A Model for Enhancing Tacit Knowledge Flow in Non-Functional Requirements Elicitation

Sandra L. Buitrón, Francisco J. Pino
Grupo IDIS, Facultad de Ingeniería Electrónica y Telecomunicaciones. Universidad del Cauca
Popayán, Cauca. Colombia
{sandrabr, fjpino}@unicauca.edu.co

Brenda L. Flores-Rios, Jorge E. Ibarra-Esquer, María Angélica Astorga-Vargas
Instituto de Ingeniería, Facultad de Ingeniería
Universidad Autónoma de Baja California.
Mexicali, Baja California. México
{brenda.flores, jorge.ibarra, angelicaastorga}@uabc.edu.mx

***Abstract*—Elicitation phase includes features focused on the experience and knowledge of the people to support the capturing and tracing of the business/user requirements, functional requirements, and Non-Functional Requirements (NFR). In this phase, the stakeholders may belong to the same or different organizations and, therefore, their vocabulary and terminology used in NFR expression can be diverse, habitually depending on roles, experience, perspectives and levels of knowledge. This document focuses on the modeling of tacit knowledge flows of the stakeholders involved in the NFR elicitation process. The components of the MERliNN framework are presented, which strengthens a conceptual relationship between the disciplines of Knowledge Management and Requirement Engineering. The TCER core allows to visualize the direction of tacit and explicit knowledge flow in four scenarios of NFR elicitation. We present the description of the first scenario of knowledge instauration and its validation by case study.**

Keywords—Non-Functional Requirement; Requirement Engineering; Knowledge Management; Tacit knowledge; Knowledge flows;

I. INTRODUCTION

Within the software developing organization, knowledge is represented by intangible assets, which refer to human capital, structural capital and client capital. In several works, there has been detected a characterization of knowledge assets when referring to learned lessons, good and better practices, heuristic rules and continuous generation of ideas and experiences (tacit knowledge) that innovate the organization's processes [1, 2]. The re-utilization of knowledge assets is an efficient technique for transmitting knowledge among a software developing team. For these reasons, some authors have remarked the convenience to narrowing the gap between the disciplines of Software Engineering and Knowledge Management (KM) [1, 2, 3, 4]. Capote et al. state the necessity of a management mechanism that eases capture and use of valuable tacit knowledge during the performance of improvement cycles, useful for Software Process Improvement (SPI) teams [3]. However, a common aspect among software product and services development organization is the lack of ability to capture, create or formalize tacit knowledge that allows them to maxime their competitiveness and focus efforts on the implementation of software projects [4].

Authors such as [3, 4, 5] emphasize the obstacles derived from tacit knowledge's nature due to: a) its characteristics are difficult to imitate, substitute or transfer, making it a critical resource for an organization; b) the complication to explain people's knowledge in words; c) the capability of using tacit knowledge without recording it, since it has no direct benefit at individual level; and d) the risk of losing the tacit knowledge when making it explicit. This situation raised a special interest in analyzing processes for the acquisition, accumulation, transfer, and use of the existing knowledge and the creation of new knowledge from tacit knowledge [2, 3, 4, 5].

Knowledge flows are an emerging topic of investigation in the KM research field, and several studies have built knowledge-flow models to illustrate knowledge sharing among tacit knowledge workers [6]. For example, the Spiral of knowledge or SECI model (as an acronym for Socialization, Externalization, Combination and Internalization processes) provides an understanding of how organizations create knowledge in order to maximize the administration, application and transfer of it through several creation and expansion patterns [6].

Besides the epistemological dimension of SECI model there are other theories that back up KM, such as the Data, Information and Knowledge (DIK) hierarchy, the Theory of Resources and Capabilities and Alavi and Leidner perspectives [7] which we have selected with the objective of analyzing tacit knowledge flows in phases of the software development's life cycle.

