would like to express my sincere thanks and heartfelt gratitude to my supervisor Prof. (Dr.) Siong Choy Chong for his invaluable advice and guidance throughout the course of my work. It has been a great privilege to have worked with such an outstanding person, and his ever willing help and cooperation contributed towards the smooth progress of my work. I also extend my gratitude to Prof. (Dr.) Ali Katibi, Dean, Faculty of Graduate Studies, University of Management and Science, Malaysia for his advice and guidance throughout the research, as without his invaluable support and assistance given to me in continuing this study, his cooperation and tolerance, my endeavour would not have become a reality.

## References

[1] AL-Hossan A., AL-Mudimigh S., & Abdullah, “Practical Guidelines for Sucessful ERP Testing, **27:** 11 – 18 (2011).

[2] Galin, D.Software Quality Assurance. Harlow: Pearson Education Limited (2004).

[3] Lewis, W. E. Software Testing and Continuous Quality Improvement. New York: Auerbeach Publications, (2005).

[4] Srinivasan Desikan. Software Testing. Chennai: Dorling Kindersley Pvt. Ltd (2011).

[5] Chris P. Learn to Program, (2nd ed.). Pragmatic Bookshelf (2009).

[6] Jae H. Y. ,Jeong S.K.& Hong S.P. Test Data Combination Strategy for Effective Test Suite Generation. IT Convergence and Security (ICITCS), 2013 International Conference, 1-4 (2013).

[7] Andrews J., Briand L., Labiche Y. Using Mutation Analysis for Assessing and Comparing Testing Coverage Criteria. Software Engineering, IEEE Transactions, **32**: 608 – 624 (2006).

[8] Tai K. Condition-based software testing strategies. Computer Software and Applications Conference, 1990. COMPSAC 90, 564 – 569 (1990).

[9] Gizopoulos D., Paschalis A. Effective software-based self-test strategies for on-line periodic testing of embedded processors. Computer-Aided Design of Integrated Circuits and Systems, **24**: 88 – 99 (2004).

[10] Basili V., Selby R., Comparing the Effectiveness of Software Testing Strategies. Software Engineering, **13**: 1278 – 1296 (2006).

[11] Junpeng L, Beihang U., Kai-yuan C., On the Asymptotic Behavior of Adaptive Testing Strategy for Software Reliability Assessment. Software Engineering, **40**: 396 – 412 (2014).

[12] Yachin D. SAP ERP Life–CycleManagement Overcoming the Downside of Upgrading. IDC Information and Data, (2009).

[13] Fraser G., Arcuri A. The Seed is Strong: Seeding Strategies in Search-Based Software Testing. Software Testing, **5**: 121 – 130 (2012).

[14] Howden W. Reliability of the Path Analysis Testing Strategy. Software Engineering, **2**(3): 208 – 215 (2006).

[15] Chaim M., Maldonado J., Jino M. A debugging strategy based on requirements of testing. Software Maintenance and Reengineering, **17**, 160 – 169 (2003).

[16] Prather, Ronald E., Myers J. The Path Prefix Software Testing Strategy. Software Engineering, **13**: 761 – 766 (2006).

[17] William E., Lewis E., Gunasekaran W. Software Testing and Continuous Quality Improvement, **2**: (2005).

[18] Cem Kaner C., Jack falk J. , Hung H. Q.N. Testing Computer Software. New Yor: John Wiley & Son,2nd ed., k, USA. (1999).

[19] Robert, R., Futrell, T., Donald, F., Shafer, F., & Linda L. Quality Software Project Management (2002).

[20] Andrei A., Daniel I.V. Automating Abstract Syntax Tree construction for. 14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (2012)

[21] Demilli R., Offutt A., Constraint -based automatic test data generation. Software Engineering, **17:** 900-910 (2002).

[22] Kuhn D., Wallace D, Gallo A. Software fault interactions and implications for software testing. Software Engineering, **30**(6): 418 – 421 (2013).

