A framework for knowledge management in requirements engineering

Edgar Serna M.* and Alexei Serna A.

Facultad de Ciencias Basicas e Ingenieria,
Corporacion Universitaria Remington,
Medellin, 4432, Colombia
Email: eserna@eserna.com
Email: alexei.serna@uniremington.edu.co
Email: alexei.serna@fundacioniai.org
*Corresponding author

Oscar Bachiller S.

Self-employed,
Bogotá, Colombia
Email: oscar.bachiller@gmail.com

Abstract: Managing knowledge properly has become a necessity in all types of professional activities, especially in those whose products are highly utilised and disseminated, such as software engineering. Because we live in a software-dependent society, this technological development requires improving the processes of understanding and manufacturing; thus, it is recommended to use knowledge as a basis to achieve such improvements. This paper presents the results of an analysis of the proposals to manage knowledge in requirements engineering, in which it is evident that the proposals fail to meet the needs of work teams. The practices evaluated in this study propose an integrated framework for the management of knowledge in RE comprised by the conjunction between the phases of knowledge and the stages of RE and characterised by including two principles of complex thought: transdisciplinarity and multidimensionality.

Keywords: knowledge management; requirements engineering; software engineering; multidimensionality; transdisciplinarity.


Biographical notes: Edgar Serna M., Theoretical Computational Scientist, with over 15 years of industry experience as Project Leader in Information Systems and as a Software Architect, and University Professor and Researcher with over 30 years of experience. His areas of research are logic, software engineering, computer science, formal methods, computer mathematics, and knowledge management, around which has published books and papers, and participated in the conference with papers in national and international events.
1 Introduction

To most researchers, as for industry, requirements engineering (RE) is the most important and complex phase of software engineering (SE). This is because it is necessary to fully understand what the customer needs and define a strategy to solve their problems with a software product. To achieve the former, the team interacts with all stakeholders to identify the requirements through a process in which the team uses various communication and observation techniques. The requirements can be represented formally or informally, with the latter being more common because the options regarding functionality and quality of the system are verbalised in natural language. All of the activities involved in RE require engineering, planning and management that, on their own, are collaborative and oriented toward problem-solving. Therefore, RE is the SE phase in which stakeholders discover, capture, transform, capitalise and use the most data-information-knowledge.

This is why, in certain aspects, SE resembles science and mathematics more than engineering itself. For example, alternative frameworks are continually proposed and evaluated to improve the phases of SE, which constitute a scientific process (experimentation). In addition, the continuous definition of layers upon layers of interpretations (abstraction) develops through a process similar to the progress of mathematics. These characteristics differentiate SE from other disciplines because they are the direct result of the role of software as a source of complexity and as a means of choosing to implement new functions, which are discovered through knowledge management (KM) from the RE phase. Such complexity and novelty introduce puzzles that the work team faces; therefore, they must discover, create and manage knowledge.

For some researchers, the problem of not achieving adequate KM in RE is that with current techniques, the knowledge generated in the stages of RE is treated as a byproduct and is rarely exploited (Wan et al., 2010; Silva and Dasilva, 2015). This negatively impacts software development because, in an industry of time and cost constraints that emphasises the end product, KM receives little attention and is minimised to achieve the goal of delivering on time and within budget. It is a process that some characterise as a simplistic formula in which, in the end, the action of translating intentions to code wins out (Sutcliffe and Sawyer, 2013; Alvear and Quintero, 2015). In addition, it is self-destructive because the emergence and inevitability of KM in RE cannot be avoided, and failing to take advantage of what is generated, the team will be condemned to remake the same mistakes and to rethink the same solutions over and over again. On the other hand, not assuming RE as a process of high creation and utilisation of knowledge increases the probability of making wrong decisions in the following phases because what is believed to know about the needs of the client will always be incomplete.
In this paper, it is argued that the KM in RE is fed and enlivened by giving value to
the data-information-knowledge chain, which is generated in the realisation of the stages
that make up this phase because the end product cannot be only the system itself but
rather also includes all the knowledge generated through the different phases of the life
cycle. This includes complete documentation of the decision-making, reasons, reasoning
and criteria that the team took into account to achieve a complete specification of
requirements. This encyclopaedia of knowledge includes, but is not limited to, lessons
learned, experience acquired, mistakes made, solutions proposed and links to the wider
body of knowledge related to the problem. As a solution, a framework is required to
manage knowledge in this SE phase, which allows the work team to build such an
encyclopaedia.

