THE IMPACT OF FACTORS OF SUCCESS ON THE IN-HOUSE SOFTWARE DEVELOPMENT FOR PRESERVING TACIT KNOWLEDGE: SURVEY

Mohammed W. Shbair
Islamic University of Gaza
Gaza Strip, Palestine
mwshbair@gmail.com

Wael F. AlSarraj
Islamic University of Gaza
Gaza Strip, Palestine
wsarraj@iugaza.edu.ps

Khalid M. Kahloot
Islamic University of Gaza
Gaza Strip, Palestine
kkahloots@hotmail.com

ABSTRACT

Factors of success are aimed to provide knowledge-intensive organisations to better manage their knowledge value. There are multiple ways to capture organization's knowledge and make it available to all their members while it is not easy to capture/share the tacit knowledge among the stakeholders. The purpose of this study was to investigate the impact of factors of success on the in-house software development for preserving tacit knowledge. We conducted a survey to study the impact of these five factors on the tacit knowledge sharing between the developers within the in-house software development environment.

This paper is firstly exploring the definition of the knowledge and introducing the types of knowledge those are explicit and tacit knowledge. We discuss the in-house software development concept in which the non-IT organizations may need to develop their own software internally with no need to have a third party software development organization. For tacit knowledge sharing, we considered four factors reviewed in other researches and we added to them the pair programming as a practice. Case study is local bank in Palestine. Based on the results we have, it is confirmed the hypothesis of a positive impact of factors of success on the process of knowledge sharing.

Indexing terms/Keywords

Tacit Knowledge, Knowledge Sharing, Nontechnical Aspects, Software Development Practices, Pair Programming, Factors Of Success

Academic Discipline And Sub-Disciplines

KNOWLEDGE MANAGEMENT FOR IN–HOUSE SOFTWARE DEVELOPMENT

SUBJECT CLASSIFICATION

SOFTWARE ENGINEERING

TYPE (METHOD/APPROACH)

Survey

1. INTRODUCTION

It is important to distinguish between data, information and knowledge. Data is discrete, it is the essential raw material used to create information. Data become information when the meaning is added, also has meaningful structures. Knowledge is the interpretation of information within its context. It is the result of perception, learning and reasoning. Knowledge assists the organization to make decisions and take effective actions to achieve their businesses. There are two types of knowledge based on Polanyi's definition [1], namely explicit and tacit knowledge[2].

Tacit knowledge is defined as knowledge that is intuitive, unarticulated, non- verbalized or even non- verbalizable. Furthermore, tacit knowledge is built by self- processes like experience, reflection, internalization or individual talents [3]. Therefore, it is difficult to be managed and taught in the same manner as explicit knowledge. On the other hand, explicit knowledge, deals with more objective, rational and technical knowledge. It is simply articulated knowledge, which can be easily stored in files, folders, databases, documents, email messages or software codes.

Knowledge management (KM) is the process through which organizations generate value from their intellectual and knowledge-based assets. Most often generating value from such assets involves sharing them among developers and even with other departments in an effort to devise best practices [4]. Software developers are required to develop knowledge of emerging technologies, while at the same time ensuring that they adhere to organizational processes and methodologies. KM in the software engineering environments is seen as an opportunity to create a common language of understanding among the software developers, so that they can interact, negotiate and share knowledge and experiences.

In traditional models of software development process, such as waterfall, the knowledge is explicit, and could be easily captured from documents, manuals that are released. However the process models such as agile deals with tacit knowledge [5], [6]. Therefore, capturing the tacit knowledge and sharing among the developers is the main task. Knowledge management seeks to find ways to maximize the sharing of knowledge among the knowledge workers. Pair programming [5] is a practice that may provide assistance for some of the challenges faced by knowledge management in the case of tacit knowledge management.

In pair programming, two programmers work together in a single computer side-by-side work together on the same design, algorithm, code, or test. As a result, face-to-face communication enables knowledge dissemination between the pair, which promises to propagate the experience, to encourage the team members to make commitment, and to stimulate the interaction among team members. Our study is mainly focused on the In-house software development environments. In-house software is a software that is produced by a corporate entity for purpose of using it within the organization [7].

