A knowledge management practice investigation in Romanian software development organizations

SCORȚA IULIANA
The Economic Informatics Department
The Academy of Economic Studies
Bucharest, Piața Romană no.6
ROMANIA
iuliana.scorta@ie.ase.ro

Abstract: - Knowledge plays an increasingly larger role in organizations and many consider it the most important factor of production in a knowledge economy. Knowledge is dynamic and evolves with technology, organizational culture and the changing needs of organization's software development practices. In this paper I frame my research by discussing the importance of knowledge management in software engineering. After presenting the landscape inspiring the study research questions I conduct an investigation for evaluating aspects of knowledge management practice in Romanian software engineering industry and discuss the major findings.

Key-Words: - Tacit knowledge, Explicit knowledge, Knowledge management, Romanian software engineering.

1 Introduction

In today's fast growing IT industry many companies are constantly experiencing shortages of software developers, a key resource in the computer-based world that we live in. This frustrating reality comes from the fact that training a competitive software developer is a long process that may take years which is incompatible with the much faster ups and downs of the software industry. As a consequence many managers from software development are attempting to accelerate the process of training software developers [8]. In this context, knowledge is the most important value of an organization, influencing its competitiveness. In software development organizations, knowledge management (KM) can be used to capture the knowledge and experience that the employees accumulate during the software development process. KM can help software engineers to improve their efficiency, and managers to capture the domain knowledge that software developers acquire during their work. There is currently a gap in literature concerning the implementation of KM practices or KMS (knowledge management systems) in Romanian software development organizations that I will attempt to reduce by conducting an investigation.

2 Knowledge

Knowledge (K) represents a mix of framed experience, values, contextual information and expert insights that provides a framework for evaluating and incorporating new experiences and information. In organizations, knowledge often becomes embedded, not only in documents and repositories, but also in organizational routines, processes, practices and norms [12].

There are different levels of refinement to the items related to knowledge, the lowest one being data, followed by information, and knowledge at the highest level. The relations between data, information and knowledge can be described by a hierarchy pyramid [15]. Data is essential raw material for the creation of information. Information is data that is organized in a way that makes it useful for an end-user when making decisions. Knowledge is broader than information and data and requires understanding of information. Experience is applied knowledge [29].

Knowledge can be classified into general knowledge and specific knowledge. General knowledge is broad, often publicly available, commonly shared knowledge. In contrast, specific knowledge is context specific in an organization [31], [32].

Knowledge has many features, attributes, and dimensions. For the purpose of this study I must explain the concept of tacit and explicit knowledge.

Explicit knowledge can be expressed in words and numbers and shared in the form of data, scientific formulae, manuals and the like. Explicit knowledge is precisely and formally articulated, although removed from the organizational context of creation

or use and can be readily transmitted between individuals [24], [26]. Explicit knowledge plays an increasingly larger role in organizations and many consider it the most important factor of production in a knowledge economy [26].

Tacit knowledge is subconsciously understood and applied, difficult to express, emerged from direct experience and action, and usually shared through highly interactive conversation and shared experiences. Researchers stated that are two dimensions to tacit knowledge. The first is the technical dimension, which encompasses the kind of informal personal skills or crafts often referred to as „know-how". The second is the cognitive dimension. It consists of believes, ideals, values and mental models which are deeply ingrained in us and which we often take for granted. While difficult to articulate, this cognitive dimension of tacit knowledge shapes the way we perceive the world [24].

In traditional perception of the role of knowledge in any business tacit knowledge is often emphasized as key for success and creation of new values. Reason for that view is the fact that explicit knowledge is useful only in combination with individual's tacit knowledge, for which some assessments say that constitutes up to 80% of all knowledge in company [15]. Knight and Howes identified in 2003 six types of tacit knowledge:

- Know-how: knowledge that defines procedures needed for successful accomplishment of tasks.
- Know-who: knowledge that allows individuals faced with a specific problem to contact the right person in a search for solution.
- Know-why: knowledge which explains why is something done or what the desirable objective is.
- Know-when: knowledge that defines when to do something, and when not to.
- Know-where: knowledge which allows people to find what they need.
- Know-that: instinctive knowledge that defines correct course of action.

There are four possible conversion paths between these two types of knowledge: socialization which involves capturing tacit knowledge through physical proximity and disseminating it among colleagues, externalization which means the translation of tacit knowledge in comprehensible forms that can be understood by others and also translation of highly professional knowledge in explicit knowledge, combination in which stage are taking place communication and diffusion processes and the systemization of knowledge and internalization when explicit knowledge is embodied in action and practice and in this way it actualizes concepts or methods about strategy, tactics, innovation or improvement [24].