This can be achieved through a general framework for managing knowledge in RE,
which can be widely used and serve as a basis for structuring a KM model at this stage.
In this paper, a framework is proposed with the objective of providing work teams with a
support tool to manage the knowledge they discover, analyse and integrate into the
specification of requirements. This document is structured as follows: Section 2 details
the conceptual bases that support the proposed framework, Section 3 summarises and
analyses some proposals regarding KM in RE that the authors have presented, Section 4
presents an analysis of the importance of knowledge in RE, Section 5 describes the
proposed framework for KM in RE, and Section 6 presents the conclusions of the work.

2 Conceptual bases

2.1 Requirements engineering

The term ‘requirement’ is similar to the term ‘characteristic’, although the former’s scope
is greater and approach more technical. A widely accepted definition to refer to it is
summarised in that it is a statement of what the system should do, how it should behave,
what properties and qualities it should have, and the constraints it must meet (Clements
and Northrop, 2006). In addition, a requirement differs from a necessity in that it is
synonymous with need, so it is oriented toward lack or lack of something (Pagani, 2013).
Aurum and Wohlin (2003) consider that requirements are a verbalisation of the
alternative options regarding the functionality and quality of the system.

For its part, RE is the phase of SE that is responsible for discovering, analysing and
specifying the properties and limitations that clients require to be satisfied with a
software product (Davis, 2003). For Nuseibeh and Easterbrook (2000), it is a process of
discovering the purpose of the system, identifying the stakeholders and their needs and
documenting them such that they can be analysed, communicated and implemented.
Therefore, RE is considered a complex task in which, through processes of
comprehensive communication with stakeholders, the real properties and limitations of
the system are concretised. This communication is developed through collaborative and
problem-solving activities in which a great deal of knowledge is exchanged and analysed
(Maalej and Thurimella, 2013).
2.2 Knowledge

In general terms, knowledge is a term that relates to the facts, information and skills that a person acquires through experience or training. This term is used popularly and daily in disciplines such as computer science, where, often, they use the term interchangeably with data and information. However, according to Thierauf (1999), data represent unstructured facts and figures, information is structured data that can be used to analyse and solve critical problems, and knowledge is gained from people based on their actual experiences. That is, information is data about data, and knowledge is information about information. For Davenport and Prusak (2000), knowledge is a fluid mixture of experiences, values, information and intuition that people possess.

2.3 Knowledge management

The term KM became popular in the late 1990s and early 2000s when it became a buzzword. However, the promises that emerged from this euphoria were eclipsed by a reality in which KM initiatives yielded absolute projects failures (Akhavan et al., 2005). It should be noted that simply trying to define the term can lead to significant confusion because a completely accepted definition has not yet been achieved (Frost, 2014). The many definitions that have been published include that from Skyrme (2011), for whom KM is the explicit and systematic management of vital knowledge and its associated processes of creation, organisation, dissemination, use and exploitation in the pursuit of a specific objective. This statement approximately describes the activities that comprise RE.

3 Related work

- ISO STEP – AP233 (ISO, 1994). The objective of this proposal is to provide a mechanism capable of describing the information of a product throughout its lifecycle, regardless of the system. This makes it suitable for implementing and sharing databases of products and files, in addition to knowledge of people. Although typically used to exchange data and information, it can also be structured into RE. Those who have evaluated this proposal affirm that it provides domain-specific information and that its protocols offer specific knowledge to find and define requirements (Heimannsfeld and Müller, 2000). Scott and Cook (2003) argue that this is a rigorous attempt to identify and track the progress of the stages of RE. Ratchev et al. (2005) adapt it and incorporate it into systems development, whereas Johnson (2006) states that it adapts over time to clarify and update requirements. As insurmountable characteristics, all agree about the following:
  - it does not solve the problem of ambiguity about what is a requirement
  - it is not sufficient to provide rigour to KM in RE because the actors do not achieve a broad understanding of the problem
  - requirements are expressed in natural language that represents the discovered knowledge.
MTKISDP. Hickey and Davis (2003) present a formal proposal to elicit requirements that prioritise the necessary critical knowledge from

- understanding what analysts need to do to manage knowledge,
- knowing how to select an elicitation technique, and
- realising that as the ability to perform elicitation improves, the likelihood of the system meeting the requirements improves.

