International Journal of Information Management 37 (2017) 155–161

Contents lists available at ScienceDirect

International Journal of Information Management

journal homepage: www.elsevier.com/locate/ijinfomgt

Research Note

Knowledge meaning and management in requirements engineering

Edgar Serna M.[a,*], Oscar Bachiller S.[b], Alexei Serna A.[a]

[a] Corporación Universitaria Remington, Medellín, Colombia
[b] Universidad de Cundinamarca, Fusagasugá, Colombia

ARTICLE INFO

Article history:
Received 30 September 2016
Accepted 9 January 2017

Keywords:
Knowledge management
Knowledge meaning
Software engineering
Requirements elicitation

ABSTRACT

It is traditionally assumed that requirements specification, as a product of requirements engineering, has a high impact on the ensuing software development stages. Therefore, the knowledge management used to construct the requirements specification should be performed in a structured manner to discover, analyze and understand the data–information–knowledge chain, both tacit and explicit, that the interested parties possess. In this article, the results of a literature review are presented, seeking to answer the following questions: (1) What is the meaning of knowledge in requirements engineering? (2) What approaches are proposed to manage knowledge in requirements engineering? (3) Can the efficiency and the efficacy of knowledge management models be evidenced in requirements engineering? Thirty-six works were chosen for analysis out of a total 83 found in our search. The analysis showed that (1) knowledge has a central significance at this stage, but the authors have yet to agree on the best methods to impart and apply that knowledge; (2) no general framework has emerged as a validated approach to manage knowledge for requirements engineering; and (3) the evaluation marks for model efficiency and efficacy are low, consisting mostly of personal interpretations.

1. Introduction

In the software development community, it is generally accepted that requirements engineering is the lifecycle stage with the highest influence on the quality of the final product. However, because the applications have become more complex, while traditional models continued to be applied to manage the knowledge generated in this field, it has become difficult to attain a quick and objective understanding of the needs of the interested parties. To contribute to the search for solutions, various authors have been proposing solutions to manage this knowledge. However, to date, they have not found a solution that is widely accepted and recognized by the community. Therefore, knowledge of the current state of such investigations and past proposals is required, if the goal is to contribute to the improvement of software quality, to ensure that formative processes include them and to include any relevant curricular content (Serna & Serna, 2016a). Given the short-term goal of the software community of improving the quality and therefore the reliability and security of their products, it is required that software engineers be trained on new knowledge management proposals in requirements engineering (Terstine, 2015).

The need described above has not gone unnoticed by researchers, who have even been motivated to carry out analyses of how to manage knowledge in requirements engineering. Jurisica, Mylopoulos, and Yu (1999) state that knowledge management, in the phase of requirements engineering, is concerned with its representation, organization, acquisition, creation, use and evolution into its multiple forms. However, they also state that improving the understanding of how knowledge is used by individuals, groups and organizations is necessary. Although their proposal is interesting and has been validated, it has an excessively broad coverage, such that adapting it for analyzing knowledge management in the phase of requirements engineering requires too much work. Bresciani, Donzelli, and Forte (2003) analyzed an agents-based knowledge management framework for requirements engineering, with the goal of designing supports to capture and formalize the knowledge incorporated or extracted from the organization. It is interesting to follow the work of these authors as they apply their proposals and validate their results; however, the framework they use is based on agents, a principle in computer science that is still making its way into research.

Andreas Breiter (2004) adapted some existing models to the context of requirements engineering for knowledge management. On that basis, he derived the specific functional requirements,

* Corresponding author.
E-mail addresses: eserna@eserna.com (E. Serna M.), oscar.bachiller@gmail.com (O. Bachiller S.), alexei.serna@uniremington.edu.co (A. Serna A.).

http://dx.doi.org/10.1016/j.ijinfomgt.2017.01.005
0268-4012/

and integrated them into the system development through a participatory design process. Breiter's proposal is easily adapted for knowledge management; however, the formats proposed do not provide adequate range. Andrade et al. (2006) proposed introducing a knowledge management program that supports the software process, structured under a formalization scheme, and capable of representing, capturing and transmitting the knowledge that can be exploited in requirements engineering. Their work is among the few that demonstrate how to manage knowledge during the requirements engineering phase, although its goal is oriented toward software engineering in general.