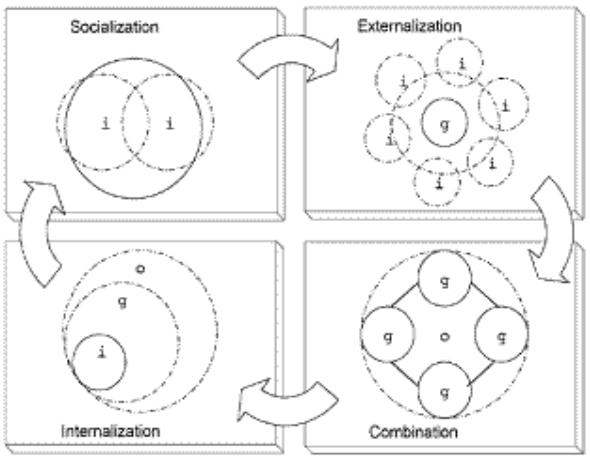

Fig. 1 Adapted SECI knowledge spiral from reference [24] - (i) Individual; (g) group; (o) organization.

The model presented in Figure 1 illustrates these four paths and the inherent relationships and evolving nature among these four conversions. In an organization, knowledge can be retained at three levels: individual, group, and organization. The model describes how knowledge changes. The process begins in the socialization quadrant, where knowledge is tacit existing in individuals. This allows exchange of thoughts and ideas between individuals leading to an improved understanding of the system, which results in knowledge creation. Once the tacit knowledge has been created, it can be formalized and standardized in order to be communicated in groups, which leads to explicit knowledge. Once explicit knowledge is created, it can be combined with other explicit knowledge and expressed in a format that it can be retained at the organizational level. The application of explicit knowledge occurs in the final quadrant where each group and individual assimilates and internalizes the knowledge [23]. To be effective knowledge management must enable the conversion of knowledge from the tacit to the explicit.

Also, knowledge can be internal or external to a software engineering organization. Internal knowledge tends to be unique, is highly important in gaining strategic advantages for the organization, is specific and tacit and resides within employees of the organization, embedded in behaviours. The external knowledge is far less valuable than the first, due to the fact that it can only provide a new way of

thinking in the organization which is also available to competitors.

Depending on the set of activities in software engineering to which knowledge pertains, there can be different types of knowledge, such as [29]:

- Organizational knowledge, for example, how to run the company, what are the business objectives, human resources aspects, etc. This type of knowledge for a software organization might not differ too much from the organizational knowledge for other industries.
- Managerial knowledge is related to planning, staffing, tracking, and leading a project.
- Engineering knowledge refers to development knowledge and skills, such as requirements analysis, designing, programming, testing, and technical writing, using specific tools and methods (such as object oriented development) or specific programming languages.
- Domain knowledge, related to the application domain and the specific system to which the software pertains (for example office support tools or bank transactions).

As a human and knowledge-intensive work [5], the software development process involves both explicit and tacit knowledge. In the context of software development explicit knowledge includes software engineering (SE) methods, document templates, components, software artefacts, and so on, while tacit knowledge is embedded in an individual experience obtained through discussions and lessons learned [3].

Software engineering knowledge is dynamic and evolves with technology, organizational culture and the changing needs of organization's software development practices [29]. After an extensive research, Bjørnson and Dingsøyr concluded that the major finding on knowledge management in software engineering which is repeated over several papers and across several schools is the need to not focus exclusively on explicit knowledge but also on tacit knowledge [6].

3 KM in software development organizations

Knowledge management is seen as a strategy (or practice, systematic process, set of policies, procedures and technologies) that creates, acquires, transfers, brings to the surface, consolidates, distils, promotes creation, sharing, and enhances the use of knowledge (or information, intellectual assets, intellectual capital) in order to improve organizational performance; support organizational adaptation, survival and competence; gain competitive advantage and customer commitment; improve employees' comprehension; protect intellectual assets; enhance decisions, services, and products; and reflect new knowledge and insights [29].

The knowledge management cycle includes the following steps: knowledge identification and capture, knowledge sharing, knowledge application, and knowledge creation [7]. According to newest advances the KM cycle has evolved and includes the steps presented in Figure 2.