Nowadays, the study of tacit knowledge to improve the tasks of Requirement Engineering (RE) is an open research question [4, 8]. RE is one phase within software development process particularly critical in terms of managing tacit knowledge because of the presence of multiple stakeholders with different backgrounds, perspectives and expectations [9]. There are several methods and techniques used to elicit requirements, such as introspection, interviews, focus group and even protocol analysis, but some of them have limitations in handling tacit knowledge, and generate ambiguous, inappropriate, and incomplete requirements. These limitations

0-7695-6342-2/17/31.00 ©2017 IEEE
DOI 10.1109/CONISOFT.2017.00011

can lead to inadequate software solutions, or be the cause of increasing the project development time [9].

In RE, the term Non-Functional Requirements (NFR) refers to concerns not related to the functionality of the software, such as usability, flexibility, performance, interoperability, and security [10]. In addition, published literature exposes a lack of mechanisms for NFR elicitation, and an unawareness on this type of requirements from the stakeholders and the software development organizations [11]. This evidences the need to establish mechanisms or strategies that allow enhancing and reducing the loss of all the tacit knowledge that may be useful for the achievement of activities of requirement elicitation.

In this paper, we present the MERliNN (from its name in Spanish, Marco de trabajo para la Elicitación de Requisitos No Funcionales) framework for managing the tacit knowledge that team members acquire during Non-Functional Requirements phase. The purpose of the core of MERliNN is to identify knowledge flows that enable a software development team to enhance and support the RE process. The main contribution is that it considers: 1) the integration of knowledge flows for data and information gathering in NFR; and 2) a visualization of how knowledge transformation takes place, and the tacit knowledge flows between stakeholders according to their levels of knowledge. This way, MERliNN takes into account four different knowledge transformation scenarios that might be present within a context of continuous communication, understanding, and learning of concepts, uses, identification, valoration, negotiation, and verification, among other actions related to NFR, taking place in a software development organization. This document focuses on the first of the four transformation scenarios, dealing with knowledge instauration.

The remainder of this paper is structured as follows. Section II introduces general description regarding the multi-views to explain how tacit knowledge flows within the software organization through Data-Information-Knowledge hierarchy, SECI model and knowledge perspective approach. Section III brings the meaning of knowledge flows and its elements. Following, section IV presents research methodology. Subsequently, section V describes components of MERliNN Framework. Then, section VI presents a preliminary validation through a single-embedded case study. Finally, the conclusions are drawn in Section VII.

II. BACKGROUND

A. Data-Information-Knowledge hierarchy

The DIK Pyramid notices the knowledge around the added value process, associated to elements of an informational chain: Data (D), Information (I) and Knowledge (K), also called it DIK hierarchy for its acronym.

Alavi and Leidner define data as raw numbers and facts [7]. Information is a collection of processed data that turns into knowledge once processed in the individual's mind and then into information again when it is articulated or communicated to other people through oral communications (tacit knowledge), texts, electronic formats, or any other form of written or printed communications (explicit knowledge). Knowledge refers to the information with added value [12] that an individual has in her mind; it is customized and subjective, related to other facts, procedures, concepts, interpretations, ideas, observations, judgments and elements that may or may not be useful, accurate or structured. Thus, knowledge involves strategies, practices, methods or approaches. Once made validly and oriented to an objective, it generates intelligence.

B. Tacit knowledge in the SECI model

The organizational knowledge creation model is based on the assumption that knowledge is created and spread through the social interaction of tacit and explicit knowledge [13]. Tacit knowledge has its roots in individual experience and actions, as well as in ideals, values, emotions, intuition, ideas, and the subjective aspects of each person. Tacit knowledge is difficult to be imited or shared, and it is not easy for a competitor or any other person to steal or copy it, therefore it constitutes the core competitiveness of enterprises and forms the source of sustainable competitive advantage of enterprises [14].

