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.