The work by Al-Karaghouli, Taylor, and AlShawi (2008) aimed to build a theoretical framework oriented toward closing the gaps between different types of knowledge while managing the business requirements and the information flow between the interested parties. Their proposal is a practical framework that describes some techniques and derived tools, but the framework has not yet demonstrated that it can function beyond the specific areas on which it is founded; specifically, it has not been tested for requirements engineering. Schmitz (2010) worked toward providing improved media to support knowledge in the elicitation, analysis, documentation, and other operations on the requirements. He also addressed the dynamics of the requirements engineering process, considering its volatility. Although this approach is novel, it has not yet demonstrated the ability to adapt to the paradigm of object-oriented programming.

Chikh (2011) stated that in requirements engineering, the collaboration between the interested parties and the analysts must be facilitated in a manner such that knowledge management is minimized to obtain better results. They proposed a management framework based on the SECI (Gourlay, 2003) knowledge creation model, whose purpose is exploiting the tacit and explicit knowledge of the requirements within a project. The inconvenience of this framework is that it is restricted to the SECI model, which is not sufficiently flexible to be adapted to contexts such as requirements engineering. However, Schneider et al. (2013) stated that in software development, the requirements are not identified nor implemented correctly because the process depends mostly on human knowledge (tacit and explicit). To solve this problem, those authors identified the methods associated with the knowledge creation theory by Nonaka (1994) and analyzed to what extent they aid in overcoming these problems. Although it is neither obvious nor easy to apply those methods to software projects, the methods identified by those authors are applied to reduce risk in knowledge management.

This work presents the results of a literature review to determine the meaning of knowledge and how it is managed in requirements engineering. The goal is answering the research questions while simultaneously determining whether there exists a way to adopt those proposals for managing this knowledge, or if, on the contrary, it is necessary to structure a different knowledge management model for this software development stage for each application.

2. Methodology

According to Brereton, Kitchenham, Budgen, Turner, and Khalil (2007), a literature review has three primary phases: (1) planning of the review; (2) conducting the review; and (3) documenting the results. These phases and other necessary processes are summarized in the following six activities (Kitchenham, 2003; Kitchenham et al., 2009):

2.1. Research questions

Three questions were formulated for this investigation: (1) What is the meaning of knowledge in requirements engineering? (2) What approaches are proposed for managing knowledge in requirements engineering? (3) Can the efficiency and the efficacy of the knowledge management models in requirements engineering be evidenced?

2.2. Search process

The initial goal of this investigation was to identify candidate studies. For that purpose, a plan was designed to query the databases ACM, IEEE, ScienceDirect, Springer and Wiley. The search parameters included keywords such as: *knowledge management*, *requirements engineering*, *models*, *methodologies*, *knowledge types*, and *meaning of knowledge*. One of these keywords had to appear at least once in the document.

2.3. Inclusion and exclusion criteria

The primary inclusion criterion was the work relevance for answering the research questions. Therefore, criteria such as the following were considered: The work had to be an explicit investigation, occur within the 2005–2016 timeline, present a theoretical description, describe a practical application, discuss a case study in detail, present a knowledge management model or methodology, and cite works by other authors. Initially, the candidate is discarded if it does not meet at least one of these criteria.

2.4. Quality evaluation

To determine the quality of the candidates, criteria such as the following were considered: Formality and pertinence of the distribution medium; author's authority; quality of the results and data sources; degree to which the thesis was upheld; applied research process; coherence between results and conclusions; degree of acceptance (number of citations); evaluation by the community; and recognition in the industry after having tried the proposal. A value was assigned to each criterion to determine the quality.

2.5. Recompilation of the data

A matrix containing the following information was created: (1) Type: article, book chapter, book, conference presentation, other; (2) title; (3) author; (4) contribution: theoretical description, practical application, study case, model, methodology; and (5) year. A total of 83 documents were found.

2.6. Defining the data analysis

In this phase, the method of Dyba and Dingsoyr (2008) was applied to analyze a series of documents by filtering the set of primary studies to (1) identify the relevant studies, (2) exclude studies based on their titles, (3) exclude studies based on their abstracts and (4) analyze and select those that make relevant research contributions, based on the full text. Taking into account the inclusion-exclusion and evaluation criteria, 29 works were extracted from the initial sample in this analysis. Subsequently, a cross-referencing of information was performed to determine the efficiency and the efficacy of each contribution. Then, a further 14 works were discarded. After this phase, the final sample consisted of 40 documents, whose analysis is presented below.

3. Results

From the discussion and analysis of the works in the final sample, we obtained the necessary information to answer the research questions. Each of the authors individually approached the analysis, and subsequently a joint discussion was performed to answer the formulated questions.