The processes of the epistemological dimension (Socialization, Externalization, Combination and Internalization - SECI) constitute the engine for the creation of knowledge. Specifically, the socialization process involves sharing information and communicating tacit knowledge between persons [13] with a common culture working together effectively. Socialization generally begins with the creation of an interaction field, which allows members to share their experiences and mental models. But the most important part is how to find the right way to effectivevly express tacit knowledge not necessarily through writing [14]. The objective of the socialization process is to create and share the tacit knowledge into form of mental models and shared technical skills. However, this process by itself is a limited form of creating knowledge. Otherwise, the externalization process provides an explicit way for profiling and transfer of tacit knowledge, which usually covers the organization needs [14].

In software development organizations there are activities where it is essential to manage tacit knowledge properly. Problems arise where not all the concepts and their relationships are formally defined, or where the solutions for most of the problems are diverse, consensual and unverifiable, and finally where the roles use large amounts of tacit knowledge [8].

C. Knowledge perspectives approach

Organizations possess assets, skills, and capabilities [15]. An asset is understood as something, e.g. a brand or marketing point, that is better than the competition; meanwhile a skill is something that the organization's personnel makes better than their competitors. Knowledge may be seen as an intangible asset or resource that provides the capability of using information and applying it in the solution of a problem, making a decision or activities [16] that belong to the organization [15].

Knowledge is seen as a capability when it provides an explanation about the nature and structure of the organizational capabilities [7]. This happens when individuals combine their tacit knowledge in order to create an organizational capability [17]. The perception of knowledge as a capability suggests a perspective of KM focused on the basic competences: understanding a strategic advantage of knowing-

how (Theory of Resources and Capabilities) and the creation of intellectual capital [7]. For example, a requirement is a capability that a software system must provide or an attribute that this system must have in order to solve a problem.

If knowledge is seen as an object or is compared with access to information, then KM should focus on the construction and administration of knowledge stocks. However, if knowledge is a process then it's implied that KM should focus on knowledge flows and on the processes of creation, exchange, and distribution of knowledge. This is made evident in the four processes of SECI model.

D. Multi-perspectives of knowledge

The knowledge perspectives approach defined by Alavi and Leidner [7] is also associated to the SECI model and the Information Pyramid. Fig. 1 shows the flow of the SECI model's processes in relation to the DIK hierarchy. Data is found in the lowest level of unidirectional flow (represented by dotted lines) since it's meaningless by itself. Data collections represent information in accordance with the association measure or existing relations among them in order to generate judgment.

Alavi and Leidner specify that an individual is able to communicate her tacit knowledge to process, use and internalize the information, so she transforms such knowledge again (socialization) [7]. The higher the tacit knowledge an individual possesses, it's assumed that she possesses a wider combination of vision, principles, information, context, and experience [18].

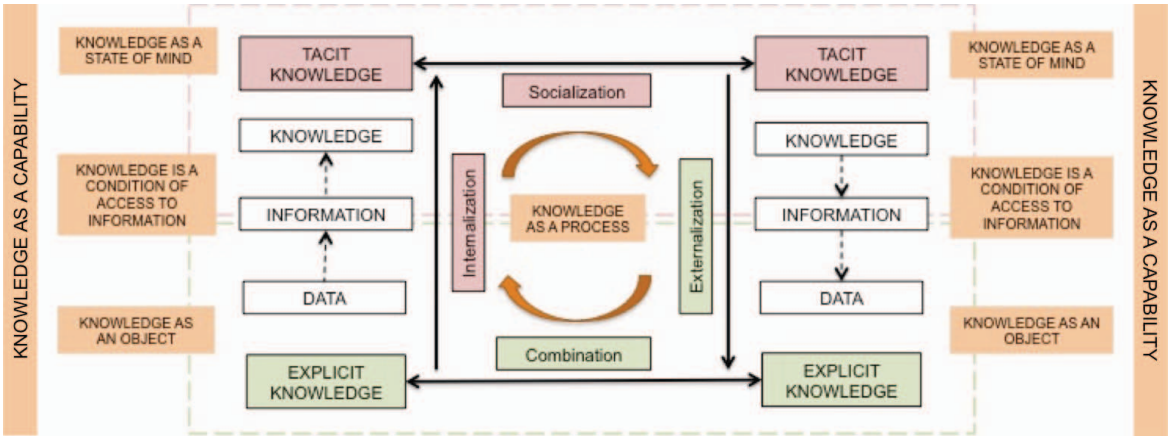

Fig. 1. Multi-perspectives of knowledge. Adapted from [18].