The need to develop such software may arise depending on many circumstances, which may be non-availability of the software in the market, potentiality or ability of the corporation to develop such software or to customize a software based on the organization's need. Usually In-house developers, devote their full attention to the projects. They can produce high-quality work faster, and they can fix bugs more efficiently. We need to study the impact of knowledge acquisition, information technology, sharing of knowledge, mechanism of assessment and pair programming in a local bank in Palestine. Then after, the identification of the impact of various KM practices in software engineering in addition to suggestion of suitable technical and social infrastructure to enhance KM capability.

The rest of this paper is organized below. The section two introduces the related work. The section three describe the problem statement. The research methodology detailed in section four. Section five listing the results while section six discuss the evaluation. Finally, section seven concludes the findings.

2. RELATED WORK

2.1 Knowledge Management

Conversion of tacit knowledge to explicit offers greater value to the organization [8]. Different methods such as direct interaction, practical experiences, social interaction and networking are suitable methods for sharing tacit knowledge in organizations. Conversion of knowledge from one form to another could lead to the creation of new knowledge. Nonaka and Konno[9] created a model of knowledge creation in the SECI model which includes four components such as socialization, externalization, combination, and internalization.

Socialization includes the essential social interaction (tacit knowledge is converted to tacit knowledge). Since then, both tacit and explicit knowledge were considered for Knowledge management activities in organizations [8]. Knowledge management has a variety of definitions but the key researcher I. Nonaka [10] mentioned that knowledge management means creating a learning environment to support knowledge creation and transfer that, use and reuse both personal and organizational knowledge. That is basically the purpose of knowledge management is to create and maintain a system which can produce, maintain, enhance, acquire, and transmit the enterprise’s knowledge base.

2.2 Software Engineering

The NATO conference held in the late 1960s mainly highlighted an idiom software crisis and schedule overruns of projects and problems with the quality and reliability of the delivered software. Software engineering is defined in IEEE standard glossary of software engineering as “the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is the application to software”. Here, engineering is defined as the application of a systematic, disciplined, quantifiable approach to structures, machines, products, systems, or processes [11]. The software process model defined as a simplified description of a software process.

Software process models are important means of implementing the software process in any software organization. software process means the set of activities required to produce a software system. Generic activities in all software processes are specification, development, validation and evolution [12]. Furthermore, many software process models have been designed in the literature [12] for structuring, describing and prescribing the process of building a software. Choosing right model for development of the software product or application is very important.

Software Engineering Body of Knowledge (SWEBOK) [13] is an international standard specifying guide to the generally accepted SE areas of Knowledge, to promote a consistent view of software engineering worldwide, to characterize the contents of the software engineering discipline and provide a topical guide to the literature describing the generally accepted knowledge within the discipline. There are ten main knowledge areas defined in SWEBOK. The description of the knowledge areas are designed to discriminate among the various important concepts, permitting the readers to find their way quickly to subjects of interest.

2.3 Knowledge Management in Software Engineering

SE knowledge is dynamic and evolves with technology, organizational culture and the changing needs of an organization’s software development practices. Kess and Haapasalo [14] argue that software processes are essentially knowledge processes, structured within a KM framework. Aurum et al. [15] point out that software development can be improved by recognizing related knowledge content and structure, as well as appropriate knowledge and engaging in planning activities. Basili et al. [16], [17] acknowledge that for an organization to implement the ‘Experience Factory’ (EF) approach for KM, a number of potential barriers to success must be overcome.

The 'Answer Garden' approach is depicted as a short-term solution to questions that may not require extended responses. Johansson et al [18] apply an 'Experience Engine' approach to KM in SE, as a subset of the EF. They list problems identified with the EF approach, such as its experimental nature, the organizational restructuring it prompts as well as its reliance upon an experience base containing a vast amount of written documentation. They assert that experience is best transferred when the receiver is "actually doing something related to the experience being transferred" [18].

