



TACIT REQUIREMENTS ELICITATION FRAMEWORK

Bashir M. Mohamed Geblawi Nurlida Basir and Rosalina Abdul Salam
Fakulti Sains dan Teknologi, Universiti Sains Islam Malaysia (USIM), Nilai, Negeri Sembilan
E-Mail: bashir_104@yahoo.com

ABSTRACT

Requirements engineering (RE) is an integral and fundamental part of software development life cycle (SDLC). The most cited reason for software failure is the inability to fully capture and implement the exact user requirements in a timely, operationally and financially expected manner. Requirements can be categorized into two types; explicit and tacit. Explicit requirements usually expressed clearly and well-identified requirements in which the system must perform. In contrast, tacit requirements are considered as hidden or embedded requirements that incapability of getting such may caused a software failure. The importance of tacit requirements in requirements engineering (RE) has been acknowledged widely as characteristics for developing projects in software engineering (SE). Therefore, the aim of this paper is to construct a framework for tacit requirement elicitation process. The framework is design based on the integration of the tacit knowledge elicitation process of Nonaka and Takeuchi model with the tacit requirements elicitation process.

Keywords: tacit requirements, software engineering (SE), requirements engineering (RE), knowledge management, tacit knowledge.

INTRODUCTION

Software development life cycle (SDLC) is a human oriented practice, wherein different actors collaborate in identifying and defining a problem, which is also complemented with further assessment and implementation of pertinent solution. Irrespective of the SDLC approaches, the foundation for every SDLC is the requirement engineering (RE) phase. Requirements engineering is the most important area of software engineering and possibly in of the entire software life cycle as the failure in this stage may incur high cost to the project [1]. Reference [2] stated that “everything else in software development depends on the requirements. If you cannot get stable requirements, you cannot get a predictable plan”. As the most important phase in software engineering (SE), RE consists of sub-activities such as feasibility studies, elicitation, specification and validation of software requirement, which control the real-life implications of the software in terms of functionality and limitations [3, 4, 5]. In order to ensure the successful of the software development, requirements elicitation process is applied with an intensive method for eliciting both tacit and explicit requirements. Despite clear demarcation on the types of requirement and commitment towards careful elicitation of requirement, most software development projects still end with lack of satisfaction [1, 6, 7]. Since the primary source of requirements is users’ needs, perceptions and expectation, which are mostly held in their brain, thus, the requirement engineer’s ability in getting such requirements explicit is a major determinant of success, which certainly requires suitable requirements elicitation process that can help in gathering a complete and clear requirements either its explicit or tacit requirements. Explicit requirements usually expressed clearly and well-identified requirements in which the system must perform. In contrast, tacit requirements are considered as hidden or embedded requirements that incapability of getting such may caused a software failure [8]. Here, we focus on the process of tacit requirements

elicitation. The aim of this work is to construct a framework to elicit tacit requirements. The framework is design based on the integration of the tacit knowledge elicitation process of Nonaka and Takeuchi model [9] with the tacit requirements elicitation process. Figure-1 shows the general research methodology of conducting the research.

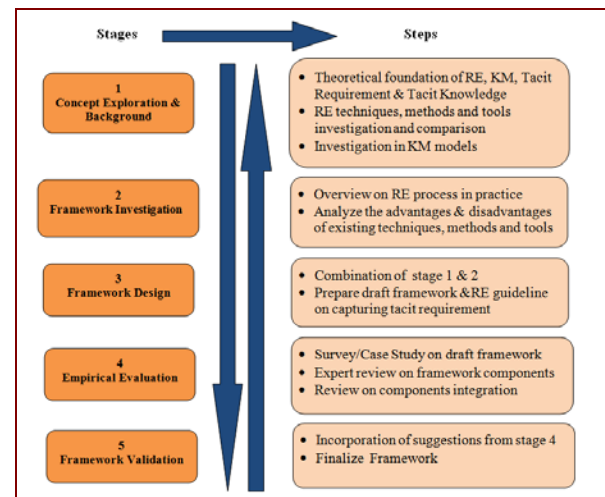


Figure-1. Steps of conducting research.

