Software Failures and Risk Management Practices in Nigeria, Kenya and Zimbabwe: A Survey

Esiefarienrhe M. Bukohwo
Dept. of Maths/Stat/Computer Science
University of Agriculture, Makurdi.

Greg Wajiga
Department of Mathematics and Computer Science
Federal University of Technology, Yola

M. O. Egwurube
Department of Mathematics and Computer Science
Federal University of Technology, Yola

***Abstract*—** Software Project failure particularly in Nigeria, Kenya and Zimbabwe is partly due to lack of adequate knowledge on the part of software developers on software risk management methodologies and the roles they play in software engineering. Over the years and with increase in software project failures, no attention has been paid to the reasons behind project failures. This paper seeks to investigate the degree of awareness of risk principles by software developers and their application of these principles in software development. The instrument used was research questionnaires. Four hundred (400) questionnaires were analyzed using percentages and chi-square and the results shows that software failure is not as a result of ignorance of risk management principles but the non adherence and usage of the principles involved in developing risk mitigated software.

Keywords*: Risk; management; mitigation; Chi-square; Information Technology*

I. INTRODUCTION

As more products begin to include software, understanding the behavior of software has become a more widely relevant issue. Software has become paramount to our everyday activities. Why is software so important? Software flies our airplanes [1], controls our automated teller machines (ATM) [2], and even controls our car engines [3]. The implications are that software drives the economy. The extent of the application of software to the economy varies with countries. The efficiency of these applications also varies. A fact that is easily established is that developing countries are more eager to information technology applications without recourse to risk implication. This has becomes a fundamental problem if not addressed, would lead to serious consequences that may even endanger lives and properties.

As developing countries are beginning to embrace information technology in their daily operations, concurrently, internal software development activities are increasing. Software developments courses at the various institutions of higher learning are on the increase producing personnel who are vested with project development and management skills. How well equipped are these personnel and how well are they familiar with software risk management as been a major concern to software users over the decade. Mostly surprisingly, the rate of software failures has been on the increase to an almost alarming rate.

Countries	Number of Questionnaires Administered	Number of Questionnaires completely filled and returned	% of questionnaires returned
Nigeria	150	145	97%
Zimbabwe	150	120	80%
Kenya	150	135	90%
Total	450	400	89%

In a survey [4] of 600 firms indicated that 35% of them had at least one runaway software project. A post-mortem of various software disasters have indicated that their problems would have been avoided or strongly reduced if there had been explicit early concern with identifying and resolving their high-risk elements. Frequently, these projects were swept along by a tide of optimistic enthusiasm during their early phases that cause them to miss some clear signals of high risk issues that proved to be their downfall later [5].

Most often, the focus is on the risk symptoms rather than on the root cause of the problem – lack of adequate knowledge of risk management and principles in software development. In managing software risk, all phases, people and environment are potential sources of risk since they involve technology, hardware, people, cost and schedule. This view is expressed succinctly by taking a holistic approach to software risk management and thus approaching risk management from the system perspective [6].

The purpose of this paper is to conduct a research survey among software developers and managers who are in charge of software development in other to verify the rate of software failures and to ascertain if those involved are aware of risk management and do incorporate the principles of effective risk management in their software development.

II. RESEARCH METHODOLOGY

The research instrument used was research hypothesis. Questionnaires were administered and the results collected from respondents were analysed using statistical means such as percentages and chi-square.

III. RESEARCH HYPOTHESES

The following two hypotheses were used for this research.

Hypotheses 1

Ho: Software developers and managers are familiar with software risk management principles and techniques.

H_1: Software developers and mangers are not familiar with software risk management principles and techniques.

Hypotheses 2

Ho: Software project failure is not a common occurrence in developing countries.

H_1: Software project failure is a common occurrence in developing countries.