3.1. Meaning of knowledge in requirements engineering

Knowledge is a term with different meanings in different contexts. For example, according to Webster's dictionary, knowledge is the fact or condition of knowing something with familiarity, i.e., something that has been acquired through experience or association. For Davenport and Prusak (1998), knowledge is a mixture of experiences, values, and contextual and specialized information, originating from and applied in each individual in the know, and which provides a framework for evaluating and incorporating new experiences and information. According to Biggs and Tang (2011), knowledge is a set of ideas or thoughts that people possess and utilize to make efficacious decisions. Therefore, it is specific and individual. In the organizational context, it is the sum of what is known and resides in the intelligence and capabilities of people (BusinessDictionary.com).

According to these definitions, knowledge originates and resides in people, such that it can be classified and characterized, as done by Nonaka and Takeuchi (1995). They understood tacit knowledge as that which comprises people's technical abilities and their cognitive dimensions, and explicit knowledge was that which they can communicate, distribute and transmit. Biggs and Tang (2011) classified types of knowledge as declarative-prepositional (*know what*), procedural (*know-how*), conditional (*know how*) and functional (*know when*). In any case, the characteristics of knowledge are that it is valid, useful, clear, pertinent, and it is meaningful and important within a context (Denning, 2000). In summary, knowledge is personal, and to exploit it, we have to contextualize it. The context that will be used in this work to analyze the meaning of knowledge and its implications is requirements engineering.

In this stage of software development, data and information are exchanged permanently between the interested parties and the team of analysts. This team must develop the ability to convert the data into knowledge, such that the team can acquire enough wisdom to understand, model and present a solution to the problem. At the requirements elicitation stage, various techniques are applied to compile the opinions, appreciations, interpretations, data and information that the users possess about the problem; a similar treatment is applied to the needs that the solution must fulfill (Serna, Serna, Sepúlveda, & Guevara, 2015). One of the issues in this process is that communication takes place in a natural language, which causes misunderstandings because of its natural ambiguity. This does not facilitate obtaining reliable information to understand the context and the domain of the problem to be solved.

For decades, software engineering researchers have sought ways to exploit tacit and explicit knowledge possessed by people and organizations to attain a better understanding of the problems they face. Although inconveniences exist, success has been attained in some cases where the techniques have shown success. In these cases, the teams accumulate and exploit data and information to create a map and a model for the situation and problem. This situation is commonplace during requirements engineering because often, the interested parties interpret that what they know does not contribute to the situation, or they are simply unwilling to share it. In what follows, some articles are presented where the meaning of knowledge at this stage of the software life cycle is analyzed.

Lambe (2007) distinguishes between four dimensions from which knowledge can originate: (1) the information possessed by each person; (2) what each individual has acquired with experience and is able to transmit; (3) what the group possesses regarding a problem; and (4) the culture and history of the organization. Based on these dimensions, White (2010) proposes a way to exploit the information and knowledge of the interested parties in requirements engineering by (1) capturing individual information; (2) sharing what has been learned; (3) summarizing information; and (4) integrating what has been learned with the organization's culture and history. Lambe proposes this practice as a way of giving meaning to individual knowledge, to understand personal interpretations, to delimit and clarify the ambiguities, to model the problem and then to find a solution to the problem.

Abdulmajid (2010) asserts that a large part of the knowledge shared in requirements engineering is found in documents that the interested parties understand because it is explicit. However, in the minds of the participants, there remains substantial and undocumented knowledge, which the author identifies as tacit knowledge. In addition, because tacit knowledge is not visible and remains consciously or unconsciously hidden, it provides special challenges when it is included with explicit knowledge accumulated by the team. Abdulmajid created debates to address issues that require clarification; each participant must substantiate their position with arguments that only they know. Therefore, the participant must resort to their tacit knowledge to defend it, exploiting their experiences, abilities and strengths to give knowledge a meaning in the elicitation.

For Hussain (2010), and according to the theory of interpretive research, knowledge is acquired only through social constructs, i.e., language, conscience, common sense, documents, tools, and other artifacts. In requirements engineering, this translates to acknowledging the individuals' different experiences regarding the problem, integrating and analyzing them in depth and translating them to requirements. Hussain's goal is to overcome the problem generated by the volatility of the requirements; therefore, the team cannot allow individual interpretations to take precedence because they are subjective and based on tacit knowledge.

Wan et al. (2011) maintain that software development is an enterprise highly dependent on knowledge, and thus it is fundamental to give the knowledge a meaning before transmitting it. For them, the key resides in the forms in which the explicit and the tacit knowledge are captured. In the former case, they propose combining practices to filter explicit data and delimit knowledge about the processes, and then apply internalization practices within the team, creating an interpretation for the problem. As for tacit knowledge, they suggest applying externalization practices, such as diagrams and cause-effect models, for the interested parties to model what they know about the context of the problem, giving it a meaning in that manner.