Dingsoyr et al. [19] provide an insight into problems faced by small to medium organizations in addressing KM in SE. They consider postmortem reviews and experience reports as two approaches suitable for collecting software development knowledge. They conclude that lightweight postmortem reviews perhaps reveal more about software development practices, while experience reports are more suited to client relationships and interaction.

Rus and Lindvall [20] declare organizations must facilitate both formal and informal knowledge sharing between software developers. They assert that KM complements existing approaches to software process improvement, rather than seeking to replace them.KM activities designed to support SE are grouped into three categories: purpose of outputs, scope of inputs and effort required to process inputs. A number of options for implementing and using KM systems for SE are advanced, such as expert identification, the creation of KM champions, document management and using predictive modeling to direct decision-making.

2.4 Factors of success for knowledge management

Many studies are performed on knowledge management in organizations and its benefits to offer a comprehensive list of these success factors [21], [22]. (Wong, 2005) defined the success factors of knowledge management as the activities or actions which should be identified to ensure the success of KM implementation [23]. Accordingly, Davenport and Prusak defined eight factors in the management success [24], Ryan and Prybutok introduced five factors [25], Moffett et al. identified ten other factors [27] and finally, Chong and Choi recognized eleven factors in this regard [26].

Abundant models are offered on knowledge management, most of which are similar regarding content but different in terms of words and phrases. In these models, it is assumed that the stages and activities are mostly simultaneous, sometimes successive and seldom in a linear sequence. Identifying the primary factors affecting the KM success in organizations should be considered prior to establishment of a full-scale knowledge management program. Regarding explanation of how the knowledge is implemented,

Competitive Advantage Model is one of the best among them. Since little attention is paid to the factors of knowledge management establishment, the Competitive Advantage Model is chosen and studied as success factors of the presented knowledge management [28]. Contrary to other models, this one focuses on five factors effective on the success of knowledge management. It addresses the issues prior to knowledge establishment and organization readiness and pays attention to all organizational levels (individual, team and organization).

3. PROBLEM STATEMENT

Developers declare not to have the time to document. Often enough, the lack of time is an excuse for not being interested in documenting. Documentation can be a good means to knowledge preservation as articulated knowledge. Unfortunately, documentation fails with tacit knowledge, which leads to the absence of documenting the process of knowledge sharing. Organizations suffer from the lack of clear documents, which describe software development process. Actually, there is no documentation model can express the tacit knowledge.

Software development environments built a group of developers who had successfully solved similar problems by enabling them all to work together to create realistic solutions to the problems. Existing of such documents can play a significant role when one developer is absent. In addition to their role to prevent failure of the process of development. Thus, the importance of representing meaningful documentations in details should be highlighted. In this paper, we need to study the impact of knowledge acquisition, information technology, sharing of knowledge, mechanism of assessment and pair programming within software development environment in a local bank in Palestine. Various KM practices need to be addressed to study their impact on the software development process. Then after, the identification of the impact of various KM practices in software engineering in addition to suggestion of suitable technical and social infrastructure to enhance KM capability.

4. METHODOLOGY

4.1 Choosing Factors

We have collected primary data as well as secondary data from a local bank operating in Palestine. To help provide a project management perspective for managers responsible for in- house software development, we conducted a survey in an attempt to determine the factors that lead to successful knowledge management. We chose a survey because of its simplicity and because we hoped to find relationships among factors. We selected factors reviewed in [29] and we added one more factor to study which is pair programming as a practice. Drawing upon the review and analysis of factors of success for knowledge management, we discuss several important research issues and the role of IT in support of these processes. Factors are Knowledge Acquisition,

Information Technology, Sharing of Knowledge, Mechanism of Assessment and pair programming. Knowledge acquisition for software development processes is a purely practical problem to be solved by experiment, independent of software engineering. However, the conduct of experts will be influenced by implicit or explicit of KM. Knowledge acquisition covers all forms of knowledge and any methods by which they may be obtained. The importance of information technology in the knowledge management has become of particular at software development processes, in order to straighten and control processes of change and development within the organization.