Regarding this proposal, Bendjenna et al. (2008) state that it focuses basically on the selection of a technique and leaves KM in the background. In this same sense, Vásquez et al. (2014) argue that KM processes must encompass much more than identifying and capturing knowledge, so this proposal fails to capture the relevant knowledge for each specific domain. After evaluating this proposal, Tuunanen (2003) concludes that although it allows users to be represented and uses groups and communities, it does not take into account the multidimensionality or transdisciplinarity of requirements, thus making it difficult to manage knowledge.

RMKMSE. The objective of the proposal of Andrade et al. (2006) is to provide those involved in software development with access to the best possible knowledge in a timely manner, which is why they integrate management, classification and knowledge characteristics. For Carreteiro et al. (2016) this proposal essentially focuses on the capture of knowledge in the requirements phase, making use of corporate memory as the main source, but does not efficiently take advantage of the stakeholders. Rodríguez et al. (2014) and Talib et al. (2010) say that it focuses mainly on participation in knowledge and reuse of lessons learned, leaving aside other aspects of RE. On the other hand, Anquetil et al. (2007) argue that it does not take into account quality control and correction and that it neglects the exchange of knowledge capture.

KBRE. Pavanasam et al. (2010) propose Knowledge-Based RE through the integration of expert systems and domain-specific modelling. After applying this proposal, Valero et al. (2013) state that decision-making requires knowledge of multiple dimensions, so a knowledge base, policies and requirements must be created to consider them. For them, KBRE focuses on how to model and use domain knowledge but forgets the tacit knowledge held by stakeholders.

KMPRE. Pilat and Kaindl (2011) adopt a spiral process to transform tacit knowledge into explicit and vice versa, with the idea of overcoming the problem of knowledge exchange and emphasising the importance of eliciting the requirements from that reciprocity. For Venkatesh and Kumar (2013), this proposal dominates the communication between the stakeholders, which solves part of the problem of the transfer and the exchange of knowledge when these parties are reluctant to do so. Olmos and Rodas (2014) argue that the process is incomplete, costly and time-consuming to implement; in addition, it does not achieve the transformation or the transfer of knowledge in the dimension that is required in RE.

KMFSE-SECI (Fernández and Gómez, 2002) uses ontologies to describe the specification and represent the content of the requirements; it is based on the SECI model. It needs to go beyond explicit knowledge because, in RE, the tacit is very
important (Smith, 2004). It is somewhat flexible and adaptable, but it is difficult to apply to other functionalities (Malafsky and Newman, 2009).

- \textbf{KMoS-RE} (Olmos and Rodas, 2014) is oriented to the transformation and transfer of knowledge and is characterised by
  - taking into account the Jackson model (Jackson, 1995) for software development
  - being based on SECI (Nonaka and Takeuchi, 1995)
  - incorporating methods to identify, capture, index and make explicit tacit knowledge.

In this regard, Humayoun and Qazi (2015) argue that it is a KM proposal for an RE strategy that does not take into account the complexity and multidimensionality of requirements, structured informal domain, or the ambiguities that need to be addressed in RE.

4 Knowledge of requirements engineering

In RE, knowledge originates from two dimensions:

- epistemological (tacit or explicit)
- ontological (individual or collective).

In which it exists in the forms

- embrained
- embodied
- encoded
- embedded.

The interrelationships between the dimensions and the forms of knowledge in RE are shown in Table 1.

- \textit{Embrained knowledge} is individual and explicit, depends on individual conceptual skills and cognitive abilities and can be formal, abstract or theoretical.
- \textit{Embodied knowledge} is individual and tacit, action-oriented, practical and built on practical experience. In addition, it has a strong component of automation and voluntariness; thus, its generation and application do not easily fit into a conscious process of decision-making. It is developed in a specific context and is considered relevant only when the use solves the problem; thus, the generation cannot be separated from the application.
- \textit{Encoded knowledge} is collective and explicit and shared in the form of symbols and signs; thus, it is simply known as information. After its encoding, it is stored in documents and other storage media that trace the path of collective behaviour.
A framework for knowledge management in requirements engineering

Inevitably, it is simple and selective and thus not efficient for capturing and preserving individual tacit abilities and judgements.