According to Flückiger (2011), developing software means working with knowledge, especially because collaboration and communication play important roles in requirements engineering. Flückiger asserted that the knowledge required for eliciting the requirements comes from different dimensions, i.e., the business model, people's experiences, the business rules and the technology involved. Therefore, the team must make an effort to discover the knowledge, integrate it and give it a meaning in a common interpretative language using a process where joint work must take precedence while preventing opinions and personal postures from overshadowing the conclusions agreed upon by the team.

Taheri et al. (2014) assert that eliciting requirements implies working with large volumes of knowledge, which are easily obtained, utilized and given meaning. For them, this knowledge comes from three dimensions: (1) documentation, (2) the interested parties and (3) the development team, so that categories are created that demand immediate interpretation, before that they can hinder integration. They define factors to measure the

Table 1
Knowledge management models in requirements engineering.

Model	Description
Wiig Knowledge Management Cycle (WKMC) (Dalkir, 2005)	Its purpose is facilitating the creation, accumulation, deployment and use of quality knowledge. In requirements engineering, this means that if the goal is acquiring relevant knowledge to apply it in the requirements definition, the team must have the necessary experience to approach the phases.
ICT spiral for knowledge management processes (Pérez and Dressler, 2007)	This model uses ICT as an aid in knowledge management. The team takes tacit and explicit knowledge from the different dimensions, interiorizes it, socializes it, exteriorizes it and combines it to search for requirements.
Integrated Knowledge Management Systems (IKMS) (Botha, Kourie, & Snyman, 2008; Zaki et al., 2008)	It can be considered as an extension of KMS with improved software models, modules and feedback, to operate in wider information domains. They provide access to the data-information-knowledge chain, necessary to elicit and analyze requirements. In addition, they facilitate the access to pertinent information to understand the problem.
Knowledge Management Software Process Improvement (KMSPI) (O'Connor and Basri, 2011)	This is a theoretical-relational model oriented toward facilitating the discovery and acquisition of knowledge that people possess about a problem. Subsequently, by means of iterative tasks, the team assimilates and shares that knowledge to identify the requirements.
Customer Knowledge Management (CKM) (Buchnowska, 2014)	It is defined as a continuous process for generating, distributing and utilizing the knowledge of the interested parties inside the work team, and between the team and the interested parties. The goal is to manage and exploit all types of knowledge possessed about, on and to understand the problem (Buchnowska, 2011).
Knowledge Management Systems (KMS) (Ghorab and Hegazy, 2014)	These are computer-based knowledge management information systems. In requirements engineering, they are used to retrieve, transfer and distribute the knowledge, structured or unstructured, explicit or tacit, of the interested parties to understand and solve problems and make decisions (Turban, Sharda, & Delen, 2011).

Table 2
Evaluation of the efficiency and efficacy of the models.

Knowledge management indicators (Serna and Serna, 2016a,b)	Evaluation					
	WKMC	ICT Spiral	IKMS	KMSPI	CKM	KMS
Creating/Discovering	++		++	++	++	++
Understanding	+	++	++	++	+	+
Comprehension						
Sharing	++	++	+	++	++	++
Applying					++	+
Experimenting	++	++		+		
Validating						
Documenting	+	++	++	+		+
Updating						
Innovating						

knowledge and give it meaning, i.e., completeness, correctness and intelligibility. The team uses this structure to model the acquired knowledge and update it as new knowledge is discovered or existing knowledge is discarded.

For Humayoun and Qazi (2015), requirements engineering teams must model the knowledge involved because, otherwise, they think they lose some potential to use the knowledge to better understand the problem. Their proposal consists in modeling it through six interactive steps, i.e., learning, exploring, capturing, storing, sharing and exploiting. Humayoun and Qazi are convinced that through these six steps, knowledge from people, organizations and documentation is exploited and given meaning.

3.2. *Approaches for knowledge management in requirements engineering*

For this investigation, the term "approach" is used as a way of seeing things and ideas, and dealing with problems related to them (Bunge and Ardila, 2002). In the studies analyzed, we found that the approaches depend significantly on the particular context of the organization and on the make-up of the team implementing the approaches. It was not possible to determine a general framework that represents a clearly established structure about how knowledge is managed in requirements engineering because this review only found interpretations from the points of view of the consulted authors. In general, three approaches were identified that encompass the viewpoints focused on requirements elicitation. These approaches were taken as the basis for answering the second research question.