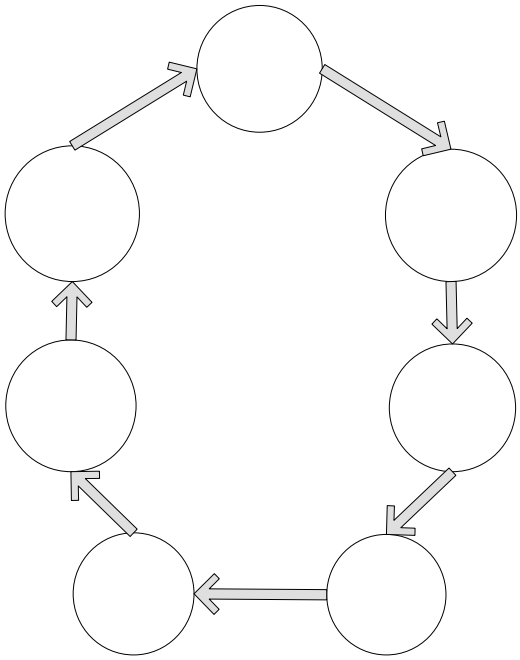

Fig. 2 KM cycle (adapted after the Organizational Knowledge Evolution Cycle in [34])

Once the critical knowledge is identified and captured, it is typically shared with others. Those individuals then apply this knowledge and internalize it to their situation, which in turn creates new knowledge. This new knowledge is then captured, shared and applied and the cycle continues [7]. Generally, knowledge sharing between individuals can be ad hoc or systematic. Transfer of knowledge from one software developer to another can happen on an ad hoc basis within a project or an organization. This occurs when individuals initiate communication, for example, when they need to solve a problem and ask for help from other individuals who are known to have the appropriate expertise. If this communication and sharing is systematic and there is a process in place to document it, then exchanged software engineering knowledge will be captured and organized into an

organizational memory. Thus, the next time this piece of knowledge is needed, it will be retrieved from this repository rather than solicited from an individual. This will lead to time-savings, both for the solicitor of the information and for the provider.

Software engineering involves a multitude of knowledge-intensive tasks: analyzing user requirements for new software systems, identifying and applying best software development practices, collecting experience about project planning and risk management, and many others [5]. Rus et. all identified three main categories for software engineering tasks [29]:

- Tasks performed by a team focusing on developing a software product based on customer requirements. This represents the core task of any software organization. The project manager is responsible for ensuring that work is completed on time and within budget and possesses the intended functionality and quality. Software Engineering is document-oriented and what is produced during the project is a set of documents such as contracts, project plans, requirements and design specifications, source code, test plans and related documents. These documents are not just work products. There is also additional information embedded within them: during the project they document the decisions; after the project's completion, they contain the history of the project. The documents can be reused in different ways by the next project so that people can learn from them, by analyzing the solutions to different problems that these documents capture.
- Tasks that focus on improving a team's ability to develop a software product (that is improving tasks in the first category). Here I can include tasks that might be conducted during and shortly after the project. The reason for performing these tasks is to ensure that potential knowledge gained in the project is not lost. Included here are all forms of lessons learned and post-mortem analyses that identify what went right or wrong in the project. Also included are analyses of data from the project, for example, comparisons of budgeted and actual costs, estimated and actual effort, and planned and actual calendar time. Tasks in this category attempt to collect and create knowledge about one particular project. The results from this activity are useful by themselves, but can also be the basis for further learning. They can be stored in repositories and experience bases (for example, in lessons learned repositories).
- Tasks that focus on improving an organization's or an industry's ability to develop software. This category represents activities that analyze results from several previous projects in order to identify similarities and differences between them. The insights gathered by these analyses can be formulated as knowledge or experience packages and can be qualitative, quantitative, or a mix of both. Examples of qualitative packages are patterns, heuristics and best practices based on a number of experiences from different sources. Examples of quantitative packages are estimation models based on the measured attributes of previous projects and their budgeted and actual outcome. Other examples are knowledge that is packaged in terms of executable software programs that automate steps of the development process based on knowledge derived from previous projects. Industry-wide standards and recommendations also fall into this category.

Based on these three categories Rus et. all stated that there are three levels of KM in software engineering:

- First level KM as support for core software engineering activities – they consider that because software engineering is so dominated by the documents that are produced during the various activities and processes, the foundation for a knowledge management system is a document management system. Also, in order to fully utilize the competence of the organization there is a need for keeping track of who knows what, therefore an elaborated solution to this problem is competence management or skills management or expert network. Competence management systems were initially developed with the major objectives of being able to find employees with the right skills in order to staff new projects and to find individuals who have specific pieces of knowledge or the identification of de-facto experts (the solution for identifying experts was to assume that one's expertise in a certain area is reflected by the documents they author).
- Second level KM as support for enhancing organizational memory for software development – the researchers distinguished between forms of organizational memory:
  - Memory consisting of regular work documents and other artefacts that were developed primarily to assist development of the product (examples in this category are

requirements specification, and design specification)
- Memory consisting of entities that were developed specifically to support the organizational analyses)
- A mix of the first two forms;

and presented regular software tools used by software engineers in their daily work that also support the creation of an organizational memory.