- *Embedded knowledge* is the form of tacit knowledge that resides in shared norms and rules. It is based on the beliefs and collective interpretations that make possible an effective communication between the communities of practice. In addition, it exhibits specific, contextual and dispersed relationships and is organic and dynamic.

Because of these different forms, knowledge in RE can be expressed as explicit or tacit (*epistemological dimension*); the differences between the two are as follows:

- **Encoding and transfer.** Explicit knowledge can be encoded, abstracted and stored in the objective world (context of the problem), in addition to being understood and shared without the assistance of a *connoisseur* because it is easy to communicate and transfer. In contrast, the tacit is personal, intuitive, disjointed and resides in a non-objective world (e.g., beliefs, interpretations, myths, and assessments of the problem), so it is not easy to communicate, understand or use it without the assistance of a *connoisseur*. From this dimension, a work team, through common communication and interpretation techniques for all stakeholders, transfers, shares and analyses the explicit knowledge in the space-time of RE. In contrast, the tacit requires specific techniques and methods to make each connoisseur share what he knows.

- **Achievement and compilation.** Explicit knowledge is achieved through logical deductions, which the work team uses through observation and analysis, and compiled from formal studies with stakeholders on documents and standards. In contrast, tacit knowledge is acquired through experimental practices with individual actors and compiled through observation and deduction.

- **Accumulation and appropriation.** The explicit can be added from a single location, stored as objectives and appropriated without the intervention of the knowledgeable actor. In contrast, the tacit, because it is personal and contextual, is distributed and cannot be accumulated nor appropriated with ease; thus, the team must use different techniques for each knowledgeable subject.

Additionally, knowledge in RE can also be manifested individually or collectively (*ontological dimension*). In the former, knowledge resides in the mind and in the skills and abilities of individuals, so it is considered their property and is autonomous. It is characterised by being inevitably specialised and of a specific domain, and it can be transferable, but it moves with the owner, which is why it generates retention and accumulation. The collective knowledge, on the other hand, is shared and distributed among different individuals and stored in the rules, processes, routines, guidelines and rules that they use to solve problems.

Since the turn of the century, SE researchers have focused their attention on the study of the knowledge involved in software development and have published approaches and tools with the aim of improving their management throughout the life cycle (Bjørnson and Dingsøyr, 2008; Babar, 2009). Some were geared to producing experiences (Basili et al., 1994), pattern design (Gamma et al., 1995), team processes (Humphrey, 1999), enriched knowledge (Basili et al., 2001), KM-based processes (Holz, 2003), personal processes (Humphrey, 2005) and rational management (Dutoit...
et al., 2006), but, with the exception of rational management, they have paid little attention to KM in RE, focusing mainly on the design, implementation and maintenance phases.

### Table 1 Dimensions and forms of knowledge in RE

<table>
<thead>
<tr>
<th>Epistemological dimension</th>
<th>Ontological dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Individual</td>
</tr>
<tr>
<td></td>
<td>Collective</td>
</tr>
<tr>
<td><strong>Explicit</strong></td>
<td>Embrained</td>
</tr>
<tr>
<td><strong>Tacit</strong></td>
<td>Embodied</td>
</tr>
</tbody>
</table>

On the other hand, the empirical studies of this same subject have focused little on the KM of stakeholders. For example, Ko et al. (2007) study the information needs of developers, Sillito et al. (2008) analyse the need for developer knowledge for software maintenance, and Robillard (2009) studies the obstacles that developers face when reusing knowledge of components. None of the above studies focus on stakeholder awareness when eliciting or implementing requirements, an important issue to understand their needs and provide them with effective support for understanding the problem.