In addition, Fig. 1 shows a hierarchy with an opposite flow. Knowledge may have an information-regressive transition and turn from this to data; hence the importance of distinguishing each one in their respective categories. Tuomi argues that data exists as a solution to a practical problem, where knowledge is taken out of context and structured in accordance to a semantic [19]. Then, the process of externalization involves the formalization of tacit knowledge, turning it into information and data for its storage and management as products or objects.

It's important to note that the implementation of KM as a transversal process for supporting a software development organization aiming to high maturity levels, it is a way that has already been put into practice by some organizations [3, 4, 18]. The challenge of the current investigations is to analyze and measure the amount of tacit knowledge that can be captured, created, identified, represented, modeled, recovered, and shared.

III. KNOWLEDGE FLOWS

Some authors have seen knowledge flows as transfer of skills and technology between organizational subunits [20]. Knowledge flow has four key elements: knowledge, source, recipient, and context [21]. Source and recipient can be the same knowledge-processing participant. So, knowledge flow can carry and accumulate knowledge when it goes from one team member to another [22]. The relation of source and recipient indicates the direction of knowledge flow. The context represents the application environment wherein knowledge flow takes place. According to the definition, knowledge flow is always triggered by one or some tasks. These tasks may be knowledge-needs, business needs or customer demands [21]. Through knowledge flow, organizations can provide task-relevant knowledge to participants to fulfill their knowledge-needs [23].

Some knowledge flow models have been developed to classify and visualize various patterns of knowledge flow of significant and practical value in different situations [6, 22, 23, 24]. These models represent the dynamic aspects of knowledge processing, as reflected in various stages of knowledge lifecycle management [21].

IV. METHODOLOGY

The present work was performed following an action-research research method with a multi-cycle with bifurcation structure [25]. This method considers the following cycles: 1) Conceptual research cycle; 2) Methodological cycle; and 3) Validation cycle. The first cycle consisted on an analysis of the process of NFRE and its impact on the software development process.

In the Conceptual research cycle a preliminary process of approach to other research proposals that served the same purpose was carried out (Table 1). This approach made it possible to show that the proposals analyzed are not based on the understanding of the stakeholders regarding the concepts of non-functional requirements; i.e., they do not rely on the management or knowledge management of those involved in the requirements elicitation process.

TABLE I. EXISTING RESEARCH PROPOSALS

Ref.	Approach	Basis
[26]	Eliciting and prioritizing quality requirements supported by ontologies: A case study using the ElicitO framework and tool	Ontology and knowledge about the ISO/IEC 9126 Standard
[27]	NDR-Tool: A Tool to Support Knowledge Reuse in Non-Functional Requirements	Ontology and Knowledge Reuse
[28]	Making medical treatments resilient to technological disruptions in telemedicine systems	Hierarchy and Related Layers and Tuning of Knowledge
[29]	Supporting of requirements elicitation for ensuring services of information systems used for education	Specification and knowledge document adhering to ISO/IEC 9126 standard
[30]	Requirements engineering using mockups and prototyping tools: Developing a healthcare web-application	Prototyping and Knowledge Exchange
[31]	A systematic classification and analysis of NFRs	Multi-dimensional analysis of existing approaches
[32]	How do software architects consider non-functional requirements: An exploratory study	Perspective of the architects and software qualities
[33]	Towards a unified Requirements Modeling Language	Requirements traceability and visual modeling
[34]	A Formalism for Extending the NFR Framework to Support the Composition of the Goal Trees	Soft-gold and representation of requirements
[35]	Non-functional requirements elicitation and incorporation into class diagrams	UML methodology and specification document

A systematic mapping following the method published in [36] was made. From a total of 37 proposals, 14 were classified as theoretical, focusing on the RE process at a conceptual level and without validation results [11, 37, 38]; 17 dealt with tools and concepts related to RE; three were presented from an architectural perspective; and three were pertaining to KM [7, 39, 40].