1 The first approach corresponds to the social interaction processes, where individual knowledge acquisition and subsequent transfer into collective knowledge is of primary relevance. The way in which knowledge is organized and shared inside the organization is also relevant. This model is supported primarily by Nonaka and Takeuchi (1995), who gave particular importance to the formal and informal knowledge (tacit and explicit) focused on experience and unambiguity. In requirements engineering, the analysts use different techniques to elicit the needs of the interested parties. In the process, however, inconveniences present themselves because such needs are not correctly identified and/or recollected because of the particular conditions of each environment and given that the actors do not easily share their knowledge about the problem (Burnay, Jureta, & Faulkner, 2014; Vásquez, Sánchez, Medina, & Amescua, 2014). Furthermore, given that the management and negotiation processes take place in scenarios where there exists a plurality of interests and interpretations, the work must necessarily be collective

(Azedegan, Papamichail, & Sampaio, 2013). Collective work is enabled by social interaction.

2 The second approach is related to models based on artificial intelligence, oriented toward guaranteeing the quality of the requirements elicitation process. In this sense, two characteristic currents were found:
   3 • *Recommendation systems* (Felfernig et al., 2013), focused on (1) recommendations by the interested parties, identifying those people who are able to provide a complete description of the necessary requirements and those who are more inclined to cooperate and (2) requirements recommendation, available for reuse in a software implementation.
   • *Fuzzy logic principles* (Wang, Cheung, Lee, & Kwok, 2008), whose goal is to achieve a better representation of the information, starting from the automation of the conceptual map generation process. In this way, better support is provided for the interpretation and decision-making from data recollected in the project development. This strategy is conceived as a good knowledge management tool because the concepts can be captured or consulted, allowing automatic discovery of the implicit connections to generate new maps. New knowledge is thus inferred, and from the knowledge repository and the organization's historical records, the generation of an organizational memory is favored.

4 The third approach is supported by methods associated with dynamic–game techniques (*gamification*), empowered by the use of agile software development methodologies and frameworks. The latter are founded on the fact that elicitation is not a trivial process because it cannot be guaranteed that all requirements are obtained in the interaction with the interested parties, when one simply considers questions related to what the system to be developed is expected to do (Ribeiro, Farinha, Pereira, & Mira, 2014). In addition, the process of communication and knowledge transfer can become a problem in requirements engineering because the cultural, temporal, geographic and socio-economic diversity of the interested parties are obstacles that must be overcome (Ghanbari, Simila, & Markkula, 2015). Authors turned to this approach because some techniques, such as interviews, questionnaires, usage cases and user stories, are prone to eliciting ambiguous or incorrect requirements (Ribeiro et al., 2014). This deficiency leads to the implementation of characteristics that oftentimes turn out to be unnecessary or erroneous and to the omission of some necessary and important functionalities. Therefore, they are insufficient for identifying all the system needs (Ghanbari et al., 2015).

Our conclusion from this literature review is that if the goal is to improve the quality of the interaction among the interested parties, it is necessary to promote new focuses for knowledge management in requirements engineering, where principles such as game dynamics can provide the necessary feedback and stimulate active participation by the actors (Ribeiro et al., 2014). This may increase collaboration and creativity, generating a better transfer of knowledge adjusted to the nature of the process and the elicitation, where ideas are identified through collaborative work (Ghanbari et al., 2015). In this process, consensus is obtained about what is to be developed, thus improving the requirements specification and their management and negotiation, while also increasing the quality and acceptance of the software (Snijders et al., 2015).

3.3. Efficiency and efficacy of knowledge management models in requirements engineering

In the literature there are diverse opinions regarding the meaning of the term "model," and efficiency and efficacy indicators (Mouzas, 2006). For this investigation, "model" is assumed to be a theoretical abstraction of a given situation or problem, whose purpose is to reduce the problem complexity to make concrete predictions about the situation. In the case of efficiency and efficacy, even though in many contexts they are assumed to be synonymous, here they are treated with individual and distinct meanings. *efficiency* is the measure of the success achieved by managing the input with respect to the requirements obtained, i.e., the performance of the relationship between input and output knowledge in requirements engineering (Low, 2000). *efficacy*, however, is the degree to which the models can find, assimilate and distribute the knowledge necessary to understand the problem at this stage of the lifecycle (Zheng, Yang, & McLean, 2010).