In addition, KM tools, such as document management systems (Rus et al., 2001), recommendation systems (Happel and Maalej, 2008; Robillard et al., 2010), information search and retrieval tools (Bajracharya and Lopes, 2009), wikis (Aguiar et al., 2009) and repositories based on ontologies (Happel et al., 2010) have become popular in SE processes and in the RE community. All this creates confusion in the work teams in RE because they are offered many options but no practical experience that allows them to select the most suitable tool for the problem that they are trying to solve. For this reason, an integrated framework that can be widely used and serve as a basis for structuring a KM model in this SE phase is required to manage knowledge in RE.

## 5 Framework for managing knowledge in requirements engineering

RE is an SE phase in which the interaction with stakeholders is permanent because it is here that the team recognises what they know about the problem and what they need from the system. In the different stages that compose it (Figure 1), permanent exchanges of data-information-knowledge are performed, which the team must manage adequately to achieve the structuring of a good document of the specification of requirements.

In the process, in addition to the requirements that stakeholders share, one must also locate those who cannot, or do not want, to share. These requirements are characterised by being difficult to express, convert, communicate and share; they only relate to the domain of the system; they are people’s own experiences or knowledge; they are difficult to codify and articulate; and they are generally expressed vaguely and crudely from the viewpoint of the person that possesses them. These types of requirements are classified as explicit and tacit, respectively, and, in general, the latter can be more difficult to obtain from people directly and verbally or to discover via observation.
An integrated RE KM framework is required because, in addition to understanding the problem and structuring a possible solution, the team must address both people who want to collaborate and others who barely do so. Knowledge, explicit or tacit, must first be discovered from its bearer to manage it through collaborative analysis, discussion and understanding. Such an environment must be iterative, in which stakeholders recognise the technical language and the team becomes familiar with the needs and forms of communication that they use. However, this is not as simple as is believed because the process involves working with people from different disciplines, with dissimilar views about the problem and the system and with complicated aspirations, rooted interpretations and many more characteristics. This situation generates communication problems that impact the quality of the specification of the requirements. Next, the framework that is proposed as an alternative solution to all these drawbacks is described.

In practice, RE is developed in an intensive knowledge environment, so the work team must manage it to take advantage of it efficiently to build the system. In this phase of the life cycle, knowledge is diverse and growing, so teams have problems identifying, locating and finding the best method to manage it. Based on MoDeMaRE (Serna and Serna, 2016), Figure 2 presents the integrated framework of the proximity between the phases of knowledge and the stages of RE, whose objective is to find explicit knowledge and discover and transform the tactical to generate a document that specifies solid and reliable requirements.
5.1 Discover

The objective of this phase is to select the tools/practices appropriate to the context and domain of the problem and the system, to discover the data-information-knowledge chain that the people, the organisation and/or the systems have. An immersed characteristic is to be able to detect the patterns that this chain follows because, generally, they are hidden, and it is necessary to apply specific practices for each expected type. The contents of documents, records, multimedia and data repositories represent explicit knowledge and therefore are source dimensions for these requirements. Taking into account the different dimensions (socio-cultural, political-institutional, administrative, technological and disciplinary) from which the requirements originate, we must recognise patterns and relationships among data, text or images. In these cases, observation is the most efficient tool, but data mining and text mining are also useful in this task.

Discovering tacit knowledge is more complex and requires a transdisciplinary management vision. The fact that this type of knowledge is hidden does not mean that it is impossible to find, but it requires more skills and abilities in the work team. Again, observation is a powerful tool, but there are other qualitative and quantitative alternatives, such as surveys, questionnaires, individual and group interviews, focus groups, and network analysis. On the other hand, people leave a mark of their knowledge and experience in each opinion or participation, so it is recommended to follow these marks and try to decipher them, especially when those people are reluctant to collaborate directly. This is fundamental to those individuals who are considered primary sources of requirements, that is, they are the only ones who know the central processes of the problem and the system.

Both explicit and tacit knowledge can exist in internal or external dimensions to the problem, the organisation and/or the system. This is evidenced in the associations or networks in which the stakeholders, such as suppliers, customers or users, formally or informally participate and which relate in some way to the problem. Here we must be cautious because many of these networks are based on trust between the actors, and trying to break it can generate more secrecy. The recommendation is to adopt practices aimed at discovering what each individual knows without violating their relationships or
decisions. For example, requesting a formal document in technical language describing the most important aspects of your work could offer enough mining to uncover the chain related to embedded requirements, in addition to tracking its footprint.

