Journal of the Nigerian Association of Mathematical Physics
Volume 46 **(May,** *2018 Issue***), pp191 –196**

KNOWLEDGE MANAGEMENT PRACTICES IN NIGERIAN SOFTWARE ORGANISATIONS

[1]Oveh R.O. and [2]Egbokhare F.A.

[1] Department of Mathematics and Computer Science, Western Delta University, Oghara, Delta State. [2] Department of Computer Science, University Of Benin, Benin City.

Abstract

Knowledge management is key to survival in knowledge intensive activities like software development. This research studied some human level activities in the software development process to determine the knowledge management practices in Nigeria software development organizations using quantitative methods. Factor analysis using oblique rotation was first used to summarize and group the variables into patterns that represent the core knowledge management practices. The findings showed that though not formalized, there are four main knowledge management entities from the human level activities in the software process namely: Knowledge Harvesting, Knowledge Creation, Management Commitment and Knowledge Management Culture. A starting point for most organizations with informal and undefined knowledge management structures is knowledge harvesting. We therefore recommend that adequate support be provided by software project managers to harvest and codify software process knowledge for reuse and as an intellectual capital that remains in the organization.

1.0 INTRODUCTION

The increasing demand for computer software in today's information driven economy has prompted software project managers to seek for faster and cost effective development cycles. A typical software development process consists of various phases requiring several interrelated knowledge-intensive activities. Recent research in Software Engineering is centered on defining formal structures for managing software process knowledge [1] because the level of knowledge in a software project is a key determinant of its success [2].Knowledge Management (KM) is the process of creating, capturing, sharing, and storing a company's intellectual capital [3].It provides strategies for creating and sharing knowledge within an organization to improve performance. Software development organizations require knowledge which is initially dispersed, tacit, latent and incomplete in large measures to improve the software process irrespective of the life cycle model adopted .Managing software process knowledge is perceived as a venture that exploits the knowledge stored as intellectual capital in the individuals to benefit the organization. Though majority of this knowledge is tacit, residing in the brains of the employees, it was observed that capturing this knowledge will help software engineers to reduce development cost and improve efficiency [4, 5]. Knowing the current level of knowledge management in an organization will serve as a guideline for institutions to know where to direct further and future knowledge management endeavours [6]. Maturity models are used to identify the developmental level of knowledge management processes based on acceptable criteria [7]. Maturity modeling has been proposed as a way to formally capture organizations KM processes by assessing the extent to which KM is explicitly defined, managed, controlled and effectively utilized [8,9,10].A study in [11] showed that Nigerian software development organizations are at KM maturity Level 1. A major characteristic of organizations at knowledge maturity level 1 organizations is that there is no formally defined knowledge management structures. A starting point for such organizations is to identify core knowledge activities and harvest such knowledge. This paper seeks to identify software process activities that represent core knowledge areas and group them into appropriate knowledge areas.

Corresponding Author**:** Oveh R.O., Email: omo_rich@yahoo.com, Tel: +2347036142579, +2348037180057 (EFA)

2.0 METHODOLOGY

In order to achieve the objectives of this research, quantitative techniques were adopted. Quantitative research methods seek to confirm hypotheses about phenomena [12]. Questionnaire was used to determine the software process activities that produce knowledge. The questionnaire was structured into three (3) parts: demographic Information, knowledge considered critical to software development organizations and the third part contained items relating to knowledge management practices in software development organizations based on a five point Likert scale (Strongly Agree (SA); Agree (A); Undecided (U); Disagree (D) and Strongly Disagree (SD)). To ensure content validity, a copy of the questionnaire was presented to two Professors of computer Science and a pre-test was performed on one software project manager and six (6) programmers. The corrections/suggestions made were effected before the final copies were produced and distributed. A sample population consisting of programmers, systems analysts and software project managers was used. A stratified random sampling method was used to distribute the questionnaires. The researcher personally administered the questionnaires in order to have informal interactions with the respondents and probe further were possible. One hundred and twenty (120) questionnaires were administered across four geographical areas in Nigeria (i.e. Lagos, Abuja, Benin City, and Port Harcourt). One hundred and seven (107) questionnaires were returned and used for the final analysis. Coakes and Ong [13] suggested that reliability analysis should be used to determine the internal consistency of the scales using Cronbach's Alpha. Cronbach alpha is the estimate of reliability. It measures how closely related some set of items are as a group. It checks if multiple Likert scale survey is reliable. A high scale is a good scale reliability. We obtained a value of 0.8 which is a high scale. Hence we went to administer the questionnaires. The formula in [14]is:

$$r_{kk} = \left(\frac{k}{k-1}\right).\left(1 - \frac{{s_i}^2}{{s_x}^2}\right) \quad (1)$$

where:

r_{kk} = estimated Cronbach Alpha coefficient value

k = number of items in the questionnare

${s_i}^2$ = sum of item variances

${s_x}^2$ = factor variances

Kaiser-Meyer-Olikin (KMO) test measures how well suited you data is for factor analysis. it measures the sampling adequacy for variable in the model and for the complete model.KMO test was done to identify whether the data is suitable for factor analysis. The KMO test formula as stated in [15]:

$$\text{KMO} = \frac{\sum_{j=1}^{n} \sum_{i=1}^{n} r_{ij}^2}{\left(\sum_{j=1}^{n} \sum_{i=1}^{n} r_{ij}^2 + \sum_{j=1}^{n} \sum_{i=1}^{n} a_{ij}^2\right)} \quad (2)$$

where:

r_{ij} = Correlation coefficient

a_{ij} = Partial correlation coefficient

KMO values are between 0 and 1, where 0.8 to 1 indicate that the sample is adequate and less than 0.6 indicate that the sample is not adequate. The KMO value we obtained using SPSS was 0.8, which shows that the sample is adequate and suited for factor analysis. Hence we went further to do the factor analysis.

To determine the state of practice of knowledge management in Nigerian Software Development organizations, factor analysis was used to analyze the data. Factor Analysis is a technique used by researchers to discover the number of factors influencing variables and to establish relationship between variables (group variables) [16]. In the 'classical factor analysis' mathematical model, *p* denotes the number of variables (*X1, X2,…,Xp*) and *m* denotes the number of underlying factors (*F1, F2,…,Fm*). *Xj*is the variable represented in latent factors. The model assumes that there are *m* underlying factors where each observed variables is a linear function of these factors together with a residual variate. This model intends to reproduce the maximum correlations:

$$X_j = a_{j1}F_1 + a_{j2}F_2 + \ldots\ldots a_{jm}F_m + e_{j\ldots} \quad (3)$$

Where $j = 1, 2, \ldots, p$[16]

The factor loadings are *aj1, aj2,…,ajm* which denotes that *aj1*is the factor loading of *j*th variable on the 1st factor. The specific or unique factor is denoted by *ej*. The factor loadings give us an idea about how much the variable has contributed to the factor; the larger the factor loading the more the variable has contributed to that factor. Principal Component Analysis (PCA) was used to analyze the data using SPSS version 20. The final decision made during factor analysis is to know the number of factors to be retained from the resulting component matrix. In this study, the Social Science rule of thumb that only variables with loadings ≥ .4 should be considered relevant and extracted was used. The percentage variance observed was 31.52%. The remaining 68.48% may be due to some extraneous factors with very low responses among the knowledge

management practices in Nigerian software development organizations. Examples of such factors are 'Existing structures for acquiring and creating new knowledge assets', lack of documented knowledge within the organizations; lack of defined processes such as exit interviews to harvest knowledge from employees with relevant/key knowledge before retirement, lack of investment on knowledge management methodologies and tools, etc.

Factors are rotated for better interpretation since unrotated factors are ambiguous. The goal of rotation is to attain an optimal simple structure which attempts to have each variable load on as few factors as possible, but maximizes the number of high loadings on each variable. To obtain more meaning results, a Promax rotation (i.e an oblique rotation method) was performed. Oblique rotation produces a pattern matrix that contains the factor or item loadings and factor correlation matrix that includes the correlations between the factors. Promax rotation raise loadings from Varimax to some power (e.g., 4) and rotate the resulting matrix allowing for correlated factors. The factors and their loadings obtained from the Promax rotation are presented in Table 1.