In the studies analyzed, *efficiency* is the ability to obtain the maximum possible performance from the knowledge obtained from the *origin sources*. In general terms, we measure (1) the feasibility of obtaining a better performance with the existing knowledge and (2) the possibility of improving that performance using knowledge management. Regarding *efficacy*, we measure the level to which the goals established by requirements engineering have been achieved, based on shared knowledge. Another question presented by the authors refers to the indicators for measuring the efficiency and efficacy of the models. For them, a good indicator is (1) useful because it answers the questions related to the goals and objectives; (2) reliable because its use always produces the same results any time, under the same conditions and with the same topics; (3) valid because it precisely reflects the concept being utilized for measuring; (4) timely because results must be available to inform decision-makers; and (5) profitable because it must not be excessively costly.

To answer the third research question, we chose the divulged models that included some sort of appraisal of their efficiency and efficacy, as described in Table 1. Note that the chosen works also meet the inclusion–exclusion and quality criteria defined for this literature review.

Based on these findings, and according to the definitions used to answer the questions regarding the efficiency and the efficacy of the knowledge management models in requirements engineering, Table 2 presents a summary of the indicators found, and the researchers' appraisals of the models'.

4. Results analysis and conclusions

The importance of knowledge in requirements engineering can be perceived in the works analyzed in this review. There was consensus among the authors that the appraisal of the data-information-knowledge chain is essential to understand the problem and to structure a solution. For them, giving a meaning to the explicit and tacit knowledge of the interested parties is the first step toward adequately eliciting their needs, while it becomes the basis for the work teams to specify the requirements with clarity.

However, although they recognize its importance, the authors have yet to agree on a method of assigning meaning to the knowledge they compile using various requirements elicitation techniques. Explicit knowledge is amassed relatively easily, whereas tacit knowledge can be personal visions and interpretations; often, interested parties cannot integrate it and give it a meaning that complements what they have learned from the explicit knowledge. This personalization creates a barrier to understanding and comprehending what the interested parties truly express and thus for contextualizing it in the problem. Some of the proposals are aimed specifically at convincing the actors to communicate what they know about the context, and although some may work in certain situations, most fail because they do not take into account that it is necessary to give the context a meaning to associate it with the explicit knowledge.

Regarding the approach to knowledge management in requirements engineering, in this review, it was not possible to find a general framework that represents a clearly established structure for that purpose. This is because the works analyzed clearly reflect the authors' points of view and their experiences in specific contexts. However, the processes of requirements engineering cannot be standardized for all situations. The three approaches found in this literature review need to innovate because according to the authors themselves, it is still not possible to assert that any of them alone is sufficient for the knowledge management in this stage of the lifecycle.

An interactive focus is required, structured such that its principles and axioms allow the team to find knowledge and give it meaning for its subsequent management. That is, it is necessary to understand that requirements engineering is a social communication practice (Kirkwood et al., 2016) in which one must take into account (1) the principle of totality because the whole is more than the sum of its parts; (2) the principle of circular causality because the elicitation of requirements is a set of mutual implications and circular actions and retroactions; and (3) the principle of regulation because the processes in requirements engineering must follow norms and conveniences established as rules. This interaction between the team and the interested parties could be the tool that allows compiling the necessary knowledge to understand the problem and specify the requirements.

The efficiency and the efficacy of the knowledge management models in requirements engineering did not receive high marks from the researchers, however. From the six models analyzed in this work, the following can be deduced:

1 The maximum mark given is Good (++) for some indicators and Acceptable (+) for others.
2 None of them were evaluated for all knowledge management indicators. From the 10 indicators, five were given marks for three of the models and four for the remaining three.
3 The indicators that received the best marks are Understanding and Sharing, although with different perceptions. This is because the models seek to first understand the knowledge to then share it, omitting comprehension. Thus, the knowledge management remains incomplete and the requirements are not adequately elicited.
4 The ICT spiral model has the highest marks for the Understanding, Sharing, Experimenting and Documenting indicators, but no values were found for the remaining six indicators.
5 The most recent models, CKM and KMS, received the lowest marks. This can be understood in terms of the industry and academia possibly still being in the process of comprehension and application, so that their comments have not yet been published.

In conclusion, it can be asserted that, based on these results, knowledge management in requirements engineering is only *Acceptable*. Therefore, if the goal is to contribute to improving the reliability and security of software products and fulfilling the demands of a software-dependent society, it is necessary to structure and promote new models for knowledge management at this stage of the lifecycle. Thus, the hypothesis of this investigation from which this literature review originates is validated, i.e., knowledge management in requirements engineering is not adequate for the complexity of current problems. Therefore, it is recommended that the next model uses the best practices of the approaches and models proposed to date to create an innovative new knowledge management system with an integrated and validated model.