REQUIREMENTS ELICITATION

Requirements elicitation is the process of gathering information of the proposed systems and distilling the user and system requirements. It specifies the system boundaries and identifies the functional behavioral properties of a system [10]. Requirements elicitation is the specific processes of gathering, determining, extracting, or exposing software requirements [11]. Requirements can be categorized into two types; explicit and tacit. Explicit requirements usually



expressed clearly and well-identified requirements in which the system must perform.

In contrast, tacit requirements are considered as hidden or embedded requirements that incapability of getting such caused serious problems [8]. Concerning on software quality, both requirements explicit and tacit should be included in the process of software development, if the explicit requirements included and the tacit requirements neglected, then the quality not guaranteed to meet users' satisfaction [12].

TACIT REQUIREMENTS

Tacit requirements are defined as (1) hard to express, convert, communicate and share; (2) often related to application domain; (3) describe as users' tacit knowledge; (4) are experiential knowledge which developing team accumulates step by step in practice during a long period of time; (5) are hard to encode and articulate; (6) can be expressed hazily and crudely" [13]. Many techniques, tools and methods for requirement elicitation don't support the elicitation of tacit requirements in direct way or they may act as complementary to each other. All in all, tacit requirements elicitation process without effective knowledge conversion is strongly determined to fail [13]. Project failure is mostly an outcome of the challenges associated with the clear articulation of user's personal knowledge and expectation in the requirement specifications [14, 15].

DEFINITION OF TACIT REQUIREMENTS

The characteristic of "tacit requirements" is defined as [16] "Tacit Requirements are inexplicit requirements that are not directly expressed or captured but are essential to meet System's goal. They are something that is assumed to be "there". One example adopted from [16] illustrates the missed "tacit requirements" which had a negative impact on their project as showed below. **Explicit requirement 1:** A Publisher should be able to create an Article, send it for approval and finally publish the Article on the portal. **Explicit Requirement 2:** A Portal user should be able to search Articles published on the Portal.

After finishing the system development, it was realized that the system was created to meet both requirements where the author could create and publish an article and the end user was able to search the articles by using the search functionality. But unfortunately, the search displayed those articles which were in draft or still under approval. **Tacit requirement** of this example supposed not to display the articles which still in draft state or under approval.

KNOWLEDGE MANAGEMENT (KM)

Knowledge management in general is about the efforts that have to be applied to capture, store and deploy knowledge using IT to facilitate business process in the organization [17]. The central idea of KM is that the work efforts to create, codify and share valuable knowledge to the organization [18]. Reference [19] classified knowledge

management as "a systemic and organizationally specified process for acquiring, organizing, and communicating both tacit and explicit knowledge of employees so that other employees may make use of it to be more effective and productive in their work". Besides, [20,21] declared that, the main key point of knowledge management is to capture tacit knowledge which is considered as individual knowledge and convert it to explicit knowledge, in order to complete the rest of the stages of knowledge management. Also, the most beneficiary from knowledge management is software engineering, because it is based on the use of knowledge-intensive [22].

Despite of existence of several models for knowledge creation in knowledge management, yet Nonaka's knowledge management model SECI is one of the top models that used to achieve its task and it approves that it is a correct model of representation of knowledge conversion in the process of introducing organizational changes [23]. This model is widely accepted as knowledge conversion model that was explored by [9], in which, its process encompasses four modes as socialization, externalization, combination and internalization (SECI) each mode of knowledge transfer operates differently as shown in Figure-2 [24, 25, 26]. Through socialization mode tacit knowledge is transferring among individuals by imitation, observation, sharing experience, working together as team and direct interaction, but externalization mode aims to be as dialogue, collective reflection and analogy involves transferring tacit knowledge to documents or procedures. One purpose is to articulate self-knowledge in images, ideas and words, the second purpose is to eliciting and translating tacit knowledge to understandable forms (explicit knowledge),and therefore, dialogue is very important means for both. Continuously, combination mode reconfigures objects of explicit knowledge through sorting, adding, combining, and classifying processes which considered as explicit-explicit, once the knowledge became explicit it can be transferred to process which Nonaka calls combination. Lastly, internalization operates to translate explicit knowledge to tacit knowledge that possessed by individual, also, internalization is the process of understanding and absorbing explicit knowledge to tacit knowledge held by the individual and internalization is largely considered as experiential.

