

Knowledge Management in Software Development Based on the Leavitt's Model: Traditional versus Agile Methods

Taghi Javdani Gandomani*, Zeinab Tavakoli, Mina Ziaei Nafchi

Abstract— Knowledge management has been a challenging issue in software projects. Due to the knowledge driven nature of the software products, paying enough attention to this concept is necessary. Recently, Agile methods are widely used in software companies. A big challenge in these methods in associated with knowledge management, is lack of formal communication and documentation. Meanwhile, several knowledge management models have been proposed in which Leavitt's model is famous one. This study tries to adopt Leavitt's model to Agile methods and map this model to structure of the Agile methods.

Keywords— Agile software development, Agile methods, Knowledge management, Leavitt's model, Software engineering

I. Introduction

Knowledge is a critical factor which helps companies offer innovative services which encompass a wide range of knowledge and understanding [1]. Benefiting from knowledge and knowledge management has provided a firm situation for survival of dynamic and innovative companies and even the ability of companies for competition in market and business relies on the kind of acquisition, development, and utilizing personal and organizational knowledge [2]. So far, there was no consensus on the definition of knowledge management and the reason is multi-dimension nature of knowledge management in which researchers tend to look at this phenomenon from their own viewpoints and interests [3]. Some definitions are based on the knowledge management processes and define knowledge management as a process which deals with developing, storing, retrieving, and distributing information and companies' specialization in improving operations [1]. Some other definitions which are proposed by technology producers emphasize on the effects of technology on knowledge management [3]. These viewpoints believe that knowledge management is formed by extracting, packaging, managing, and reutilizing all of the different forms of knowledge and products of software engineering such as codes, designs, needs, models, and data in particular [4]. Some other researchers emphasize on cultural environment in which knowledge is evolved and they criticize technology-based view of the knowledge management [5].

knowledge management, or their approaches are not consistent with the well-known approaches, and human and technology process frameworks [5].

Considering knowledge management is also a major issue in software companies. Individual-oriented nature of these companies has led to form of the different types of knowledge in different levels of software companies and teams. These companies often use different methodologies for software development. Each methodology, due to its different viewpoint, defines different approaches, practices and activities for its software development cycle. It is also the case with knowledge management as well and in some cases the way of managing knowledge is obviously different from a specific methodology to the other one.

Today, traditional methods in software development have been replaced by Agile methods. One of the major different points of these two methodologies is the way of managing knowledge in them [6]. This has turned to be one of the serious and sensitive opposite points between the traditional and Agile methods [7]. Current study is an attempt to study this issue and offer a general comparison on the differences of viewpoints between the traditional and Agile software development methods from the viewpoint of knowledge management. The rest of this article has been organized as follows: Section 2 defines roles of methodology in software development. In Section 3 essential elements of knowledge management are introduced based on Leavitt's model. In Section 4 software development methodologies are explained, followed by Section 5 which explains adaptation of these methodologies to the elements of knowledge management. In section 6 Optimization of the Leavitt's model based on the adaptation to software development methodologies is discussed. Finally, Section 7 concludes the paper.

II. Role of Methodology in Software Development

Methodology means a collection of methods which are employed in software development life cycle [8]. In the development of a software system, there are three important processes including developing, managing, and supporting, the most important of which is the development process [8]. The process of development begins with designing a conceptual solution for requested problem and after receiving requirements and analyzing the system, designing is done. The designed system is finally converted to a real system using appropriate tools and technology [9, 10]. Final purpose of development process is in one way satisfying customers'

Taghi Javdani Gandomani*, Zeinab Tavakoli, Mina Ziaei Nafchi
Dept. of Computer Eng., Boroujen Branch, Islamic Azad University, Boroujen
Iran

Nonetheless, recent research studies reveal that most of the companies have not any specific approach regarding

needs and in another way guaranteeing the quality of the performance of system. Although for production of any kind of software the process of development can be different depending on the customers’ request. On the other hand, software development process doesn’t include only code writing, but it encompasses a cycle of all of the processes related to software production which is called Software Development Life Cycle (SDLC) [11] This cycle has some steps including needs analysis, system analysis, designing, executing, testing, and evaluation) [11].

III. Essential Elements of Knowledge Management

According to the Leavitt’s diamond model [12] knowledge management often comprises four essential elements including people, tasks, structure and technology. As shown in Fig. 1, each of these elements has a direct effect on other elements in a way that changes in each of them leads to changes in the others. Although this model has been extensively used for managing organizational changes, it seems that its elements have capability for being applied in software development methodologies as well.