- Third level KM in which the researchers include packaged knowledge that supports knowledge application - Rus et. all classified the available tools according to:
  - *The KM life cycle as:*
    - Tools supporting knowledge deployment and application.
    - Tools supporting knowledge acquisition.
    - Knowledge organization tools.
  - *The software engineering activity that they support:*
    - Interactive Domain Understanding Tools.
    - Intelligent Requirements Assistants.
    - Knowledge Based Program Designers.
    - Knowledge Based Code Generators.
    - Smart Code Analysis Tools.
    - Documentation Generators.
    - Software Maintenance Tools.
    - Predictive Models and Best Practices.
    - Process Design.
  - *Knowledge Life Cycle Phases That They Support:*
    - Knowledge Deployment/Application Tools.
    - Knowledge Organization Tools.
    - Knowledge Acquisition Tools.

In the context of software development, KM can be used to capture the knowledge and experience generated during the software process. Although every software development project is unique in some sense, similar experiences can help developers to perform their activities. Reusing knowledge can prevent the repetition of past failures and guide the solution of recurrent problems [11].

Knowledge in software engineering context is managed both using the codification approach (focuses on amalgamating individual knowledge in organizations, putting it in a cohesive context and making it available to organizational members) and the personalization approach (knowledge sharing is fostered through people-to-people interactions and dialogue) [13].

In software engineering, reusing life cycle experience, processes and products for software development is often referred to as having an "Experience Factory" [3]. In this framework, experience is collected from software development projects, and are packaged and stored in an experience base. By packing, I mean generalizing, tailoring, and formalizing experience so that it is easy to reuse [6]. The Experience Factory enables organizational learning and acknowledges the need for a separate support organization that supports the project organization in order to manage and learn from its own experience. The support organization helps the project organization observe and collect data about itself, builds models and draws conclusions based on that data, packages the experience for further reuse, and most importantly, feeds the experience back to the project organization [29]. The Experience Factory approach was initially designed for software organizations and takes into account the software discipline's experimental, evolutionary, and non- repetitive characteristics. The Experience Factory approach has components that address capturing, storing, distributing, applying, and creating new experience. It also has components that address analysis and synthesis of knowledge [4]. A physical implementation of the Experience Factory in an organization is called the Experience Management System (EMS). The EMS is composed of content, structure, procedures and tools. The content can be data, information, knowledge or experience. Structure is the way the content is organized. The content and the structure are often referred to as the experience base. Procedures are instructions on how to manage the experience base on a daily basis, including how to use, package, delete, integrate and update knowledge. Tools support managing the content and the structure, and carrying out the procedures, as well as helping capture, store, integrate, analyze, synthesize and retrieve knowledge [4].

In contrast to this codification approach, wherein a central repository of knowledge is offered, is the personalization approach in which knowledge sharing between individuals is realized in an ad-hoc or organized manner.

An efficient knowledge management approach (in software development organizations) must be able to model, capture and support the creation and use of both explicit and tacit knowledge [25]. Corbin et. all stated that are several misconceptions about KM in SE and that in order to have an effective knowledge management, it is necessary to adopt a systematic scheme of planning, stipulating, and distributing the knowledge management tasks and activities. They presented a three-tier approach that treats the people

and SE process as central pieces of the KM effort [10].
Also, researchers studied the role of KM in software development process and suggested that KM can be conceptualized as a major component of Software Process Improvement (SPI) initiatives [22] or argued that KM in SE assists software developers in defining software processes, pursuing a process-oriented approach and improving and adapting existing software processes for future use [19]. Other several studies were focused on the application of KM in software engineering organizations [16], [17], [20] and have revealed favourable views of existing KM models and frameworks specifically developed for SE, while others showed a negative view towards KM in SE organizations [28].

4 Research

In this study, I used a KM model to investigate the KM process for SE in Romanian organizations, developed by Aurum et al. [2] based on the SECI model (socialization, externalization, combination, internalization), the Experience Factory model and a third model developed by American Productivity & Quality Centre and Arthur Andersen. This integrated KM framework presents a list of major KM enablers, KM process activities and corresponding KMS (as presented in Table 1).