Knowledge sharing is an activity through which knowledge (namely, information, skills, or expertise) is exchanged among developers in the organization. Knowledge Management Assessment of an Organization examine why we need to assess knowledge management in an organization and describe alternative approaches for assessing knowledge management in an organization. Pair programming is a practice of extreme programming [30], where two programmers, working side by side, develop the same piece of code. One programmer, usually named 'driver', actively writes the code while other programmer, usually named 'observer', identifies tactical and strategic defects and issues. The roles are periodically switched.

4.2 Target Group

We conducted wide-ranging, structured discussions with 15 senior software developers at a local bank to document their views regarding the factors of successful knowledge management in software development environments and the practices they consider important. We developed our survey based on these discussions, which focused on the knowledge transfer factors and best practices. A survey schedule has been designed and responses were collected on a predefined four-point rating scale. The data were interpreted on the basis of weighted scores for each parameters.

5. RESULTS

The results of the data analysis are presented as per the sequence of the survey schedule.

5.1 Knowledge Acquisition

The Knowledge Management Index for acquiring knowledge in the sample comes out at 59.3% [Refer Table No. 1]. It seems that the developers are aware of the importance of documentation but this is not enough. Information is documented in general but mainly critical information is recorded and archived. This may cause additional time-overhead on knowledge acquisition prior projects initialization.

5.2 Information Technology

Knowledge Management Index for information technology in the sample was acceptable, coming out at 45.0% [Refer Table. No. 2]. It is clear that there are working KM solutions for information sharing and centralized cataloguing of reports. These systems go a long way in managing knowledge throughout the organization. To be very effective in this process, we suggest to extensively using of information technology as a tool for dissemination of information and knowledge sharing.

5.3 Sharing of Knowledge

The Knowledge Management Index for sharing of knowledge in the sample comes out at 43.3%. According to the managers and developers, they have participated in various forums to share their knowledge. During the software development processes, sharing of knowledge brings about propagation of information and leads to a positive change in the productivity of the organization.

5.4 Mechanism of Assessment

The Knowledge Management Index for assessment mechanism in the sample population comes out at 73.3%. We encourage the organization to adapt a formal mechanism to transfer the knowledge gained through seminars and training data.

5.5 Pair Programming

The Knowledge Management Index for pair programming as a practice in the sample population comes out at 30.8%. This proofs that the pair programming is one of the best practice for knowledge sharing among the developers.

6. EVALUATION AND COMPARISION

We want to evaluate five factors from the Knowledge Management Index. Figure (1) demonstrates a comparison of acquiring knowledge, information technology, sharing of knowledge, assessment mechanism and pair programming in the sample of software developers. Acquiring knowledge comes out at 59.3%, which cause extra time-overhead prior projects initialization. Information technology coming out at 45.0 % but it needs better extension of using of information technology for dissemination of information.

Sharing of knowledge comes out at 43.3%. However, sharing of knowledge brings about propagation of information and leads to a positive change in the productivity of the organization. Assessment mechanism comes out at 73.3% but we encourage the organization to adapt seminars and training data. For pair programming comes out at 30.8%. This proofs that the pair programming is one of the best practice for knowledge sharing among the developers.

7. CONCLUSION

Software development requires the application of both explicit and tacit knowledge. With the first one, we mean knowledge in which can be formalized and transferred in the form of handbooks, tutorials, rules and procedures. Tacit knowledge cannot be formalized and transferred easily and the main means of diffusion is dialoguing. We claimed benefits of factors of success is fostering of tacit knowledge transfer between the team members. Thus, this practice could be applied specifically to development, with the purpose of improving tacit knowledge leveraging among developers. We selected the most important factors of success according to related work. The conjecture was that factors of success can enforce and speed up the building of development knowledge, that is mainly tacit and requires more than seminars and tutorials to be built.