5.2 Capture, analyse and understand

The objective of this phase is to classify and understand the data-information-knowledge chain discovered such that it can be used in the other phases. The idea is to establish good analytical practices to distill the knowledge that will really be useful to capture the requirements. It is very important that the teamwork as such because it must discuss what has been discovered and reach an understanding. To achieve this goal, it is necessary to take into account the different points of view and interpretations of the stakeholders, in addition to the dimensions and disciplines involved, i.e., the team must act as a knowledge manager to achieve the necessary understanding.

There are different methods to achieve this, but because the process involves capturing, analysing and understanding the discovered knowledge, we recommend using ontologies (Fernández and Gómez, 2002; Smith, 2004) and taxonomies (Malafsky and Newman, 2009). In addition, when functioning as a knowledge manager, the team can apply performance measures and flows, thus obtaining a mapping of the discovered chain. Without losing sight of the fact that it is about capturing the knowledge that indicates the presence of requirements, the analysis is a task that, if not organised as a process, can complicate the capture. The team must have sufficient ability to filter what it does not contribute to the objective of the phase, that is, at this moment, it has discovered a great amount of data-information-knowledge, both explicit and unspoken, but not everything is useful or understandable. In this phase, only the knowledge that the team determines as essential for RE is mapped.

On the other hand, it is common to confuse terms such as understanding, understanding, knowing and knowing, but there is a wide variety of their meanings. Without entering into the controversy that can be generated, we clarify what is meant by understanding in this work: people can know data and information, but they understand them only when they become knowledge, meaning when they give meaning and then generate learning. For this reason, in this phase, knowledge is built from data and information but analysed through a process of selection, organisation and integration, to which the team must add what it individually knows and understands about the problem and the system.

The proposed process is as follows:

- Perform a screening of the results of the analysis to differentiate the myths, beliefs, curiosities and interpretations of the true needs that the system must satisfy. It should be remembered that each discipline presents an individual view of the context of the problem and the system, therefore, is integrated to arrive at an overall understanding of the situation.

- Determine how much of the data-information-knowledge chain is actually understood. The team takes what it understands and knows from the disciplines and dimensions involved to create a map of the understanding of the problem and the needs of the system.
• Relate the understanding of the context to the current situation. Each chain analysed is located in an earlier moment of the context, so we have to relate it to the current moment and project it to a later moment because otherwise, we will be trying to solve the problem based on the understanding of a past context.

• Give meaning to contextualised knowledge. Through teamwork and integrating all stakeholders, the understanding is modelled such that everyone agrees on the learning achieved.

• Convert learning, from disciplines and dimensions, into a list of basic requirements that the system must satisfy. This understanding is fundamental to creating the first prototype of the solution.

5.3 Integrate and transform

So far, KM has been performed jointly by all actors, but there are still individual/disciplinary interpretations that must be synthesised in a single representation. For this, we must involve the body of knowledge external to the problem situation, i.e., take into account other dimensions that are represented in the problem and in the system. This is an integration process in which the existing knowledge interacts with the external to achieve a model of knowledge that is broad and agreed upon in terms of the disciplines and dimensions involved. In this manner, an acquired and integrated learning about the problem, in addition to the context and dominion of the system, is achieved.

It is important not to forget that transdisciplinary integration and transformation means integrating people, knowledge and technologies that belong to the different domains and contexts in which the system exists or with which it has a relationship. The two activities to achieve this are as follows:

• find a common language, which is necessary to discuss needs and requirements, options and paths and the content of the document specification

• design a methodology for integration.

Transdisciplinary integration should not be confused with attempts to establish knowledge of the situation as a whole because this would be like regrouping knowledge in each discipline and origin dimension.