IV. RESEARCH DESIGN

A total of four hundred and fifty (450) questionnaires were administered in three countries out of which four hundred (400) were returned. This represents 89% response as shown on Table 1:

Table 1: Numbers of Questionnaires used for Analysis
Source: (Field Study, 2010)

V. METHOD OF DATA ANALYSIS

The data obtained from the questionnaires were analyzed using percentages, hypothesis and Chi-square.

Chi-square Analysis:

The chi-square test is a measure of relationship, association or independence. It is a nonparametric test. The probability density curve of a chi-square distribution is asymmetric curve stretching over the positive side of the line and having a long right tail. The form of the curve depends on the value of the degrees of freedom.

Types of Chi-square Analysis

Chi-square Test for Association is a (non-parametric, therefore can be used for nominal data) test of statistical significance widely used bivariate tabular association analysis. Typically, the hypothesis is whether or not two different populations are different enough in some characteristic or aspect of their behavior based on two random samples. This test procedure is also known as the Pearson chi-square test.

Chi-square Goodness-of-fit Test is used to test if an observed distribution conforms to any particular distribution. Calculation of this goodness of fit test is by comparison of observed data with data expected based on the particular distribution.

The chi-square is used to test hypotheses about the distribution of observations into categories. The null hypothesis (Ho) is that the *observed* frequencies are the same (except for chance variation) as the *expected* frequencies. If the frequencies you observe are different from expected frequencies, the value of χ^2 goes up. If the observed and expected frequencies are exactly the same, $\chi^2 = 0$. You test whether a given χ^2 is statistically significant by testing it against a table of chi-square distributions, according to the number of degrees of freedom for your sample, which is the: number of columns – 1 x number of rows - 1. The chi-square assumes that you have at least 5 observations per category (cell). The formula is:

$$X^2 = \sum \frac{(FO - FE)^2}{FE} \qquad df = (c-1)x(r-1)$$

With degrees of freedom equal to the number of categories minus 1, and where

FO = Observed frequency
FE = Expected frequency

Analysis of Hypotheses

Hypotheses 1:

Table 2: As software Developer, are you aware of the term 'Software Development Risk"

Countries	Responses		
	Yes	No	Row Total
Nigeria	105 (112)	40 (33)	145
Zimbabwe	97 (93)	23 (27)	120
Kenya	108 (105)	27 (30)	135
Column Total	310	90	400

Source: (Field Study, 2010)

While 72% of respondents in Nigeria agreed that they are aware of the existence of software development risk, only about 28% are not aware of the term software

development risk. In Zimbabwe and Kenya, 80% of the respondents are aware of the term software development risk and about 20% of system developers are not aware.
Critical Value: the degree of freedom are (2-1)(3-1) = 2 and ∞ =0.05.
Decision Rule: Reject Ho if χ^2cal > χ^2tab i.e. 3.07 < 5.99
Decision: The decision is to accept the null hypothesis Ho and reject the alternate hypothesis H_1 because the calculated value of p is less than the table value of p at critical values of 5%.

Hypothesis 2
Ho: Software project failure is not a common occurrence in developing countries.
H_1: Software project failure is a common occurrence in developing countries.

Questions 2 and 3 of the questionnaire in Appendix 1 will be used to analyze hypothesis 2.
Table 3 and 4 provides the data for the analysis.

Table 3: Have your organization suffered any project failure of lately?

countries	Responses		
	Yes	No	Row Total
Nigeria	131 (121)	14 (24)	145
Zimbabwe	105 (100)	15 (20)	120
Kenya	97 (112)	38 (23)	135
Column Total	333	67	400

Source: (Field Study, 2010)

χ^2 =17.98
Critical Value: the degree of freedom are (2-1)(3-1) = 2 and ∞ =0.05.
Decision Rule: Reject Ho if χ^2cal > χ^2tab i.e. 17.98 > 5.99
Decision: The decision is to reject the null hypothesis Ho and accept the alternate hypothesis H_1 because the calculated value of p is greater than the table value of p at critical values of 5%.

.

Table 4: How often does your organization suffer project failure?

