American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-8, Issue-7, pp-186-192
www.ajer.org
Research Paper Open Access

Impact of Risk Management on Software Projects in Nigeria Using Linear Programming

Adimora Kyrian Chinemeze and B. C. E. Mbam
Department of Computer Engineering, Michael Okpara University of Agriculture Umudike, P.M.B. 7276, Umuahia, Nigeria
Corresponding Author: Adimora Kyrian Chinemeze

ABSTRACT: *The global research on software indicates that the success rate of software projects worldwide is currently very low, and has been low for several decades. However, the application of risk management has improved the success rate of software projects in the developed world. This research study is conducted to know if the success rate of software projects in Nigerian is also low, and whether risk management might also improve these success rates. The research results indicate that the average success rate in Nigerian is extremely low and often experience the same risks as the institutions in the developed world. We also observed from our research results that where risk management is applied, software projects produce better results than software projects with no risk management. Quantitative research methodology was deployed for data collections and all simulations were performed in Matlab and Microsoft Excel.*

KEYWORDS: *Risk Management, Developed World, Software Projects, Success Rate, Nigeria*

Date of Submission: 07-07-2019 Date of Acceptance: 25-07-2019

I. INTRODUCTION

In the last few decades especially at the end of 20th century and in the beginning of 21st century, people have shown a great interest in software applications and automation of different human activities, which has made software practically indispensable in our daily activities. In this present generation, software has solved numerous problems through the use of artificial intelligence, neural networks, and expert systems to simplify the most difficult tasks of human beings and delivering result in time. However, the complexity of software development has significantly increased over the years due to its demand in almost every aspect of life. Surprisingly many software industries are faced with regular software project failures thereby decreasing projects success rate. Avizienis et al. (2004) identified two levels of project failures according to its degree of severity. (a) Partial failure and (b) Complete failure**. Partial failure:** is a kind of project failure that resulted when the system has been delivered to the user, but some of the users' expectations or success criteria are not satisfied. The criteria could be either the budget exceeding the original cost in the plan, or schedule overrun. On the other hand, **complete failure** exists when the software product (the system) cannot be delivered to the customer. This may occur as a result of insufficient fund before the delivery phase (a budget failure) or delaying the planned delivery so much that the system will no longer add value to the customer's business (a schedule failure). A complete project failure results in project termination. Risk management is viewed as series of steps taken to identify, address, and eliminate software risk items before they manifest as threats to successful software operation (Boehm (1989) cited in William (2004)). Schwalbe (2003) cited in Abdelrafe (2014) defined risk management as the process of identifying, analyzing and controlling risk throughout the life of a project, to meet the project objectives. This research paper is aimed at measuring the effectiveness of risk management within the context of software projects in Nigeria. Dikmen et al (2004) defined risk management as the objective functions to represent the expected outcomes of a project, measuring the probability of achieving objectives by generating different risk occurrence scenarios and development of risk response strategies to ensure meeting the preset objectives. Visser (2013) described software risk management as an ingredient to project success rate. The Standish Group (2014) reported the overall, success, challenged and failed projects with the depicted values in Table 1.

Table 1. Projects Classification

YEAR	1994	2009	2010	2011	2012	2013
SUCCESSFUL	31%	36%	38%	37%	41%	36%
CHALLENGED	53%	44%	40%	46%	40%	48%
FAILED	16%	20%	20%	17%	19%	16%

II. MATERIALS AND METHOD

2.1 Materials

a) Matlab 2013 and Microsoft Excel 2013 for results simulations and analysis
b) Questionnaires for primary data collections
c) Literature review of scholars in risk management

2.2 Method

Linear Programming (LP) approach was used in the design of the mathematical model that evaluates the effect of risk management on software projects success.

2.3 Questionnaire

A questionnaire with twelve questions was used in the analysis of the respondents' feedback on the aspects of software projects failures against projects success by deploying risk management strategies. The twelve questions were designed to understand the importance of risk management application on software projects. A total of 448 questionnaires were sent to some tertiary institutions in Nigeria that offers ICT related courses such as computer engineering, computer science, information management technology, software engineering and cyber security. However, 256 questionnaires were returned out of 448 distributed, 26 of the returned questionnaires were completed by students who do not study any of the above courses. The rest of the returned questionnaires were filled by students of the above mentioned courses. Figure 3 showed the distribution of the positions held by the respondents with respect to software projects development.