It is necessary to achieve a common language because, in this manner, the barriers of understanding between disciplines are broken, and people can express their interpretations such that all can understand and/or transform. One method to do this is to identify concepts common to disciplines that are widely known to any professional, such as from mathematics and systems theory, or basic concepts of management and administration. In this sense, it is convenient to consult about the different schools of thought systems. After achieving a common language, the next activity is to integrate methods and theories into a suitable methodology to transform knowledge. This is a team effort, and although the result might not be a unified theory, a mosaic in which the methods are connected through the shared concepts that the team determined from the common language is created. The final product of this phase is a list of requirements integrated and transformed into a technical language, which all participants understand and enrich from their disciplines with the knowledge and experience that they possess.
5.4 Concretise and capitalise

Regarding knowledge, there are two concepts that have been popularised and studied for some time: the abstract and the concrete. The former refers to a type of interpretation isolated from a reality that, to be represented, requires imagination and ingenuity. This happens continuously in RE because the stakeholders construct their own interpretations of the problem and of the system from their discipline and experience, with the drawback that they come to accept it as knowledge, which creates barriers of communication that must be demolished to reach a true understanding of the situation. In this phase of KM, the team still maintains abstract interpretations of the requirements of the system, so it is necessary to specify the knowledge from the common language and the methodology structured in the previous phase.

On the other hand, concrete knowledge refers to the image that the team moulds of the problem and the system that is an amalgam of singular aspects and properties that they understand because they consider them as moments of a whole, determinable by the specific content of the same problem. Here, it is important to keep in mind that people do not decipher or solve real-world problems but rather abstract representations of them, and in RE, it is common to involve a reduction process in which they sacrifice aspects that they consider not relevant to concentrate on variables they believe give a more general scope to their interpretation. This happens because this image is constructed from their discipline and, from it, they find the answers that they can and need to find, with the aggravating factor that they accept them universally. For this reason, it is important that the teamwork from the multidimensional and transdisciplinary because, in this manner, it can concretise the integrated and transformed knowledge.

In addition, it must be borne in mind that problems are analysed in situations, without using general definitions, because each part of the same is linked and belongs to a specific situation, and only through a concrete compression of this link can we arrive at a correct analysis of the knowledge involved. By understanding the knowledge and finding the common language to express it, the team can concretise it, i.e., capitalise on the discipline to find a solution to the problem. This is because the sum of experiences, interpretations and skills of all the actors enables the potential development of ideas to synthesise what is really known about the problem and the needs that the system must satisfy.

5.5 Validate and use

In general terms, validation is a process to ensure that something is right or conforms to a certain level of demand. In the framework presented in this paper, the term is used to guarantee that the data-information-knowledge chain is located within the accepted limits of the understanding of the problem and/or the system. This process should be considered critical in the RE because knowledge and interpretations are crucial components that must be validated. Most authors agree that this process is developed through two activities:

- verification, which intends to reach a correct structure for knowledge
- evaluation, which intends to demonstrate the usefulness of that knowledge to arrive at the right conclusions, that is, to identify the requirements to be satisfied.
Traditionally, to facilitate the completion of the procedure, verification activities rely on specific techniques, such as decision tables or trees, some graphical forms and even automatic learning techniques. Because much of the concrete knowledge is represented in the business rules of the company, it is recommended to formalise these requirements to facilitate unified interpretations. On the other hand, evaluation activities are based on evidence that concludes that knowledge is correct. Both verification and evaluation are transdisciplinary tasks that take into account the dimensions of the data-information-knowledge chain. At this stage, the chain must be broken to work only with the concrete knowledge and capitalised because the objective is to write the document of the specification of requirements, that is, it is necessary to use the validated knowledge.

The first items to establish are the concrete objects that represent that knowledge, that is, materialise the understanding of the problem. This is achieved through obtaining knowledge of the domain and the context of the domain, in addition to knowledge of the model that represents the validated knowledge. Subsequently, the scope of the use is defined, that is, the transdisciplinary referential limits are demarcated. This is because disciplinary knowledge is limited in individual interpretations; then, the relationships between its boundaries are required to delimit the use of the whole set of disciplines involved in the problem and the system. In any case, the knowledge utilisation criteria must

- be compliant with the procedures of the other phases,
- have specific dependence,
- be independent of the evaluation criteria,
- have a practical unit
- be complementary to each other.

6 Validation and verification of the comprehensive framework

Verification and validation (V&V) is a methodology that allows one to identify the certainties and errors of proposals, models and proposed frameworks in any discipline. Quantifying the confidence and accuracy of the proposed provides important information to make decisions about its applicability and functionality. However, whatever the modelling paradigm or the solution technique that is applied, the performance measures will only have value if the framework is applied in a good representation of the real system. Of course, what constitutes a good proposal is subjective, but from a logical point of view, the criteria to judge its benefits are based on the results of V&V and correspond to those expected in its actual application.