[23] Wallace D, Kuhn D. & Gallo A. Software fault interactions and implications for software testing. Software Engineering, **30**(6): 418 - 421. (2013).

[24] Chan F., Chen T., Mak I. Proportional sampling strategy: guidelines for software testing practitioners. Information and Software Technology, **28**: 775–782 (1996).

[25] Satyanarayana, C., Niranjan, V., and Vidyadara, C. Method, process and technique for testing erp solutions. U.S. Patent Application, (2012).

[26] Dennis, D., Shahin, S., & Zadeh, J. Software Testing and Quality Assurance, **4**: 4 (2008).

[27] Software Testing Techniques, Hariprasa, P.,Hariprasa, P, (2015).

[28] Laurie Williams, Testing Overview (2006).

[29] Jovanovi. Software Testing Methods and Techniques. The IPSI BgD Transactions on Internet Research, **30** (2014).

[30] Wikipedia. Software Testing. Retrieved from wikipedia.org: http://www.wikipedia.org/wiki/software_testing (2014).

[31] Anthony A., John J. Acceptance Testing. Encyclopedia of Software Engineering, 1st edition (2002).

[32] Elfriede D., Jeff R., John P. Automated Software testing, (2008).

[33] Ahimbisibwe, A., Cavana, R. Y., & Daellenbach, U. A contingency fit model of critical success factors for software development projects: A comparison of agile and traditional plan-based methodologies. Journal of Enterprise Information Management, **28**(1): 7-33 (2015).

[34] Thummalapenta S., Sinha S., Mukherjee D., Chandra S. Automating Test Automation, (2011).

[35] Haugset B., Hanssen G. Automated Acceptance Testing. A Literature Review and an Industrial Case Study, 27-38 (2008).

[36] Rankin C. The Software Testing Automation Framework. IBM Systems Journal, **41**: 129-139 (2002).

[37] Karhu K., Repo T., Taipale O., Smolander K. Empirical Observations on Software Testing Automation. Software Testing Verification and Validation, **9**: 201-209 (2009).

[38] Bansal A., Muli M., Patil K. Taming complexity while gaining efficiency: Requirements for the next generation of test automation tools. AUTOTESTCON, 1 – 6 (2013).

[39] Borjesson E., Feldt R. Automated System Testing Using Visual GUI Testing Tools: A Comparative Study in Industry. Software Testing, **5**: 350 – 359 (2012).

[40] Guide, G. C. Best practices for developing and managing capital Program Costs **GAO-09-22**: (2009).

[41] Jain, A., Jain, M., & Dhankar, S. A Comparison of RANOREX and QTP Automated Testing Tools and their impact on Software Testing. IJEMS, **1**(1): 8-12 (2014).

[42] Creswell, J. "Research Design: Qualitative and Quantitative Approaches. Sage Publications, (1994)

[43] Sekaran & Bougie. Research Methods for Business A Skill Building Approach. (5th, Ed.) John Wiley & Sons Ltd, (2009)

[44] Kristopher J. P. Calculation for the Sobel Test. Retrieved June 10, 2015, from http://quantpsy.org/sobel/sobel.html (2010).

[45] Gerrard, P. Test Methods and Tools for ERP Implementations. PO Box 347, Maidenhead, Berkshire, SL6 2GU, UK: Gerrard Consulting, (2015).

[46] Central Bank, O. S. Annual Report 2014. Central Bank of Sri Lanka, (2014).

[47] Suskie,. A. "Questionnaire Survey Research: What Works". Assn for Institutional Research, (1996).

[48] Board of Investment. BOI. Retrieved from http://www.investsrilanka.com/: http://www.investsrilanka.com, (2012).

[49] Colombo Stock Exchange. CSE. Retrieved from /www.cse.lk: https://www.cse.lk (2012).

[50] Karuthan. Statistical Analysis using SPSS. Lecture Notes. Management and Science University. (2014).