References

Abdulmajid, M. (2010). Facilitating tacit-knowledge acquisition within requirements engineering. In *Proceedings 10th WSEAS international conference on applied computer science* (pp. 27–32).

Al-Karaghouli, W., Taylor, P., & AlShawi, S. (2008). Knowledge management: Using a knowledge requirements framework to enhance UK health sector supply chains. In *Proceedings European and Mediterranean conference on information systems* (pp. 1–21).

Andrade, J., et al. (2006). A reference model for knowledge management in software engineering. *Engineering Letters*, *13*(2), 159–162.

Azedegan, A., Papamichail, N., & Sampaio, P. (2013). Applying collaborative process design to user requirements elicitation: A case study. *Computers in Industry*, *64*(7), 798–812.

Biggs, J., & Tang, C. (2011). *Teaching for quality learning at university*. Berkshire: McGraw-Hill.

Botha, A., Kourie, D., & Snyman, R. (2008). *Coping with continuous change in the business environment, knowledge management and knowledge management technology*. London: Chandice Publishing Ltd.

Breiter, A. (2004). Requirements development in loosely coupled systems: Building a knowledge management system with schools. In *Proceedings 37th Hawaii international conference on system sciences* (pp. 1–10).

Brereton, P., Kitchenham, B., Budgen, D., Turner, M., & Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, *80*(4), 571–583.

Bresciani, P., Donzelli, P., & Forte, A. (2003). Requirements engineering for knowledge management in egovernment. In M. Wimmer (Ed.), *Knowledge management in electronic government* (pp. 48–59). USA: Springer.

Buchnowska, D. (2011). Customer knowledge management models. Assessment and proposal. *Lecture Notes in Business Information Processing*, *33*, 25–38.

Buchnowska, D. (2014). Social CRM for customer knowledge management. *Contemporary Economy*, *5*(4), 65–80.

Bunge, M., & Ardila, R. (2002). *Filosofía de la psicología*. México: Siglo XXI Editores.

Burnay, C., Jureta, I., & Faulkner, S. (2014). What stakeholders will or will not say: A theoretical and empirical study of topic importance in requirements engineering elicitation interviews. *Information Systems*, *46*, 61–81.

Chikh, A. (2011). A knowledge management framework in software requirements engineering based on the SECI model. *Journal of Software Engineering and Applications*, *4*, 718–728.

Dalkir, K. (2005). *Knowledge management in theory and practice*. Amsterdam: Elsevier.

Davenport, T., & Prusak, L. (1998). *Working knowledge: How organizations manage what they know*. Cambridge, Harvard: Business School Press.

Denning, S. (2000). *The springboard: How storytelling ignites action in knowledge-era organizations*. Boston: Butterworth Heinemann.

Dyba, T., & Dingsoyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, *50*(9–10), 833–859.

Felfernig, A., Ninaus, G., Grabner, H., Reinfrank, F., Weninger, L., Pagano, D., et al. (2013). An overview of recommender systems in requirements engineering. In W. Maalej, & Thurimella (Eds.), *Managing requirements knowledge* (pp. 315–332). Berlin: Springer.

Flückiger, M. (2011). The brain's perspective on requirements engineering. In *Proceedings fourth international workshop on managing requirements knowledge* (pp. 1–2).

Ghanbari, H., Simila, J., & Markkula, J. (2015). Utilizing online serious games to facilitate distributed requirements elicitation. *Journal of Systems and Software*, *109*, 32–49.

Ghorab, K., & Hegazy, F. (2014). The influence of knowledge management on organizational business processes' and employees' benefits. *International Journal of Business and Social Science*, *5*(1), 148–172.

Gourlay, S. (2003). The SECI model of knowledge creation: Some empirical shortcomings. In *Proceedings 4th European conference on knowledge management* (pp. 1–10).

Humayoun, M., & Qazi, A. (2015). Towards knowledge management in RE – Practices to support software development. *Journal of Software Engineering and Applications*, *8*(8), 407–418.

Hussain, W. (2010). Requirements change management in global software development: A case study in Pakistan. In *Master Degree Project*. Linnaeus University.

Jurisica, I., Mylopoulos, J., & Yu, E. (1999). Using ontologies for knowledge management: An information systems perspective. In *Proceedings Annual Conference of the American Society for Information Science* (pp. 1–15).