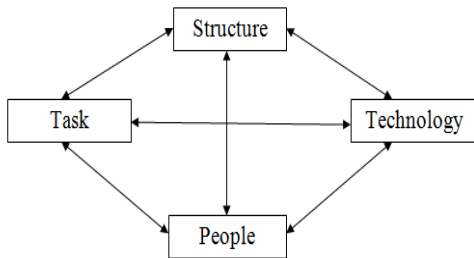


Figure 1. Leavitt’s knowledge management model ([10])

IV. Software development methodologies

Agile methodologies are based on Agile values and principles and emphasize on high levels of communications and collaborations with customers, acceptance of changes, and low documentation [13]. They make revision and reconsidering in project phases and compared to the traditional methodologies, they reduce errors and risks [14]. The important point in the difference of Agile and traditional methods is in the realm of knowledge management. The traditional methods, due to their predictive nature, are seriously required to record all of the steps related to the production, distribution, and updating knowledge and for this reason they pay special attention to the documentation. This is exactly contrary to the Agile methods which with their adaptive nature not only don’t pay much attention to the documentation, but also they produce and distribute knowledge in an implicit way. This leads to the major difference between the traditional and Agile methods in terms of adaptation to the knowledge management models.

V. Software Methods and Knowledge Management

Traditional and Agile methodologies are comparable from the point of view of factors such as viewpoint, emphasis, culture, method of management, basic programming, documentation, recurrences, team size, and criteria for success, as shown is Table 1 [14]. Considering that methodology selection before any action regarding software development is essential. In software development we deal with knowledge which needs right management. Therefore, we study elements forming knowledge management and adapting them to comparable factors in traditional and Agile methodologies with the purpose of being able to adopt an appropriate methodology for knowledge management in the process of software development in software companies.

TABLE I. AGILE VERSUS TRADITIONAL METHODS

Factor	Agile methods	Traditional methods
Viewpoint	Adaptive	Predictive
Culture	Collaboration-leadership	Command-control
Management style	Decentralized	Centralized
Initial planning	Limited	Comprehensive
Documentation	Little	Massive
Iteration	Much	Limited
Team size	Small	Large

A. People

Software development process consists of steps including needs analysis, system analysis, planning, executing, testing, and evaluation of software Thus, in this process, people as requesters or developers of software play essential roles. People directly affect viewpoint, culture, method of management, and size of the software development team [14]. In traditional methodologies peoples’ viewpoints are predictive and change-resistant whereas in Agile methodologies people have viewpoints which are compatible and adaptable to the changes [14]. However, this adaptive viewpoint in Agile methodologies might face with challenges due to the effect of cultural issues, wrong interpretations, lack of awareness of the procedure of the execution of Agile methods and lack of collaboration [8, 15]. people are the most essential elements of knowledge management [13]. People in small teams bring more creativity toward software development. Therefore, Agile methodologies could be more effective than traditional ones if they are supported by peoples’ culture.

B. Tasks

Task in methodologies is a phenomenon which refers to the existence of communicative situation and knowledge transfer channels for sharing knowledge and implementation of knowledge procedures [16]. This phenomenon comprises activities such as communication, knowledge circulation and knowledge management activities [16]. Software development methodologies have to do with procedures such as documentation, basic planning, and reiteration [14]. It has been seen in the comparison of procedures that in traditional methodologies documentation procedure is massive, basic planning is comprehensive, and reiterations are seldom but big. Contrary to the ones in the traditional methodologies, in Agile methodologies documentation procedure is seldom, basic planning is limited, and there are many but small reiterations [14]. As knowledge management refers to a collection of procedures or activities which are done in order to create, acquire, store, and distribute knowledge in companies [17]. From the viewpoint of tasks or procedures which are the second essential elements of knowledge management Agile methodologies are also better than traditional ones for software development because in the process of knowledge management it is of essential importance that procedures be clear and simple as far as possible in a way that they are understandable for all of the employees and this is definitely in contrast with massive documentation and comprehensive planning existing in traditional methodologies and as in software development, procedures are different due to customer's need, small reiterations in Agile methodologies can lead to the emergence of knowledge derived from feedback received from customer having different needs and the more the reiterations are, the more the achieved knowledge can recount customer's needs and can lead to the production of a better product finally.