2.4 Research Hypotheses

The following hypotheses were postulated for this paper

2.4.1 Nature of Risks

H0: Software Project Risk (R_{sp}) faced by tertiary institutions in Nigeria are not the same as those in the developed world.
H1: Software Project Risk (R_{sp}) faced by tertiary institutions in Nigeria are the same as those in the developed world.

2.4.2 Impact of Risk Management on Project Success

H0: Risk Management (R_M) has no impact on project success
H1: Risk Management (R_M) increases the probability of project success

Table 2. Respondents' Information

Institutions	Departments	Number of Students	Year	Failed Projects [%]	Successful Projects [%]
MOUAU	Computer Engineering/ Science	223	2013 - 2017	80	20
FUTO	Information Management Technology	180	2013 - 2017	85	15
FUNAAB	Computer Science	108	2013 - 2017	78	22
UNIUYO	Electrical Electronics Engineering	143	2013 -2017	87	13

Here failure rate outweighs that success rate because risk management process was not applied in the design and implementation of the identified percentage projects failures and success shown in Table 1 above.

III. THE CONCEPTUAL FRAMEWORK FOR RESEARCH INVESTIGATION

In Figure 1 below the acronyms B, S, Q and X_p stand for Budget, Schedule, Quality and Convergence of the three variables B, S, & Q. The Software Risk Assessment Decision Framework principle shows that when planned budget and schedule are subtracted from the convergence of the three variables budget, schedule and quality the result is a failed project. Viewing the Figure 1 critically we would see that all the desired three variables must be available for a project success to occur otherwise a challenged project will be experienced when at least one variable is missing.

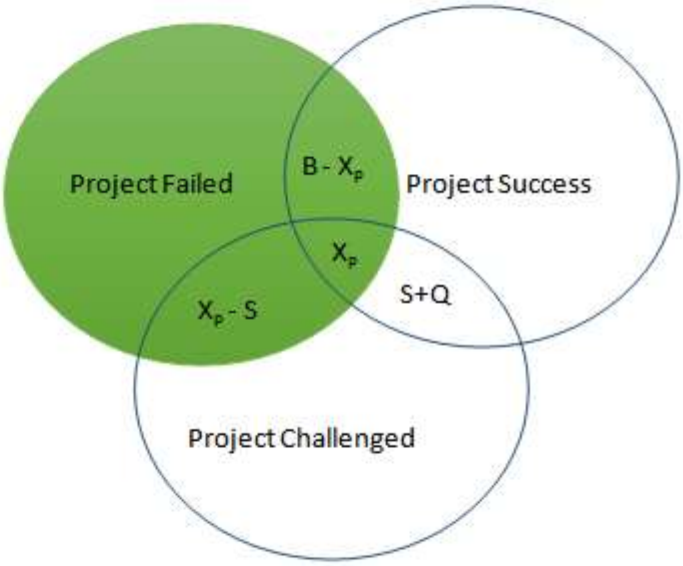

Figure 1. Software Risk Assessment Decision Framework

In this paper, projects were classified into three ways as shown in Figure 1:

- **Project Success:** The project is completed on-time and on-budget, with all the features and functions as initially specified
- **Project Challenged:** The project is completed and operational but over-budget, over- time estimate, and offers fewer features and functions than originally specified.
- **Project Failed:** The project is cancelled at some point during the development cycle.