KM enabler	KM activities	KMS
Technology	K identification	Communities of practice
Culture	K acquisition	Personal networks
Management Leadership	K creation	Organisational practices and routines
	K organisation	Document management system
		Expert systems
		Organisational routines
	K transfer	Training
		Informal networks
		Groupware
	K application	New IT
	K adoption	New products and services
		New markets

Table 1 The integrated KM framework [2].

According to Aurum et al., this model allows an analysis of various aspects of both organizational learning and effectiveness of knowledge workers that use various technologies and under different cultural environments and leadership styles [2].
The primary objective of this study is to provide a description of the KM process as applied by a sample of Romanian organizations engaged in software development. As far as the authors are aware no empirical investigation has been conducted by Romanian SE industry or other Romanian institutions for evaluating aspects of KM practice in Romanian SE organizations. In this research I have selected two industry-based Romanian organizations that claimed to engage themselves in KM practices. The research objectives are to identify: the current state of practices for KM in SE, the KM activities that comprise the KM process for SE and the level of impact of leadership, technology, culture and measurement as enablers of the KM process for SE. Data collection for this research was conducted via interviews among the employees of the selected organizations. In order to establish the current state of practices for KM in SE the employees were asked to define the concept of knowledge, to explain what they understand by knowledge management, to express which is their motivation for sharing knowledge and to present knowledge sources and tools they use or techniques and methods they apply. The second objective of the research was to examine the KM-related activities that are applied in the context of Romanian SE organizations. After a brief presentation of the KM activities and the main knowledge types, the respondents were asked to:

- Specify what KM activities were performed in an explicit manner or not performed at all;
- Describe what are the activities they perform that are leading towards the development of tacit knowledge and how they think this problem should be addressed;
- Rate the level of effort invested on KM activities in the projects they were involved in by assigning a percentage to each activity in relation to the total effort for KM in SE.

To determine the level of impact of the KM enablers the employees responded to a set of questions in which they rated the importance of each enabler of the KM process for SE using a Likert scale.

5 Research results

5.1 Current status of KM practices application

The research results showed many similarities in the current state for KM in SE in both Romanian organizations.
The employees of both companies were aware that KM would ultimately improve the quality of the work they produced. Still, not all participants fully understand the concept of knowledge. Several

employees from both organizations did not make a clear distinction between the concept of information and knowledge. In both organizations, most respondents claimed that their primary motivation for sharing knowledge with other colleagues is the need for them to perform at the same level in order to be able to finish the projects at the previous established term. Although this is a common answer, several employees motivated that they are reluctant to share knowledge with their colleagues and their motivation was the fear of being known as an expert. They argued that because of this stereotype too much of their work time is allocated to teach other colleagues or that they are assigned to projects according to their past experience. Since the SE is a continuously evolving field they fear that in this way they will not have enough time to allocate to the new developments and experimenting and that this will affect their carriers. In contrast, other employees stated that they are afraid to lose the expert status because they will become expendable as soon as their employers have captured all of the knowledge they need.

The employees of the first company were motivated to extract knowledge from the current projects they were involved in and to formalize it in an explicit manner. They concentrated their efforts on organizing their knowledge for future work, but they used an inappropriate knowledge storage tool that according to them was not very easy to use or as helpful as it should be. The participants from the second company stated that they did not use any tools for knowledge storing. Also, they stated that their colleagues form the main knowledge source and that they usually transfer knowledge using email and instant messaging tools or while engaging themselves in coffee breaks. Both organizations employees stated that they also use other sources of knowledge like Internet and magazines but some of them were concerned with the validity of the knowledge acquired in this manner. I noticed that the second organization's management was making efforts to promote standards and process methodologies to support the KM activities mainly the knowledge organization activity.

5.2 KM activities examination

Explicit knowledge increases the potential problem-solving ability of a development team and also facilitates decision-making activities during the software development process, through providing the possibility of knowledge transfer and knowledge integration [2]. Because of the two main characteristics presented above that play a significant role in reducing software development challenges [33], I decided to make an evaluation of the level of explicit and tacit knowledge within both organizations that formed the current study foundation. To complete this second objective, the participants were asked to evaluate each KM activity from the framework for every project they were involved in and to state if the KM activity was performed in an explicit manner. I have selected three projects (enterprise resource planning modules) from each organization investigated. As summarized in Figure 3, the evaluation showed that the first organization presents a higher level of KM activities performed in an explicit manner compared to the second organization investigated. Yet, I revealed that in both organizations the knowledge adaptation activity it is not conducted in an explicit manner.