Table 1

S. NO	Description	Hits(1=Yes, 4=No)				Weighted sum	Weighted average	KMI
		4	3	2	1			
1	Groups and individuals are routinely document and share information about their expertise	13	1	1	0	57	3.8	59.3%
2	Training on new systems focuses on how these technologies can be used efficiently to improve the quality and efficiency of people at work	5	6	1	3	43	2.87	
3	The electronic and traditional sources of knowledge contain a wide spectrum of state-of-art Information on critical activities	2	3	4	6	31	2.07	
4	Experts play a role in identifying collecting, classifying and disseminating important information for other developers.	2	2	4	7	29	1.93	
5	Effective cataloguing and archiving procedures are in place for document	0	1	1	13	18	1.2	

Table 2

S. NO	Description	Hits(1=Yes, 4=No)				Weighted sum	Weighted average	KMI
		4	3	2	1			
1	Information technology is the key enabler in ensuring that the	0	1	4	9	20	1.33	45.0%
2	There exists an explicit mechanism to translate ideas for business goals.	2	3	7	3	34	2.27	

Table 3

S. NO	Description	Hits(1=Yes, 4=No)				Weighted sum	Weighted average	KMI
		4	3	2	1			

1	Cross-functional groups are operational to promote knowledge sharing.	2	3	3	7	30	2.00	3.3%
2	Face-to-face interactions are used to transfer difficult to articulate tacit knowledge	0	1	2	12	19	1.27	
3	There is a program of active participation in business	2	2	4	7	29	1.93	

Table 4

S. N O	Description	Hits(1=Yes, 4=No)				Weighted sum	Weighted average	KMI
		4	3	2	1			
1	There is a review mechanism to assess whether the acquired knowledge is being transferred to the work place.	4	7	3	1	44	2.93	73.3 %

Table 5

S. N O	Description	Hits(1=Yes, 4=No)				Weighted sum	Weighted average	KMI
		4	3	2	1			
1	Developers are being encouraged to use pair programming	0	1	2	12	19	1.29	30.8%
2	Pair programming is helping to increase your experience	0	1	1	13	18	1.20	

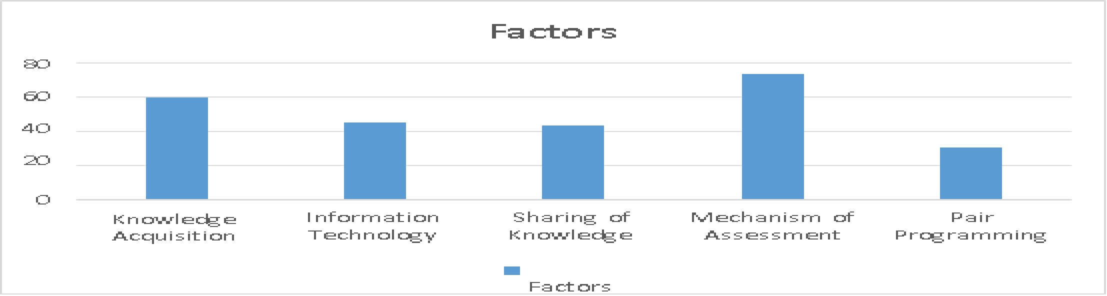

Figure 1.

REFERENCES

1.M. Polanyi, “The Tacit Dimension”. Anchor Day Books, 1966.

2. S. Vasanthapriyan, J. Tian, & J. Xiang, “A Survey on Knowledge Management in Software Engineering”, IEEE International Conference on Software Quality, Reliability and Security Companion, pp 237–244, Aug 2015.

3. Le Nguyen Haua, Felicitas Evangelista ,“Acquiring tacit and explicit marketing knowledge from foreign partners in IJVs”, June 2007.

4. Dimitris Apostolou ,“Managing Corporate Knowledge: A Comparative Analysis of Experiences in Consulting Firms.”, Knowledge and Process Management vol. 6 no. 3, pp. 129–138, 1999.

5. R. Kavitha, and I.Ahmed, “A knowledge managment framework for agile software development teams,” Int. Conf. Process Autom. Control Comput. PACC, pp. 1–5, 2011.

6. O. Hazzan, and M. Levy, “Knowledge management in practice: the case of agile software development in Cooperative and Human Aspects on Software Engineering”, in CHASE’09, ICSE Workshop on. IEEE, pp. 60–65, 2009.

7. J. Verner and W. Evanco , "In house software development: What software management practices are needed for project success" , IEEE Software. , vol. 22 , no. 1 , pp. 86 -93.