C. Structure

Structure means arrangement of elements, format and foundation of something. In the traditional methodologies an ordinal and regular structure is often dominated [14]. This ordinal structure means that there will not be the possibility of doing to the next steps unless all of the procedures related to a specific step are finished. Therefore, due to the traditional methods' dominated structure, there is an explicit and well-defined knowledge in them which is not needed to be managed. However, in Agile methodologies structure is based on four bases. Agility bases include individuals and interactions versus processes and tools, working software versus comprehensive documents, customer collaboration versus following a contract, and responding to changes versus following a comprehensive plan [18]. Therefore, it seems that in Agile approach explicit knowledge has been replaced by implicit knowledge due to lack of existence of an ordinal structure on one hand and a structure based on human interactions on the other hand [14]. Thus, it can be concluded that knowledge management is a requirement in Agile methods.

D. Technology

Nowadays, companies put too much importance on understanding, adaptability, and management of changes of the surrounding environment and surpass each other in achieving and applying up-to-date knowledge and information in order to improve activities and offering better services [19]. Companies are increasingly dynamic and a major part of this dynamic nature is created by revolutions connected with technology [20]. In order to develop efficient knowledge management skills in companies becoming coordinated with technology is critical. Software engineers are usually used to utilizing new equipment and technologies because they always believe that these technologies can help them in the development of more appropriate software. Thus, it is clear that software development process should always be coordinated with new technology [20]. It seems that in all of the steps of the knowledge management cycle including knowledge achievement, division, and sharing applying knowledge and creating technological knowledge have an essential role and both traditional and Agile methodologies of software development are based on technology, because creating and developing software have to do with human confrontation with technology and his inclination to utilize this technology in all fields and technology is the main axis of the human's knowledge and rationalism and his thoughts for utilizing ideas and giving reiterative and non-innovative affairs to the machine and also increasing efficacy and freeing human skills [20]. What is important meanwhile, is how technology is used in traditional and Agile methodologies. For instance, in traditional methodologies there are comprehensive documents in software (electronic) fields but this is not the case with Agile methodologies because of little documents. This is while regarding the communication technologies in Agile methodologies due to face-to-face and two-way communications the role of communication technology is more extremely important than that in traditional methodologies. Thus, it can be concluded that in the fourth apex of knowledge management which is technology it is seen that selection of the kind of traditional or Agile methodologies makes no significant difference in the process of knowledge management in software development procedure, although it could be claimed that Agile methodologies could be more adoptable to the information technology due to their new emergence.

VI. Leavitt's model optimization

Based on the adaptation of the major elements of Leavitt's knowledge management model to software development methodologies as in the previous sections, it seems that this model can be optimized as seen in Fig. 2.

As can be seen in Fig. 2, if knowledge management is considered as the central part of Leavitt's model, four elements of Leavitt's model can be optimized through four major factors:

- 'People element': helps for optimization of knowledge management by providing an appropriate situation for changes in culture in Agile teams.

- 'Task element': helps for optimization by clarification of the duties which exist obviously in Agile methods.

- 'Structure element': leads to the optimization of Leavitt's model through obligation of the implicit knowledge to be converted to the explicit one.

- 'Technology element': facilitates optimization through alignment with information technology.

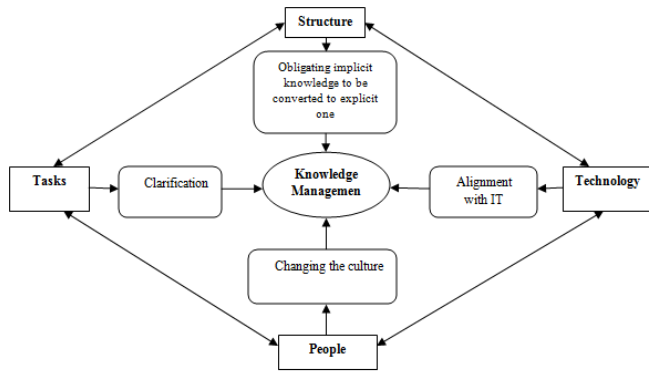


Figure 2. Optimized form of Leavitt's knowledge management model

VII. Conclusion

In this study, adopting of the essential factors constituting knowledge management including people, procedures, and technology to comparable factors in software development methodologies was examined to utilize the received feedback in selection of an appropriate methodology of knowledge management in the process of software development in software companies. It has been seen that as people in Agile development teams having an adoptable and conformable view to the changes show more creativity toward the knowledge management achieved through communication and interaction with customers and procedures such as short-time reiterations utilized in Agile methodologies can lead to the emergence of the knowledge achieved through feedback received from customer having different needs. In Agile structures, due to lack of the existence of an ordinal structure, the explicit knowledge has been replaced by an implicit one. This has made knowledge management as a requirement in these methods. Therefore, it seems that from the viewpoints of the people, tasks, and structure, selection of the Agile methodologies not only can be more influential for knowledge management in software development process in software companies, but also it has become somewhat a requirement. Regarding the fourth element which is technology, due to the reason that creation and development of software are essentially related to the human facing with technology and his inclination for utilizing this technology in all of the fields, selection of either the traditional or the Agile methodologies makes no significant difference in knowledge management in software development procedure and what is important here is how technology is used in these methodologies, although it

might be claimed that Agile methodologies due to their new emergence could be more adoptable to the new technologies.