1.1 Primary Data Collection

If (Software Project is achieved within Budget & Schedule with right Quality)

Print Successful Project

Else If (Software Project is achieved within Schedule at Budget overrun with desired Quality)

Print Challenged Project

Else

Print Failed Project

Figure 2. Algorithm for Software Risk Decision Framework

The principle of the algorithm is that when the three decision variables budget, schedule and quality are achieved as planned, the project is considered to be successful, also when either one, two or three of the decision variables is/are failed to be achieved as planned then the project is seen as challenged / failed project as depicted in Figure 1. Where Xp represents the convergence of the three variables “Budget, Schedule and Quality”. Xp in the software risk assessment decision framework shown in Figure 1, simply implies that the project will be achieved within budget, schedule with the desired quality.
In order to better understand the nature of risk with respect to either project will succeed or fail, we formulated software project risks’ equation at a constant risk factors as follows:

Software Project Risks $(R_{sp}) \propto \frac{1}{\text{Software Project Success } (S_{sp})}$ 1

Software Prject Risks $(R_{sp}) \propto$ Software Project Failure (F_{sp}) 2

The combination of equation (1) and (2) gave birth to the general risk equation stated thus:

Software Project Risk $(R_{sp}) \propto \frac{\text{Software Project Failure } (F_{sp})}{\text{Software Prject Success } (S_{sp})}$ 3

$R_{sp} = \frac{\pi F_{sp}}{S_{sp}}$ 4

Where π is the constant of risk factor and its value is $\frac{22}{7}$

3.1 Risk Resolution with LP

Optimize (Max. or Min.) $Z = \sum_{j=1}^{n} C_j x_j$ 5

Where $Z = R_{sp}$ and $C_j x_j = \frac{\pi F_{sp}}{S_{sp}}$

Therefore, the formulated equation used to analyze the impact of risk management on software projects is stated thus:

$(\text{Min})\ R_{sp} = \sum_{j=1}^{n} \frac{\pi F_{sp}}{S_{sp}} X_j$ 6

Subject to the linear constraints

$\sum_{j=1}^{n} B_{ij} x_j \ (\leq, =, \geq) W_i$ 7

where $i = 1, 2, \dots, m$; and

$X_j \geq 0; j = 1, 2, \dots, n$

B_{ij} is the budget constraints while W_i is workforce constraints. In this paper, both the budget and workforce constraint are defined as follows:

Budget Constraints: The sum of all the costs of the selected critical risk elements cannot exceed the available budget which is B.

$\sum_{j=1}^{n} X_j * \text{Cost}_j \leq B$ 8

Available Workforce constraint: The sum of all the workforce requirements of the selected critical risk elements cannot exceed the total available workforce, which is W.

$\sum_{j=1}^{n} X_j * W_j \leq W$ 9

Therefore, if we plug in the cost incur in working at each stage of the software development we can then have the following results:

$FX_n + RX_{n+1} + DX_{n+2} + CX_{n+3} + TX_{n+4} + IX_{n+5} + MX_{n+6} \leq B$ 10

Where F represent cost of the feasibility study, R represent the cost of both system and user requirements, D represents the cost of system design, C represent the cost of realizing the algorithm in programming codes, T represent cost of both unit and system testing, I represent cost of system integration and M represents cost of system maintenance. Similarly the workforce constraint is determined using equation 1.12. Therefore, if we plug in the available workforce needed in each stage of the software development project we can then have the following results:

$FX_n + RX_{n+1} + DX_{n+2} + CX_{n+3} + TX_{n+4} + IX_{n+5} + MX_{n+6} \leq B$ 11

Where F represent the available workforce in the feasibility study stage, R represent the available workforce in both system and user requirements stage, D represents the available workforce in the system design stage, C represent the available workforce required for turning the design into programming codes, T represent the available workforce in the both unit and system testing stage, I represent the available workforce in the system integration and finally M represents the available workforce in the system maintenance phase.

Therefore the developed linear programming model is mathematically expressed as follows and it is most suited in solving budgetary and workforce projects problems.