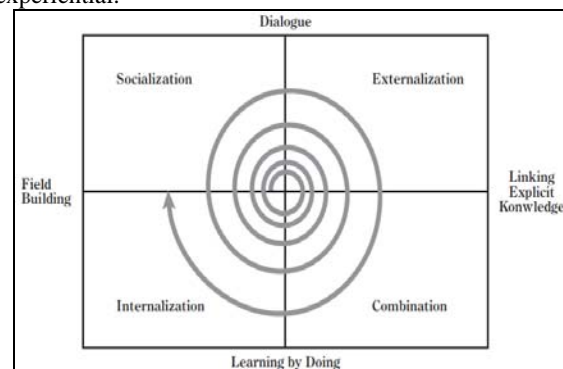


Figure-2. The process of SECI model [9].



Based on the intensive literature review, tacit requirements have similarities with tacit knowledge, where both are considered as "hidden known" and difficult to articulate because they reside in human' brains and they need an effective method in order to elicit both in perfect way. In view, we can see the similarities between the process of capturing tacit knowledge in organizations, and the process of eliciting tacit requirements in software engineering. This paper focuses on the incorporation of knowledge management in requirements engineering by proposing framework to enhance the process of eliciting tacit requirements in the field of software engineering. Table-1 illustrates the similarities between tacit knowledge and tacit requirements.

Table-1. Tacit requirements and tacit knowledge similarities.

Tacit requirements	Tacit knowledge
<ul style="list-style-type: none"> • Tacit requirements are hard to express, convert, communicate and share. • Tacit requirements are often related to application domain • Tacit requirements are often users' tacit knowledge. • Tacit requirements are experiential knowledge which developing team accumulates step by step in practice during a long period of time. • Tacit requirements are hard to encode and articulate [13]. 	<ul style="list-style-type: none"> • Tacit knowledge is difficult to articulate. • Tacit knowledge is subjective and personal knowledge [27]. • Tacit knowledge is "a non-linguistic, non-numerical form of knowledge" [28]. • Tacit knowledge or self-knowledge is based on experience and cannot be expressed in words, sentences, or numbers.

Therefore, incorporation of knowledge management into requirements engineering for the purpose of tacit requirements elicitation takes place in this study as the way of the incorporation illustrated in Figure-3. Furthermore, the incorporation meant that, the mechanism of Nonaka's model will be adopted in the process of the proposed framework with the exception of the last mode of Nonaka's model which is "Internalization" because in its process it converts explicit to tacit, and this framework does not intend to represent that, it stops in particular stage where the tacit requirements became explicit.

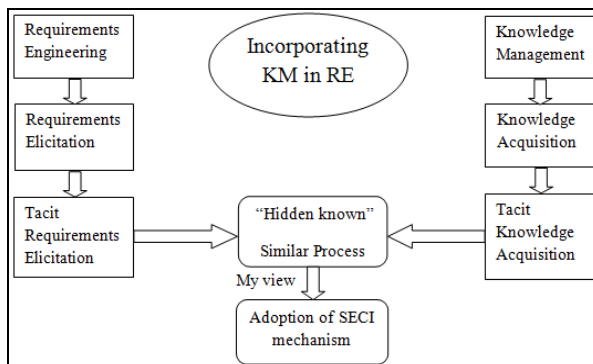


Figure-3. The way of incorporating KM in RE.

TACIT REQUIREMENT ELICITATION FRAMEWORK

Dealing with the process of tacit requirements elicitation is much like the process of tacit knowledge elicitation. The process of tacit knowledge elicitation is normally refers to knowledge management (KM). Since the process of eliciting tacit knowledge and tacit requirements are somehow similar as both dealing with human, we adapted the process of knowledge management in requirement engineering to improve the tacit requirement elicitation process. We illustrate our framework as shown in Figure-4. Our framework considered the human mind, behavior and perception throughout the elicitation process.