Countries	Responses				
	Very often	Often	Rare	Never	Row Total
Nigeria	40 (34.44)	63 (77.21)	22 (19.21)	20 (14.14)	145
Zimbabwe	30 (28.5)	65 (63.9)	16 (15.9)	9 (11.7)	120
Kenya	25 (32.06)	85 (71.89)	15 (17.89)	10 (13.16)	135
Col. Total	95	213	53	39	400

Source: (Field Study, 2010)

χ^2 = 18.86
Critical Value: the degree of freedom are (4-1)(3-1) = 6 and ∞ =0.05.
Decision Rule: Reject Ho if χ^2cal > χ^2tab i.e. 18.86 > 12.59
Decision: combining the results obtained from table 3 and 4, it is evidence that the null hypothesis ho should be rejected and the alternate hypothesis h_1 should be accepted as the calculated value of p is greater than the table value of p at critical values of 5%.
Conclusion: Table 3 and 4 shows that software project failures in developing countries are rampant. Most organisation always witness one project failure or the other. This is not a healthy development and must be addressed if there is to be sustained development.

VI. RESULTS

The above survey provides evidences to support the fact the software developers and managers are quite familiar with software risk management principles and techniques as they are quite familiar with software development risk. Although being aware of software risk management does not necessary means application of the principles. It is therefore evident that software failures results from the non application of risk management principles to software development.

VII. CONCLUSION

The researchers have shown using empirical studies the need for software developers to inculcate risk management principles in the design of software. Software project fails because developers do not follow the principles enshrined in the Software Development Life Cycle (SDLC). Within the SDLC, there are risks mitigating strategies. It is therefore our believe that a research of this type will provide management and developers the data needed to encourage good software design methodologies.

VIII. FUTURE RESEARCH

The researchers will in the future investigate the most prevalent risk in software development in these countries and most likely development models to track and mitigate such risk with the overall objective of minimizing software failures in these countries.

REFERENCES

[1] Tomayko, J. E. (1991); The Airplane as Computer Peripheral. American Heritage of Inventions & Technology, 7, 3 Winter.
[2] Kanter, J. Schiffman, S. and Horn, S. (1990); Let the customer do it; from grocery robots to photo kiosks, computerized selfservice. *Computerworld*, 24: 35.

[3] Woolnough, R. (1990); TI drives at Euro autos; makes 'PACT' to win microcontroller business. Electronic Engineering Times August.
[4] Rothfeder, J. (1988); Its late, costly, and Incompetent – But Try firing a Computer System. Business Week, Nov 7, pp 164-165.
[5] Boehm, B. (1981): "Software Engineering *Economics*", Prentice-Hall, Englewood Cliffs, N.J.
[6] SEI (1996); Software Engineering Institute, Software Risk Management, Technical Report CMU/SEI-96-TR-012

APPENDIX 1:

The Federal University of Technology, Yola
Department of Mathematics and Computer Science
Yola, Adamawa State

QUESTIONNAIRE

The purpose of this questionnaire is to obtain information that will assist in carrying out a research work on the above topic. Any information given will be treated with utmost secrecy and will be used only for research and publication purpose.
INSTRUCTION: Please tick where appropriate.

SECTION A: PERSONAL DATA

1. Name

(optional):_____

2. Sex: Male ☐ Female ☐

3. Rank: i. Management Staff ☐

ii. Software Developer ☐

SECTION B: COMPULSORY

1. Have you heard of the term software development risk?
Yes No
2. Have your organisation being faced with any risk of lately?
Yes (State the risk – you can refer to Questions 7 for list of risk)
No
3. Has your organisation computerised any of its operation s?
Yes No
4. If yes, how will you rate the success of the project
Very successful success not successful failure

5. Who, do you think, is to be blamed for the failures/non-success? Rank in order of importance

Management Development Team Client (users)

6. How often does your organisation suffer project failure?
Very often Not often Rare Always