$(Min.) R_{sp} = f(X) = \sum_{j=1}^{n} \frac{\pi F_{sp}}{S_{sp}} X_j$ 12

Subject to the following constraints

$F_Q X_1 + F_S X_2 + F_B X_3 \leq FX_n$ 13

$$R_Q X_1 + R_S X_2 + R_B X_3 \leq RX_{n+1} \quad 14$$
$$D_Q X_1 + D_S X_2 + D_B X_3 \leq DX_{n+2} \quad 15$$
$$C_Q X_1 + C_S X_2 + C_B X_3 \leq CX_{n+3} \quad 16$$
$$T_Q X_1 + T_S X_2 + T_B X_3 \leq TX_{n+4} \quad 17$$
$$I_Q X_1 + I_S X_2 + I_B X_3 \leq IX_{n+5} \quad 18$$
$$M_Q X_1 + M_S X_2 + M_B X_3 \leq MX_{n+6} \quad 19$$
$$X_1, X_2, X_3 \geq 0$$

Where Q, S and B in equation 13 through 19 represents the quality, schedule and budget variables that each of the defined constraints must consider in analyzing the cost and workforce details at each phase of software development life cycle. The model optimization result was tested in Matlab.

IV. RESULTS AND DISCUSSION

4.1 Data Analysis from Questionnaires

The respondents' information on Table 2 was analyzed in Figure 3. It can be seen from Figure 3 that the number of failed projects were greater than successful projects because risk management was not applied to the software projects. This showed that the software projects risk faced by students in Nigerian tertiary institutions are the same as those in the developed world. In Figure 3 the meaning of Fp, Sp and Std are described as follows: Fp is failed projects, Sp is successful projects while Std is number of students involved.

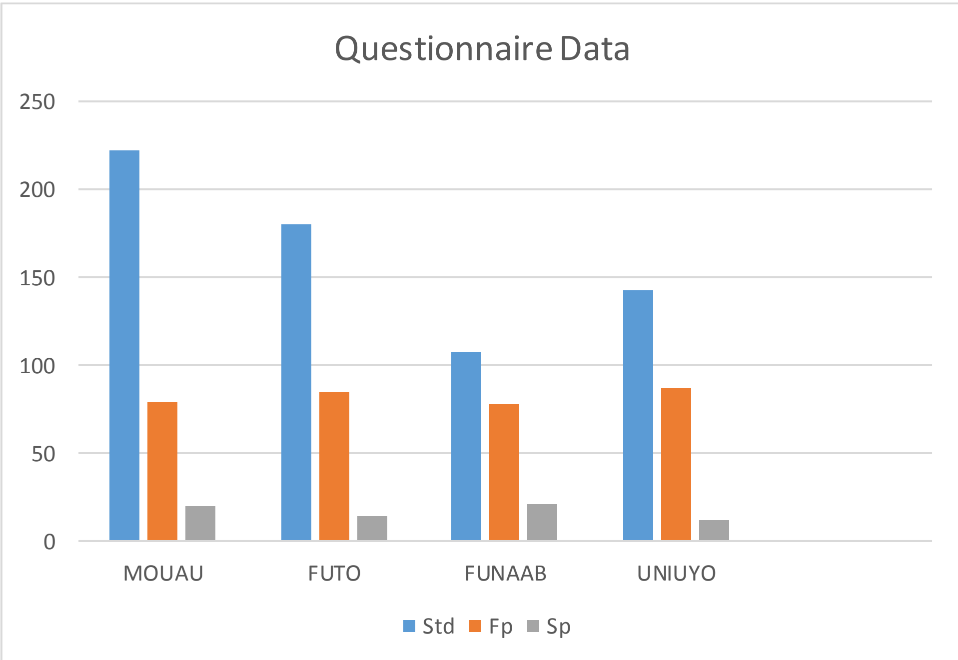

Figure 3. Projects Success Versus Failure with RM not applied

4.1.1 Impact of Risk Management on Software projects

With the introduction of risk management in the software projects of the specified years in this study. The projects' success rate were improved at a very prominent percentage values shown in Table 3 compared with the Standish Group (2014) report in Table 1. Although the percentage values of risk managements' application compared to the Standish group, were not statistically significant for the test of Figure 3. Therefore, the success rate of software projects in Nigeria is therefore roughly the same as in the developed world.