This framework has three stages that resemble the process of Nonaka and Takeuchi (1995) model. As mentioned above, we exclude the final stage of Nonaka and Takeuchi (1995) model [9] as our objective is to make the tacit requirements explicit. The criterion which was adopted for the selection of these particular stages, is that the meaning of each stage allows to perform specific task, for instance, feasibility study is to perceive and understand the environment of the desired project and to share experience (tacit to tacit), requirements definition stage is to define the requirements based on the understanding of the feasibility study that its necessity to reflect the performance of the software in matter of what the system must do (tacit to explicit), and requirements prioritization stage is to determine the importance level of the requirements that have been summarized during the definition stage (explicit to explicit).

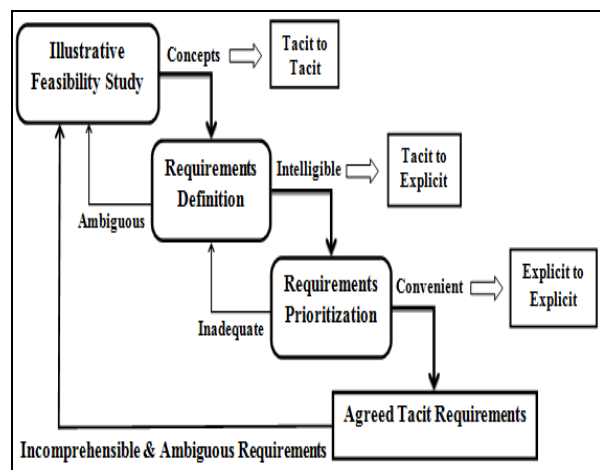


Figure-4. Structure of the framework.

ILLUSTRATIVE FEASIBILITY STUDY STAGE

The first stage provides some process of understanding the project environment and to share tacit knowledge among users, stakeholders and requirements engineers (analysts) which allows the establishment of the objectives and goals through observing users, gathering users and meet them individually or in groups. Based on some authors' recommendations of combining several



techniques as mentioned earlier, ethnography, focus group and interviews are initial process of understanding and perceiving the concepts of what must be accomplished in the other stages of the elicitation process.

For instance, ethnography technique provides observation that allows requirements engineers to understand the behavior and the patterns of culture context and social factors where the desired project is going to be deployed. Also, focus group gathers users to share experience that generates new suggestions which support the concepts obtained from using ethnography technique. While interviews act as provider of verbal exchange of information which derived from face to face discussion and brainstorming. Moreover, interviews technique reaches those users who were not able to attend focus group meetings. By doing that, opportunities to discuss non-functional requirements such as business objectives, cost, timescale, reliability and environmental constraints will take a place. Figure-4 illustrate the first stage of the framework.

Here, the first stage of our framework is considered as same as socialization mode of Nonaka's model that generate tacit-tacit knowledge. Typically, the different problems domains normally require different techniques [29] and beside that we will reach groups of people where requirements engineers understand more about the stakeholders' world, while stakeholders learn more about what is technically possible and feasible. Furthermore, this stage also aims to collect sufficient information to identify the problem boundaries (What problem?), understand the problem domain (Where is the problem?), identify the stakeholders (Whose problem?), identify the stakeholders' goals (Why is problem interesting?), visualize some scenarios (How might software system help?), identify constraints (When does problem need solving?) and to identify feasibility and risk (What might prevent us solving it?). As result, capturing tacit requirements is a task that comes after understanding users need, analyzing culture context and social factors. Lastly, instead of seeking for methods or techniques that would help in the process of absorbing or visualize a project environment, this stage was designed to allow the analyst to easily understand the environment. Table-2 summarise the process of stage 1 of the framework.