Table 1: Extracted Component Matrix Using Promax Rotation (Only Factors ≥ .4 are extracted)

Item	Factor 1	Factor 2	Factor 3	Factor 4
Availability of Documented knowledge and manuals	-	-	-	-
Existing structures for acquiring and creating new knowledge assets	.635		.640	
Availability of up to date information to carry out work effectively	-	-	-	-
Effective frameworks for knowledge sharing among developers	.762			
Ability to Reference required information without contacting the person that created it	-	-	-	-
Reuse of Knowledge acquired from past software projects in the development of new software systems	.635		.640	
Horizontal Flow of knowledge in the organization across all levels	.652			
Existence of defined processes and procedures for software development which must not be deviated from				.491
Forum for Sharing and constructively addressing Information about failures, errors and mistakes			.907	
Emphasis by Management on capturing and storing knowledge within the organization	.496			.403
Emphasis by management on identifying and supporting network of experts and people with similar job-related interest	.864			
Conduct of exit interviews and debriefing to ensure that knowledge of exiting developers remains within the organization	.954			
A directory of experts that provide information during software development projects about their experience within the organization		.480	.589	.600
Regular training/workshops conducted within the organization to transfer knowledge	.890			
Regular training conducted on knowledge management practices		.980		
Sponsorship of developers to attend conferences to keep skills current		.995		
Existing structures in place for formal mentoring to share and retain knowledge	.970			
Use of knowledge from external sources				.989
Use of common knowledge domains			.628	.560

A total of four factors named: Knowledge Harvesting, Knowledge Creation, Management Commitment and Knowledge Management Culture were obtained. The factors and their loading are presented in Table 2.

Table 2: Factors and their Loading

Factor 1: Knowledge Harvesting

- Existing structures for acquiring and creating new knowledge assets
- Effective frameworks for knowledge sharing among developers
- Conduct of Exit Interviews and debriefing to ensure that knowledge of exiting developers remains within the organization
- Emphasis by management on identifying and supporting network of experts and people with similar job-related interest
- Reuse of Knowledge acquired from past software projects in the development of new software systems
- Horizontal Flow of knowledge in the organization across all levels
- Emphasis by Management on capturing and storing knowledge within the organization
- Conduct of exit interviews and debriefing to ensure that knowledge of exiting developers remains within the organization
- Regular training/workshops conducted within the organization to transfer knowledge
- Existing structures in place for formal mentoring to share and retain knowledge

Factor 2: Knowledge Creation

- A directory of experts that provide information during software development projects about their experience within the organization
- Regular training conducted on knowledge management practices
- Sponsorship of developers to attend conferences to keep skills current

Factor 3: Management Commitment

- Existing structures for acquiring and creating new knowledge assets
- Reuse of Knowledge acquired from past software projects in the development of new software systems
- Forum for Sharing and constructively addressing Information about failures, errors and mistakes
- A directory of experts that provide information during software development projects about their experience within the organization
- Use of common knowledge domains

Factor 4: Knowledge Management Culture

- Existence of defined processes and procedures for software development which must not be deviated from
- Emphasis by Management on capturing and storing knowledge within the organization
- A directory of experts that provide information during software development projects about their experience within the organization
- Use of knowledge from external sources
- Use of common knowledge domains

3.0 DISCUSSION

Results of analysis of the questionnaire showed that there are four major areas (Table 2) where knowledge management practices are adopted during software development projects. The level of adoption was reflected on the high loadings on some factors such as effective frameworks for knowledge sharing (76.2%), Emphasis by management on identifying and supporting network of experts and people with similar job-related interest (86.4%), Conduct of exit interviews and debriefing to ensure that knowledge of exiting developers remains within the organization(95.4%), Existing structures in place for formal mentoring to share and retain knowledge (97.0%), Forum for Sharing and constructively addressing Information about failures, errors and mistakes (90.7%), Use of knowledge from external sources (98.9%).