Table 3. Risk Resolution via Risk Management

YEAR	2013	2014	2015	2016	2017	2018
SUCCESSFUL	41%	56%	68%	57%	61%	76%
CHALLENGED	33%	24%	20%	36%	23%	18%
FAILED	16%	20%	18%	17%	19%	16%

4.2 Software Risk Factors

The following three factors are taken to be the most risky in Nigeria with respect to software development.

- Unrealistic schedules and budgets
- Continuing requirements changes
- Less functionality than required

If the actual mean realized from the successful projects of Table 3 is 96.5 percent. A one-sample T-Test was applied to responses that produced the result of Table 3 to test whether a sampled mean of 96.5 percent is statistically significant, being higher than the hypothetical mean of 70 percent. Table 4 below summarizes the result of the T-Test.

Table 4. T-Test Result

Actual Mean	Target Mean	P-value
96.5 %	70%	0.0058

In Table 4 the result of T-Test produced a P-value of 0.0058 and 96.5 percent actual mean which is statistically significant as it is greater than the target of 70 percent. The null hypothesis (H0) can thus be rejected and the research hypothesis (H1) accepted. The results therefore indicates that software projects in Nigeria face the same risks as the software projects in the developed world.

Table 5. Risk Breakdown Structure (RBS)

ZONE	RBS CODE						INSTITUTIONS	STUDENT's ASSESSMENT		RISK VALUE	YEAR
								Total Project	Number Utilized		
E	0	0	0	0	0	0	Michael Okpara	430	8	High	2013
	0	1	-	P	M	B	University of	380	5	High	2014
			7	2	6	7	Agriculture,	520	3	High	2015
							Umudike	618	12	High	2016
							(MOUAU)	545	9	High	2017
							Abia State				
W	0	0	0	0	0	0	Federal	330	6	High	2013
	1	0	-	P	M	B	University	252	4	High	2014
			2	2	4	0	of Agriculture,	290	2	High	2015
							Abeokuta	248	1	High	2016
							(FUNAAB)	305	5	High	2017
							Ogun State				
S	0	0	0	0	0	0	University of	289	10	High	2013
	1	1	-	P	M	B	Uyo (UNIUYO)	301	4	High	2014
			1	0	1	7	Akwa Ibom	369	13	High	2015
							State	422	8	High	2016
								508	6	High	2017
E	0	0	0	0	0	1	Federal	180	0	High	2013
	0	0	-	P	M	B	University of	228	7	High	2014
			1	5	2	6	Technology	254	4	High	2015
							Owerri (FUTO)	215	6	High	2015
							Imo State	280	11	High	2017

V. CONCLUSIONS

The following conclusions can be made in line with this research study

- The success rate of software projects in Nigeria is low, as it is in the rest of the world. The need to investigate the impact of risk management on project success rates can thus be substantiated.
- The application of risk management procedures increases the rate of successfully executing software projects in Nigeria.
- The risks that face software projects in the developed world is generally the same as in Nigeria.

REFRENCES

[1]. Abdelrafe E.R and Roy (2014) "Software Risk Assessment and Evaluation Process (SRAEP) using Model Based Approach", International Conference on Networking and Information Technology, IEEE, pp 171-177.
[2]. Avizienis B.N, Esteves, J. & Rodriguez, N. (2004) Implementing and improving the SEI risk management method in a university software project. IEEE Latin American Transactions, 3, (1), pp. 90-97.
[3]. B. W. Bohem and Williams (2004), "Software Risk Management: Principles and Practices," IEEE Software, vol. 8, pp. 32-41, 1991
[4]. Dikmen, I., Birgonul, M. T. and Arikan, A. E., (2004) "A critical review of risk management support tools", 20th Annual Conference of Association of Researchers in Construction Management (ARCOM), 1-3 September, Vol (1), 1145-1154.
[5]. J.K. Visser and B. de W et al.(2013) "An evaluation of software project management in South Africa." South African Journal of Industrial Engineering May 2013 Vol 24(1), pp 14-28
[6]. Schwalbe (2003) Risk management for software projects, IEEE Software **11** (3) pp. 59-67.
[7]. Standish Group (2009) Standish chaos report. www.projectsmart.co.uk/docs/ chaos-report.pdf [Accessed on November 11, 2010].