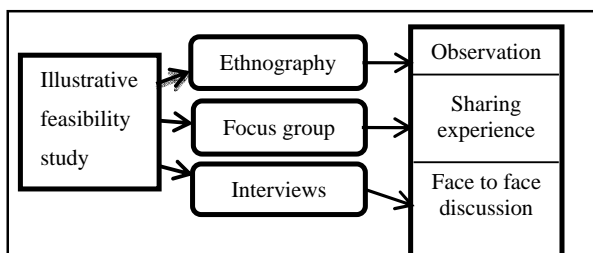


Figure-5. Illustrative feasibility study process.

Table-2. Summarization of stage 1.

Feasibility Study	
Input	Domain information and cumulative knowledge possessed by requirements engineers
Process	1. Record information for project aspects; 2. Organize these aspects into points relatedly; 3. Raise these aspects with users; and 4. Write down their feedback and suggestions properly.
Output	Raw requirements as plain text in significant quantities

REQUIREMENTS DEFINITION STAGE

After the requirements engineers (analysts) established the initial understanding of the project through the first stage of the framework, then during this stage there will be some process to define requirements which pointed out as raw requirements. This stage is based on dialogue which required meta-cognitive tools such as probing, metaphors and storytelling that their nature is to discuss variety of issues and viewpoints (solutions). Also, requirements definition stage is the process of articulating tacit requirements into explicit concepts which means making "unknown known". Besides that, negotiation process will take place for requirements definition stage because in the process of identifying the accurate requirements to develop such a system, conflicts are common, since stakeholders often pursue mismatching goals. Getting agreements among stakeholders who have different concerns, responsibilities, and priorities is quite challenging. Therefore, negotiation is useful to handle the conflicts and to resolve disagreement between the stakeholders. A part of achieving agreement, the requirements are believed to be improved in quality. Figure-6 shows the process of requirements definition stage. Table-3 summarise the process of stage 2 of the framework.

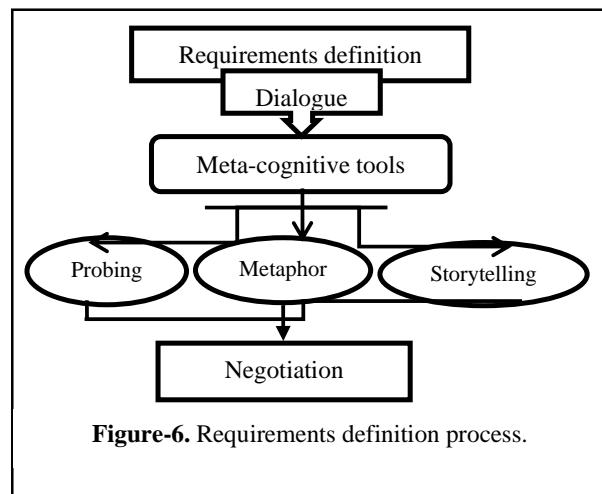


Figure-6. Requirements definition process.

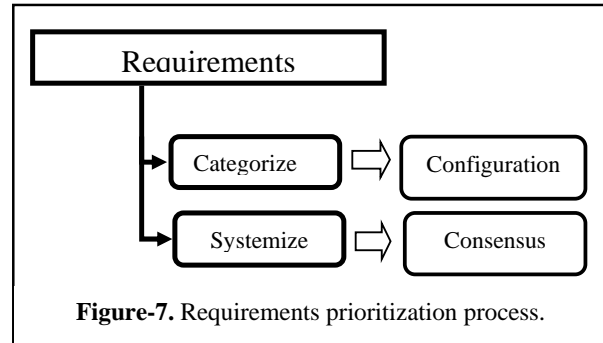
**Table-3.** Summarization of stage 2.

Requirement Definition	
Input	Those raw requirements as plain text in significant quantities derived from feasibility study stage.
Process	1. Deep elaborations of those aspects till infer the hidden requirements
Output	List of tacit requirements which are very intelligible and refined.