References

- [1] B. Gupta, L. S. Iyer, and J. E. Aronson, "Knowledge management: practices and challenges," *Industrial Management & Data Systems*, vol. 100, no. 1, pp. 17-21, 2000.
- [2] Y.-Y. Chen and H.-L. Huang, "Knowledge management fit and its implications for business performance: A profile deviation analysis," *Knowledge-based systems*, vol. 27, pp. 262-270, 2012.
- [3] E. Serna, "Maturity model of Knowledge Management in the interpretivist perspective," *International Journal of Information Management*, vol. 32, no. 4, pp. 365-371, 2012.
- [4] M. Levy and O. Hazzan, "Knowledge management in practice: the case of agile software development," in *Cooperative and Human Aspects on Software Engineering, 2009. CHASE'09. ICSE Workshop on*, 2009, pp. 60-65: IEEE.
- [5] M. Gloet and M. Terziovski, "Exploring the relationship between knowledge management practices and innovation performance," *Journal of Manufacturing Technology Management*, vol. 15, no. 5, pp. 402-409, 2004.
- [6] T. J. Gandomani, H. Zulzalil, and M. Z. Nafchi, "Agile Transformation: What is it about?," in *8th Malaysian Software Engineering Conference (MySEC)*, Langkawi, Malaysia, 2014, pp. 240-245: IEEE Computer Society.
- [7] T. J. Gandomani and M. Z. Nafchi, "The Essential Prerequisites of Agile Transition and Adoption: a Grounded Theory Approach," *Journal of Internet Computing and Services (JICS)*, vol. 5, pp. 173-183, 2016.
- [8] T. J. Gandomani, H. Zulzalil, A. A. A. Ghani, A. B. M. Sultan, and M. Z. Nafchi, "Obstacles in moving to agile software development methods; at a glance," *Journal of Computer Science*, vol. 9, no. 5, p. 620, 2013.
- [9] (2008). *SELECTING A DEVELOPMENT APPROACH*. Available: <https://www.scribd.com/doc/.../Selecting-Development-Approach>
- [10] A. Cockburn and J. Highsmith, "Agile software development, the people factor," *Computer*, vol. 34, no. 11, pp. 131-133, 2001.
- [11] A. Singh, K. Singh, and N. Sharma, "Agile knowledge management: a survey of Indian perceptions," *Innovations in Systems and Software Engineering*, vol. 10, no. 4, pp. 297-315, 2014.
- [12] H. J. Leavitt and J. G. March, *Applied organizational change in industry: Structural, technological and humanistic approaches*. Carnegie Institute of Technology, Graduate School of Industrial Administration Pittsburgh, Pennsylvania, 1962.
- [13] k. beck. (2001). *Manifesto for Agile Software Development*. Available: www.agilemanifesto.com
- [14] R. S. Pressman, *Software engineering: a practitioner's approach*. Palgrave Macmillan, 2005.
- [15] T. J. Gandomani and M. Z. Nafchi, "Agile transition and adoption human-related challenges and issues: A Grounded Theory approach," *Computers in Human Behavior*, vol. 62, pp. 257-266, 2016.
- [16] E. W. Ngai and E. Chan, "Evaluation of knowledge management tools using AHP," *Expert systems with applications*, vol. 29, no. 4, pp. 889-899, 2005.
- [17] K. C. Agbim, G. O. Oriarewo, and Z. B. Owutuamor, "The Impact of Knowledge Management Capabilities on Organizational Performance: A Survey of the Service Sector," *Journal of Business Management & Social Sciences Research (JBM&SSR) ISSN No: 2319*, vol. 5614, 2013.
- [18] D. Cohen, M. Lindvall, and P. Costa, "An introduction to Agile methods," *Advances in computers*, vol. 62, pp. 1-66, 2004.
- [19] M.-Y. Chen, M.-J. Huang, and Y.-C. Cheng, "Measuring knowledge management performance using a competitive perspective: An empirical study," *Expert Systems with Applications*, vol. 36, no. 4, pp. 8449-8459, 2009.
- [20] I. Rus and M. Lindvall, "Knowledge management in software engineering," *IEEE software*, vol. 19, no. 3, p. 26, 2002.