Kirkwood, S., et al. (2016). Towards an interactional approach to reflective practice in social work? *European Journal of Social Work*, *19*(3–4), 1–17.

Kitchenham, B., Brereton, P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering: A systematic literature review. *Information and Software Technology*, *51*(1), 7–15.

Kitchenham, B. (2003). *Procedures for undertaking systematic literature reviews. Joint technical report*. UK: Computer Science Department, Keele University.

Lambe, P. (2007). *Organizing knowledge and organizational effectiveness*. Oxford: Chandos Publishing Ltd.

Low, J. (2000). The value creation index. *Journal of Intellectual Capital*, *1*(3), 252–262.

Mouzas, S. (2006). Efficiency versus effectiveness in business networks. *Journal of Business Research*, *59*(10–11), 1124–1132.

Nonaka, I., & Takeuchi, H. (1995). *La organización creadora de conocimiento. Cómo las compañías japonesas crean la dinámica de la innovación*. México: Oxford University Press.

Nonaka, I. (1994). A dynamic theory of organizational knowledge creation. *Organization Science*, *5*(1), 14–37.

O'Connor, R., & Basri, S. (2011). Knowledge Management in software process improvement: A case study of very small entities. In M. Ramachandran (Ed.), *Knowledge engineering for software development life cycles: Support technologies and applications* (pp. 273–288). USA: Information Science Reference.

Pérez, D., & Dressler, M. (2007). Tecnologías de la información para la gestión del conocimiento. *Intangible Capital*, *15*(3), 31–59.

Ribeiro, C., Farinha, C., Pereira, J., & Mira, M. (2014). Gamifying requirement elicitation: Practical implications and outcomes in improving stakeholder's collaboration. *Entertainment Computing*, *5*(4), 335–345.

Schmitz, D. (2010). *Managing dynamic requirements knowledge – An agent-based approach*. Germany: RWTH Aachen University. PhD. Dissertation

Schneider, L., et al. (2013). Knowledge creation in requirements engineering −A systematic literature review. In *Proceedings 11th international conference on wirtschaftsinformatik* (pp. 1829–1843).

Serna, M. E., & Serna, A. A. (2016a). *La internacionalización como visión prospectiva de formación*. Medellín: Editorial Instituto Antioqueño de Investigación.

Serna, M. E., & Serna, A. A. (2016b). Information and knowledge management in engineering. *Journal of Engineering Design*, in press.

Serna, M. E., Serna, A. A., Sepúlveda, J., & Guevara, R. (2015). Software testing is more than an emergency plan. In ICCT (Ed.), *International conference on communication technology* (pp. 45–54). Melbourne: DEStech Publications, Inc.

Snijders, R., Dalpiaz, F., Brinkkmper, S., Hosseini, M., Ali, R., & Ozum, A. (2015). REfine: A gamified platform for participatory requirements engineering. In *Proceedings 1st international workshop on crowd-based requirements engineering* (pp. 1–6).

Taheri, L., et al. (2014). Identifying knowledge components in software requirement elicitation. In *Proceedings IEEE international conference on industrial engineering and engineering management* (pp. 286–291).

Terstine, M. (2015). The progress of requirements engineering research. *Revista Antioqueña de las Ciencias Computacionales y la Ingeniería de Software (RACCIS)*, *5*(1), 18–24.

Turban, E., Sharda, R., & Delen, D. (2011). *Decision support and business intelligence systems*. Boston: Pearson.

Vásquez, D., Sánchez, M., Medina, F., & Amescua, A. (2014). Knowledge Management acquisition improvement by using software engineering elicitation techniques. *Computers in Human Behavior*, *30*, 721–730.

Wan, J., et al. (2011). Research on explicit and tacit knowledge interaction in software process improvement project. *Journal of Software Engineering and Applications*, *4*(6), 335–344.

Wang, W., Cheung, C., Lee, W., & Kwok, S. (2008). Self-associated concept mapping for representation, elicitation and inference of knowledge. *Knowledge-Based Systems*, *21*(1), 52–61.

White, S. (2010). Application of cognitive theories and knowledge management to requirements engineering. In *Proceedings 4th annual IEEE systems conference* (pp. 137–142).

Zaki, T., et al. (2008). Integrated knowledge management system (IKMS). In *Proceedings IEEE conference on technologies for homeland security* (pp. 287–292).

Zheng, W., Yang, B., & McLean, G. (2010). Linking organizational culture, structure, strategy, and organizational effectiveness: Mediating role of knowledge management. *Journal of Business Research*, *63*(7), 763–771.