REQUIREMENTS PRIORITIZATION STAGE

Requirements prioritization is based on doing which punctuated by concrete actions, the stakeholders are asked to classify priorities of their requirements and then confer conflicts in priority. The goal of prioritization is to provide opportunities for drawing the process of the project and to determine the priorities as hierarchy. The process of prioritization of requirements is often a negotiation process among users, stakeholders and developers that based on iterative and incremental basis. For instance, the involved participants have to concentrate on those requirements that are might be used in the current or next release and those requirements that are in the intermediate priority which is most likely required negotiation. Therefore, the prioritization of requirements requires only two main steps to accomplish this stage, for instance, categorization and systemization. Categorization step is the process of identifying requirements in terms of high, medium or low from the perspective of stakeholders, and these processes would be done through the use of the scaling category as described in Table-5. After that the configuration will take place to emphasize the final version of the requirements that passed through the two previous stages. As for the process of requirements systematization, requirements will be included through the use of a scale that determines the importance of the requirement from the point of essential, conditional or optional (as shown in Table-5), including consensus that the opinions of the participants will be taking once again on the requirements that have been measured in the previous phase to increase the awareness among them. Figure-6 depicted the process of the prioritization stage.

The two steps of the prioritization stage are almost carrying the same meaning of gaining an agreement among all participants regarding the level of importance of the requirements. Also, they aim to structure the requirements and prepare the execution of the prioritization. It is important at this stage, that the two steps must be subjected to two typical three-level scales as shown in Table-5, to achieve the intended results and, to determine priorities by stakeholders. Basically, all scales are essential and subjective, so that everyone involved in requirements prioritization must agree on the each level of the scale that been used to facilitate the prioritization [30]. Table-4 summarise the process of stage 3 of the framework.

**Figure-7.** Requirements prioritization process.**Table-4.** Summarization of stage 3.

Requirement definition	
Input	List of tacit requirements which are very intelligible and refined.
Process	1. post the requirements to obtain the actual desire directly from targeted users
Output	Hierarchy of requirements based on the users' desire

Table-5. Two typical three-level scales.

Scale	Description
High	A mission critical requirement; required for next release
Medium	Supports necessary system operations; required eventually but could wait until a later release if necessary
Low	A functional or quality enhancement; would be nice to have someday if resources permit
Essential	The product is not acceptable unless these requirements are satisfied
Conditional	Would enhance the product, but the product is not unacceptable if absent
Optional	Functions that may or may not be worthwhile

CONCLUSION AND FUTURE WORK

As we have seen, tacit requirements contributes as major pillar of software development which eventually fulfills the needs of the stakeholders and avoids any possibility of failure. Unfortunately, as it mentioned above tacit requirements are difficult to be articulated, elicited codified. Meanwhile there is no sufficient or an ideal way to accomplish the process of tacit requirements elicitation, and therefore, it is imperative to put more efforts to reach a certain technique through which the process of tacit requirements elicitation performed in the required manner. Here, this work presented a framework to elicit tacit requirements by adopting the knowledge management framework i.e., Nonaka and Takeuchi (1995) model [9]. Our framework focuses on making the tacit requirements explicit. For future works we will evaluate the framework by expert reviews questionnaire and case study.