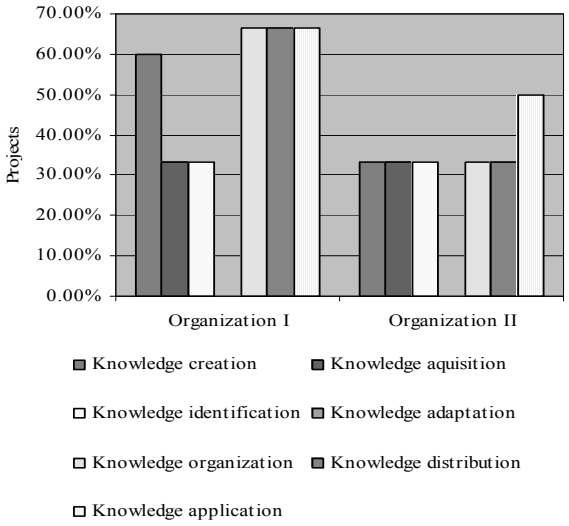

Fig. 3 The percentages of projects in which KM activities are performed in an explicit manner.

The participants recognized several activities that promote the development of tacit knowledge but they did not provided any input when asked about ways (that they could embrace easily) to diminish it. Some of them stated that they expect the organization's management to engage in KM practices or to sustain the implementation of a KMS in order to address these problems.

The results from evaluating the level of effort the participants invested on KM activities in SE are revealed in Figure 4.

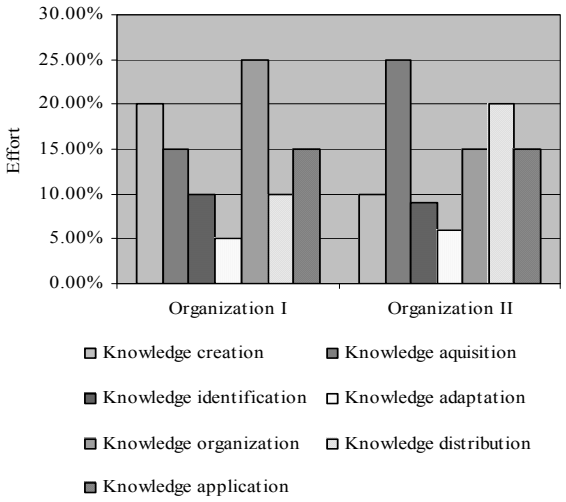

Fig. 4 The average level of effort invested on KM activities in the two studied organizations.

Employees from both organizations show almost no interest in the knowledge adaptation activity. In the first organization knowledge creation and organization are rated the highest while in the second organization the knowledge acquisition and distribution.

5.3 KM enablers investigation

In this study the participants rated the importance and applicability of each enabler of the KM process. The results are showed in Figure 5 and Figure 6.

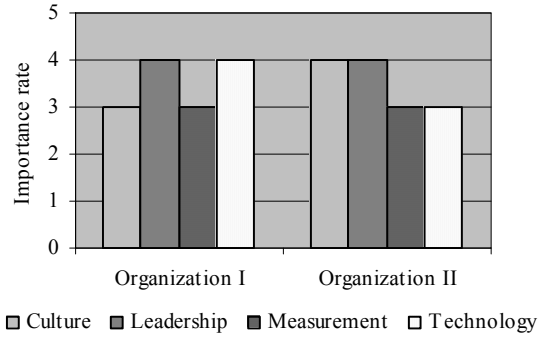

Fig. 5 The average rate of importance for each KM enabler.

The participants rated the applicability of each KM enabler in accordance with the projects they were involved in. While in the first organization, leadership and technology are the most important KM enablers in the second organization are culture and leadership.

When rating the level of applicability, leadership and technology have the most significant impact upon KM in SE for the first organization while for the second organization the culture and technology have the most significant impact. Measurement is the KM enabler with the lowest rate of applicability in both organizations, which could indicate that employees do not know proper measures that could help them make a correct evaluation of the knowledge they possess.

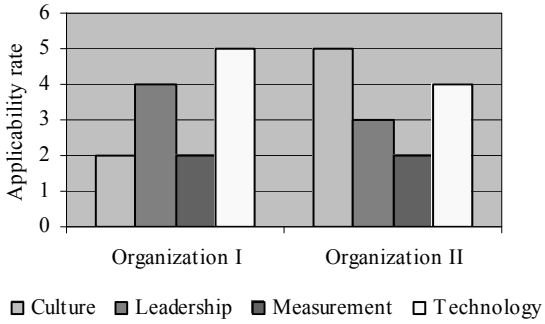

Fig. 6 The average rate of applicability for each KM enabler.