It was however discovered that functions relating to knowledge documentation had low factor loadings. Examples of such are: "Availability of Documented knowledge and manuals" and "Ability to Reference required information without contacting the person that created it" with loadings < .4 hence their values were not reflected in Table 1 (only factor loadings >= .4 were extracted). Other factors relating to knowledge documentation with low loadings are "Emphasis by Management on capturing and storing knowledge within the organization" (49.6%), A directory of experts that provide information during software development projects about their experience within the organization (48.0%), Emphasis by Management on capturing and storing knowledge within the organization (40.3%).

Based on the above findings, there is therefore need to harvest and codify knowledge for software project management. This will help software project managers to stop reinventing the wheel.

4.0 CONCLUSION

Knowledge management is critical to the success of software projects in today's knowledge driven economy. It aids knowledge retention, which is the strength on any organization. This paper showed there are four areas of knowledge management practices. Thus there is a need to adopt practices (Knowledge Harvesting, Knowledge Creation, Management Commitment and Knowledge Management Culture) that would promote knowledge management in Nigerian software organizations.

REFERENCES

[1]. Bjornson F. O. and Dingsoyr T. (2008) Knowledge Management in Software Engineering: A Systematic Review of Studied Concepts and Research Methods Used. Journal of information and software technology archive. 50(1)1055-1068

[2]. Humphrey, W.S. (1990). Managing the Software Process. Addison Wesley Longman Publishing Co., Inc. Boston, MA, USA

[3]. Davenport T.H., Prusak L. (1998), Working Knowledge: How Organizations Manage What They Know, Harvard Business School Press, Boston, MA, USA.

[4]. Rus I., Lindvall M., Sinha S.(2001), Knowledge management in software engineering A state-of the-art-report, Fraunhofer Center for Experimental Software Engineering and the University of Maryland. 1-57.

[5]. Iuliana S. (2009) A knowledge management practice investigation in Romanian software development organizations. WSEAS Transactions on Computers. 3(8) ISSN: 1109-2750.

[6]. Kruger, C.J. and Snyman, M.M.M. (2007). Guidelines for assessing the knowledge management maturity of organizations. South African Journal of Information Management. 9(3) 1-12

[7]. Balgheri R., Eslmain P., Mirfahkraee S. and Yarjanli M. (2013) The Evaluation of Knowledge Management Maturity Level in a Research Organization. *Australian Journal of Basic and Applied Sciences*, 7(2) 11-20

[8]. Kochikar, V.P.(2000) The Knowledge Management Maturity Model: A Staged Framework for Leveraging Knowledge, Infosys Technologies Limited. URL: http://www.infotoday.com/KMWorld2000/presentations/ kochikar.ppt.

[9]. Kulkarni, U., Freeze, R. (2004) Development and Validation of a Knowledge Management Capability Assessment Model, Proceedings of the 25th International Conference on Information Systems, Virginia, USA.

[10]. Kulkarni, U. and St. Louis, R. (2003). Organizational Self Assessment of Knowledge Management Maturity. Ninth Americas Conference On Information Systems 2542 – 2551Available at: http://citeseerx.ist.psu.edu/viewdoc/download? doi=10.1.1.86.8010&rep=rep1&type=pdf.

[11]. Oveh R.O &Egbokhare F. (2018) Preliminary studies on assessment of the level of knowledge management in Nigerian software development organisations. Nigerian Journal of Education, Health and Technology Research (NJEHETR) 10, 93-99

[12]. Mack N, Woodsong C, MacQueen KM, Guest G, & Namey E (2005) Qualitative research methods: a data collector's field guide. Research Triangle Park, North Carolina, Family Health International [FHI].pp 1-4

[13]. Coakes J. C. and Ong C. (2011) *SPSS Version 18.0 for Windows Analysis Without Anguish.* 1st Edition. Dougall Street, Milton: John Wiley & Sons Australia, Ltd

[14]. Fraenkel J. R. and Wallen N. E. (1996) *How to Design and Evaluate Research in Education*. Third Edition. McGraw Hill, Inc., New York.

[15]. Norusis M. J. (1994) *SPSS 6.1 Base System User's Guide Part 2.* SPSS, Inc, Chicago.

[16]. Yong and Pearce (2013) A beginners guide to factor analysis: Focusing on exploratory factor analysis. Retrieved 26/6/18 from www.researchgate.net