REFERENCES

- [1] Aybuke A. and Wohlin C. 2003. The fundamental nature of requirements engineering activities as a decision-making process. *Information and Software Technology*. pp. 954-954.
- [2] Fowler M. 2005. The New Methodology. <www.martinfowler.com>.
- [3] Mohamed A. H. 2010. Facilitating Tacit-Knowledge Acquisition within Requirements Engineering. In: 10th Wseas International Conference on Applied Computer Science (ACS '10), Iwate Prefectural University, Japan. pp. 27-32.
- [4] Sommerville I. 2011. *Software Engineering*. Pearson Education.
- [5] Bourque P. and Dupuis R. 2004. *Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society.
- [6] Hickey A. and Davis A. 2003. Requirements elicitation and elicitation technique selection: A model for two knowledge intensive software development processes. *Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03)*.
- [7] Cheng B.H.C. and Atlee J.M. 2007. Research directions in requirements engineering. *Future of Software Engineering 2007 (FOSE)*, in ICSE. Minneapolis, Minnesota, IEEE Computer Society.
- [8] Onyeka E. 2013. A Process Framework for Managing Implicit Requirements Using Analogy-Based Reasoning. *Research Challenges in Information Science (RCIS), 2013 IEEE Seventh International Conference*.
- [9] Nonaka I. and Takeuchi H. 1995. *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation?* Oxford University Press, New York.
- [10] Sajjad U. and Hanif M. Q. 2010. Issues and challenges of requirement elicitation in large web projects. School of computing, Blekinge institute of technology ronney Sweden.
- [11] Saiedian H. and Dale R. 2000. Requirements engineering: making the connection between the software developer and customer. *Information and Software Technology*. 42(6): 419-428.
- [12] Ahamed R. 2010. An Integrated and Comprehensive Approach to Software Quality. *International Journal of Engineering Science and Technology*. 2(2): 59-66.
- [13] Jiang-ping W. and Wan Dan H. 2009. Knowledge Conversion in Software Requirement Elicitation. The 1st International Conference on Information Science and Engineering (ICISE2009).
- [14] Friedrich W. R. and Poll J. A.V.D. 2007. Towards a Methodology to Elicit Tacit Domain Knowledge from Users. *Interdisciplinary Journal of Information, Knowledge, and Management*. Vol. 2.
- [15] Chikh A. 2011. A Knowledge Management Framework in Software Requirements Engineering Based on the SECI Model. *Journal of Software Engineering and Applications*. Vol. 4.
- [16] R Jha. 2009. Gathering Implicit Requirements. (10-06-2009), <http://alturl.com/ocyb5>.
- [17] Preece A., Flett A., Sleeman D., Curry D., Meany N. and Perry P. 2001. Better knowledge management through knowledge engineering. *Intelligent Systems, IEEE*. 16(1): 36-43.
- [18] Ghani S. R. 2009. Knowledge management: tools and techniques. *DESIDOC Journal of Library and Information Technology*. 29(6): 33-38.
- [19] Alavi M. and Leidner D. E. 1999. Knowledge management systems: issues, challenges, and benefits. *Communications of the AIS*, 1(2es), 1.
- [20] Sagsan M. 2006. A new life cycle model for processing of knowledge management. In *Study Presented at 2nd International Conference on Business, Management and Economics*. pp. 1-9.
- [21] Sagsan M. and Zorlu K. 2010. An Empirical Test of the Knowledge Management Life Cycle Model at a Turkish Petroleum Oil Industry Firm.
- [22] Andrade J., Ares J., Garcia R., Rodriguez S. and Suarez S. 2006. A reference model for knowledge management in software engineering. *Engineering Letters*. 13(2): 159-166.
- [23] Grzybowska K. and Gajdzik B. 2013. SECI model and facilitation in change management in metallurgical enterprise. *Metalurgija*. 52(2): 275-278.
- [24] Lindvall M., Rus I. and Sinha S. S. 2002. Technology Support for Knowledge Management. Presented at Fourth Workshop on Learning Software organizations (LSO), Chicago, Illinois, USA.
- [25] Hildreth P. M. and Kimble C. 2002. The duality of knowledge. *Information Research*. 8(1).



- [26] Spencer B. 1997. Organizational Knowledge Creation. Knowledge Advantage Conference held November 11-12.
- [27] Janz B. D. and Prasarnphanich P. 2005. Understanding knowledge creation, transfer, and application: investigating cooperative, autonomous systems development teams. In System Sciences. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on (pp. 248a-248a). IEEE.
- [28] Gourlay S. 2002. Tacit knowledge, tacit knowing or behaving? Proceedings of the Third European Conference on Organizational Knowledge, Learning, and Capabilities.
- [29] Jiang L., Eberlein A. and Far B. H. 2005. Combining requirements engineering techniques-Theory and case study. In Engineering of Computer-Based Systems. ECBS'05. 12th IEEE International Conference and Workshops on the (pp. 105-112). IEEE.
- [30] Karl E. Wiegers. 1999. First Thing First: Prioritizing Requirements. Software Development.