6 Discussion

In summary, findings were similar with previous studies that showed SE knowledge is highly tacit in nature, (much of which cannot be articulated well or be put in explicit format) and highly contextual in nature, which calls for focused applicability [13].

In Romanian organizations knowledge creation and knowledge acquisition are processes that take place usually in team meetings where different new ideas are presented. Knowledge is identified by individuals and then presented and discussed in team meetings. Also, a large percentage of participants need knowledge to be applicable in multiple situations, but as I can see little effort was assigned to knowledge adaptation in all the analyzed projects of both organizations. Although in the first organization the employees had a knowledge storage tool, the results of the study show that they still retained a significant amount of tacit knowledge from projects they had previously worked on. This is indication that either the knowledge storage tool was inappropriate or little effort was made to complete knowledge organization. Knowledge distribution is mainly done at the direct request of colleagues (because that means that the colleagues will help with future work) or management (for reducing the risk of having important knowledge residing in a few individuals only). The considerably amount of effort for knowledge application (as presented in Figure 3) emphasizes the important role this KM activity plays in SE.

In conclusion the knowledge acquisition and organization are the most prominent of all KM activities.

In terms of KM enablers, in both Romanian organizations, the participants considered that management was responsible for defining strategies that link KM and organizational management. Also, some participants manifested a series of doubts concerning the KMS capability of delivering knowledge both generic and specific in an easily accessible format, while others deplore the lack of KMS in their organization. When asked about what has been done in their organization about encouraging knowledge sharing, promoting open climate for the free flow of ideas, the developers stated that a series of meetings and presentation of topic of interest are held frequently in their organizations. Also the developers stated that they would like to receive rewards for these "extra" contributions, while others argued that they do not have enough time to manage their own knowledge and suggested that a KMS that could help them address this problem.

The study also revealed that both companies invested limited resources in developing appropriate measures for evaluating the impact of KM in SE. Developers considered that it is very difficult to evaluate or measure the level of their knowledge since most of that knowledge is tacit. Also even if a considerable effort is to be made in order to convert tacit knowledge in explicit knowledge, they fear that a correct evaluation of person knowledge can not be done just by counting the number of postings in a KMS database or other similar metrics.

7 Conclusions

Although it is based on a limited number of studied organizations and may not reflect the practices of organizations in all Romanian software industry this investigation provides a preliminary understanding of the implementation of KM practices in SE in Romanian organizations. This study shows that the current state of KM practice implementation in Romanian software development organizations is embryonic. Yet I noticed that efforts are made in order overcome this problem.

References:

[1] Alavi M., Leidner D., Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues, *MIS Quarterly*, Vol. 25, No. 1, 2001, pp. 107-136.

[2] Aurum A., Daneshgar F., Ward, J., Investigating Knowledge Management practices in software development organizations - An Australian experience, *Information and Software Technology*, Vol. 50, No. 6, 2008, pp. 511-533.

[3] Basili V., Seaman C., The Experience factory organization, *IEEE Software*, May-June 2002, pp. 30-31.

[4] Basili, V. R., Lindvall, M., Costa, P., Implementing the Experience Factory Concepts as a Set of Experience Bases, Knowledge Systems Institute, *Proceedings of the 13th International Conference on Software Engineering & Knowledge Engineering*, 2001, pp. 102-109.

[5] Birk A., Surmann D., Althoff, K., Applications of Knowledge Acquisition in Experimental Software Engineering, *Proceedings of the 11th European Workshop on Knowledge Acquisition, Modelling, and Management*, 1999, pp. 67-84.

[6] Bjørnson F., Dingsøyr T., Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used, *Information and Software Technology*, Vol. 50, No. 11, 2008, pp. 1055-1068.

[7] Bojnord N., Afrazeh A., Knowledge management in project phases, *Proceedings of the 5th WSEAS International Conference on Software Engineering, Parallel and Distributed Systems*, Madrid, Spain, February 15-17, 2006, pp.67-72.

[8] Bologa R., Lupu R., Accelerating the Sharing of Knowledge in Order To Speed Up the Process of Enlarging Software Development Teams - a Practical Example, *Proceedings of the 6th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, Corfu Island, Greece, February 16-19, 2007.

[9] Cai K., Card D., An analysis of research topics in software engineering – 2006, *Journal of Systems and Software*, Vol. 81, No. 6, 2008, pp. 1051-1058.

[10] Corbin R., Dunbar C., Zhu Q., A three-tier knowledge management scheme for software engineering support and innovation, *Journal of Systems and Software*, Vol. 80, No. 9, 2007, pp. 1494-1505.