8. I. Nonaka, and H.Takeuchi, “The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation”, Oxford Univ. Press, 1995.

9. I. Nonaka, R. Toyama, and N. Konno, “SECI, Ba and leadership: a unified model of dynamic knowledge creation”, Long Range Plan, vol. 33, no. 1, pp. 5–34, 2000.

10. D. J. Teece, and I. Nonaka, “Managing industrial knowledge: creation, transfer and utilization”, Sage Publications Ltd, 2001.

11. IEEE, “IEEE Standard Glossary of Software Engineering Terminology”, IEEE Std. Institute of Electrical and Electronics Engineers, 1990.

12. I. Sommerville, “Software Engineering”, 9th ed. Addison-Wesley, 2010.

13. A. Abran , J.W. Moore, P.Bourque, and R.A.Dupuis, “Guide to the Software Engineering Body of Knowledge”, IEEE CS Pr. 2004.

14. P. Kess, H. Haapasalo, “Knowledge creation through a project review process in software production”, International Journal of Production Economics, vol. 80, no. 1, pp. 49-55, 2002.

15. A. Aurum, R. Jeffery, C. Wohlin, M. Handzic, “Managing Software Engineering Knowledge”, Springer, Germany, 2003.

16. V.R. Basili,G. Caldiera, H.D. Rombach, Experience Factory. In Marciniack, J.J. (Ed.) “Encyclopedia of Software Engineering”, John Wiley and Sonss, Hoboken, NJ, USA, 1994.

17. V. Basili, P. Costa, M. Lindvall, M. Mendonca,C. Seaman, “An Experience Management System for a Software Engineering Research Organisation”, Proceedings of 26th Annual NASA Goddard Software Engineering Workshop, pp. 29-35, 2001.

18. C. Johansson, P. Hall, M. Coquard, “Talk to Paula and Peter – They are Experienced”, Proceedings of the Workshop of Learning Software Organizations, Kaiserslautern, Germany, pp. 69-76, 1999.

19. T. Dingsøyr, N. Brede Moe, O. Nytro “Augmenting Experience reports with Lightweight Postmortem Reviews”, Third International Conference on Product Focused Software Process Improvement, PROFES 2001, Kaiserslautern, Germany, September 2001.

20. I. Rus, M. Lindvall, “Knowledge Management in Software Engineering”, IEEE Software, vol. 19, no. 3, pp. 26-38, 2002.

21. S. H. Siadat and L. Garshasbi, “Assessing the Impact of IT in Creating Organizational Knowledge (Case Study: Iran Central Bank)”, Management and Administrative Sciences Review. 3(7), pp. 1146-1156, 2014.

22. F. Soliman, and K. Spooner, "Strategies for Implementing Knowledge Management: Role of Human Resource Management", Journal of Knowledge Management, 4(4), pp. 337–345, 2000.

23. K. Y. Wong, "Critical Success Factors for Implementing Knowledge Management in Small and Medium Enterprises", Industrial Management and Data Systems, 105(3), pp. 261–279, 2005.

24. T.Davenport, and L. Prusak "Working Knowledge: How Organizations Manage What They Know", Boston: Harvard Business School Press, 1998.

25. S. D. Ryan, and V. R. Prybutok, "Factors Affecting Knowledge Management in Practice", California Management Review, 40(3), pp. 80–90, 2001.

26. S. C. Chong, and Y. S. Choi, "Critical Factors in the Successful Implementation of Knowledge Management", Journal of Knowledge Management Practice, vol. 6, 2005.

27. S. Moffett, R. McAdam, and S. Partkinson, "An Empirical Analysis of Knowledge Management Applications", Journal of Knowledge Management, 7 (3), pp. 6–26, 2003.

28. J. Barney, "Looking Inside for Competitive Advantage", Academy of Management Executive, 9(4), pp. 49–61, 1995.

29. Maryam Alavi and Dorothy E. Leidner, “Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues”, MISR Center, University of Minnesota , vol. 25, no. 1, pp. 107-136, Mar 2001.

30. K. Beck, “Extreme Programming Explained: embrace change”, Reading, Massachusetts, Addison-Wesley, 2000