In this sense, accuracy is observed as the absence of systematic and random errors, which is called fidelity and precision. Generally speaking, verification must prove that the framework is true, whereas validation demonstrates credibility. For this purpose, there are principles that must be taken into account:
• all frameworks are wrong, although some are more useful than others
• a framework should not be considered as the only one
• it is necessary to carefully test whether a framework meets expectations.

The most important thing in this sense is that a framework, such as the one proposed in this work, cannot be validated and verified as a whole because its application context is complex and the process requires a long time, as is typically the case in industry. For this reason, it is recommended to simulate the framework individually for each of its components, incrementally and applying top-down tracking, as proposed in Figure 3.

**Figure 3** V&V process for the proposed framework

![V&V process for the proposed framework](source: Adapted from Law (2007))

This process was completely simulated on the RE phase in a security management system of income in a bank; based on an analysis of the results, the necessary adjustments and purifications were made for each of the stages of the engineering of requirements of Figure 1 and the phases of knowledge of Figure 2. The focus of these activities is the compilation and analysis of the data-information-knowledge chain found in the application of the framework at each stage because they directly influence decision-making to
adjust the associated parameters to achieve a better understanding of the knowledge acquired

- redesign the framework based on the results of the simulation
- prove the acceptability of the framework with respect to each simulated phase-stage.

The objective of the simulation of the framework was to find the following:

- **Apparent validity.** Several specialists in the subject and with knowledge about KM and RE were asked to make a conceptual evaluation to determine if the framework and/or its behaviour were logical and reasonable. The contributions and suggestions were received, and the conclusion was that it is logical and viable but that the application and testing process takes time in the industry.

- **Predictive validation.** Through simulated applications, the framework was used to predict its behaviour and response values in a real context. With the results, the necessary adjustments were made, and each stage-phase was finalised before starting the next one. In Phase 0, Discovering Knowledge, it was found that determining explicit knowledge is complicated because at this stage, the stakeholders have predispositions towards the new system, and those barriers must be identified and collapsed before the discovery phase begins. This was corrected in the framework in such a manner that first the sensitisation was realised, soon to initiate the discovery of the explicit knowledge. At this stage, it is not yet intended to find massively tacit knowledge in people, which is corroborated in the simulation. After performing the simulation in a progressive and iterative manner, in each of the RE stages and KM phases, the adjustments and the suggested purifications were made, demonstrating at the end that the framework exceeds the predictive validation.

- **Disclosure and accreditation.** This refers to the publication of the framework and the search for an industrial application. This objective is achieved with the acceptance of the paper in the journal and its subsequent edition. In addition, colleagues in universities begin to experience it in their subjects and to enable students to put it into production. In the same manner, the authors initiated contacts with the industry to provide advice in the application of the framework and to initiate its validation in real problems and systems. An adequate progressive understanding of the proposal has been achieved because it is continuously socialised and repeated simulation predictive validation.

### 7 Conclusions

RE is considered the most important and complex phase of software development because it is used to discover and specify the needs that the system must satisfy. It is a process in which the stakeholders communicate and share their ideas and appreciations about what they need and expect from the solution that the team will present. For this reason, it is the phase in which there is greater communication between the stakeholders and the SE team and in which the data-information-knowledge chain is continuously shared.
A framework for knowledge management in requirements engineering

Although this feature is recognised in the community and industry, KM in RE is still not satisfactory for meeting the needs of stakeholders and the system itself. A review of the related work indicates that previous proposals fall short in this respect and that few manage to approach the true management of this chain. Some specialise in the elicitation of requirements, others focus on the development of the same, and others address the specification, but we are homogeneous throughout all stages of this phase of the life cycle.

Consequently, and after analysing what has been published and experienced so far regarding KM in RE, this paper proposes an integral framework to achieve this goal. Some practices are taken from the proposals of the review and are enriched with principles of complex thought, such as transdisciplinarity and multidimensionality, aspects that have not been taken into account by other researchers.

References


Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1995) Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, USA.