[11] Cruz A.C., Ricardo de Almeida Falbo N., Knowledge Management in Software Engineering Environments, *Proceedings of the*

16th Brazilian Symposium on Software Engineering, 2002, pp. 238-253.

[12] Davenport T. H., Working Knowledge, Harvard Business School Press, 1998.

[13] Desouza K., Barriers to effective use of knowledge management systems in software engineering, *Communications of ACM*, Vol. 46, No.1, 2003, pp. 99-101.

[14] Dingsøyr T., Conradi R., Knowledge Management Systems as a Feedback Mechanism in Software Development Processes: A Search for Success Criteria, a revised and extended version of: Conradi R. and Dingsøyr T., Software experience base: a consolidated evaluation and status report, *Proceedings of the Second International Conference on Product Focused Software Process Improvement,* Finland, Vol. 1840, 2000, pp. 391-406.

[15] Fertalj K., Vučetić N., Challenge of Knowledge Management: From Theory to Practice, *Proceedings of the 4th WSEAS/IASME International Conference on System Science and Simulation in Engineering,* Tenerife, Spain, December 16-18, 2005, pp. 204-210.

[16] Hansen M.T., Nohria N., Tierney T., What's your strategy for managing knowledge?, *Harvard Business Review*, March–April 1999, pp. 106–116.

[17] Hellström T., Malmquist H., Mikaelsson J., Descentralizing knowledge: managing knowledge work in a software engineering firm, *Journal of High Technology Management Research* Vol. 12, No. 3, 2000, pp. 25–38.

[18] Johansson C., Hall P., M. Coquard M., Talk to Paula and Peter – they are experienced, *Proceedings of the Workshop of Learning Software Organizations*, Kaiserslautern, Germany, 1999, pp. 69–76.

[19] Kneuper R., Handbook of Software Engineering and Knowledge Engineering, Vol. II, World Scientific Publishing, 2002.

[20] Komi-Sirvio S., Mantyniemi A., Seppanen V., Towards a practical solution for capturing knowledge for software projects, *IEEE Software*, Vol. 19. No. 3 2002, pp. 60–62.

[21] Mamcenko J., Beleviciute I., Data Mining for Knowledge Management in Technology Enhanced Learning, *Proceedings of the 6th WSEAS International Conference on Applications of Electrical Engineering*, Istanbul, Turkey, May 27-29, 2007, pp. 115-119.

[22] Mathiassen L., Pourkomeylian P., Managing knowledge in a software organization, *Journal of Knowledge Management*, Vol. 7. No2, 2003, pp. 63-80.

[23] McGinnis T.C., Huang Z., Rethinking ERP success: A new perspective from knowledge management and continuous improvement, *Information & Management,* 2007.

[24] Nonaka W.E., Konno N., The concept of 'Ba': building a foundation for knowledge creation, *California Management Review*, Vol. 40, No. 3, 1998, pp. 40-54.

[25] Rodríguez-Elias O., Martínez-García A., Vizcaíno A., Favela J., Piattini M., A framework to analyze information systems as knowledge flow facilitators, *Information and Software Technology*, Vol. 50, No. 6, 2008, pp. 481-498.

[26] Romer P., Beyond the knowledge worker, *Worldlink*, 1995.

[27] Ruggles R., Knowledge management tools, Butterworth-Heinemann, 1997.

[28] Ruggles R., The state of the notion: knowledge management in practice, *California Management Review*, Vol. 40, No. 3, 1998, pp. 80–89.

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

[30] Salim J., Yahya Y., Othman M., Rashid N., The Use of Holistic Approach to Knowledge Management Initiative in Managing Information in Higher Learning Institution: A Perspective, 6th *WSEAS International Conference on E-ACTIVITIES*, Tenerife, Spain, December 14-16, 2007, pp.347-352.

[31] Scorţa I., The role of tacit knowledge management in ERP systems implementation, *Proceedings of The 3rd International Conference Knowledge Management - Projects, Systems and Technologies*, Bucureşti, Romania, 23 octombrie 2008, *Economy Informatics*, Vol. 8, No. 1-4, 2008, pp. 78-82.

[32] Sedera D., Gable G., Chan T., Knowledge management for ERP success, *Proceedings of The 7th Pacific Asia Conference on Information Systems*, 2003.

[33] Tiwana A., An empirical study of the effect of knowledge integration on software development performance, *Information and Software Technology*, Vol. 46, No. 13, 2004, pp. 899–906.

[34] Wiig K., Comprehensive Knowledge Management, Working paper KRI, Knowledge Research Institute, Inc., 1999.