Additional proposals were identified. One of them is the KARE approach (Knowledge Acquisition and Sharing for Requirement Engineering) [41], which provides a generic vision of the key RE processes, grouped as the activities of requirements gathering, analysis, and negotiation. Knowledge analysis and elicitation support each of the ER processes, allowing to transform tacit knowledge into explicit knowledge through the conceptual model. The knowledge flow model for requirements engineering (KFM-RE) proposed by [42] which it is an iterative cycle that consists of four stages: knowledge elicitation, model generation, model discussion and model validation. The authors use UML's notation for activity diagram.

Taking as a starting point the state of the art and the results of the systematic search, we perceive a lack of works that integrate the study of tacit knowledge to the process of NFRE. Thus, the concept of KM multiperspectives was included in a framework designed to help organizations in the process of NFRE.

V. MERLINN FRAMEWORK

The MERliNN framework was defined in the methodological cycle. It integrates the knowledge areas of RE and KM in a parallel way, providing a context to the TCER core (Knowledge Transformation in the Requirements Elicitation process).

This Framework aims to enable software development organizations to achieve global benefits in the later stages of the software development process, while providing a tailored elicitation process in earlier stages, as:

- Awareness of users about the importance and relevance of non-functional requirements at the time of validation of the quality of the specified product.
- Generate information about the architectural needs of the product at the time of its design.
- Greater confidence in the adequate generation of information (data quality) because the results generated by the system can be verifiable and traceable.
- Support for administrative processes based on the overall parameters of the system.
- Decrease in emergency events with respect to system availability in production environments.
- Own quantitative information related to product performance to include it in product test scenarios.
- Generate statistical information that allows the system administrator to follow up on non-functional aspects.
- A decrease on events reported to the organizations support group due to the ability of the product to indicate to the end user the reason for the errors.

A. Components

Fig. 2 shows the components of the MERliNN framework. The RE block emphasizes the NFR involved in the process, while the KM conceptual block organizes the different knowledge multiperspectives. In the middle, the TCER core defines the methods for knowledge transformation generated during the NFRE process, which are the basis for constructing the components of the framework. It is important to state that TCER does not describe the NFRE process workflows, but has as its purpose the analysis of the paths of the transformation of tacit knowledge flow within the process.

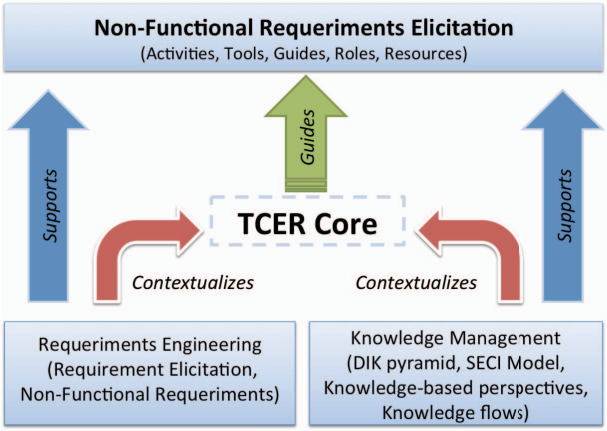

Fig. 2. Components of the MERliNN framework.

The SECI model, by means of knowledge flows, describes a spiral of knowledge that starts in the socialization process and goes through the rest of them until reaching the interiorization process, as shown in the center of Fig. 1. In MERLiNN, is advisable that knowledge takes new paths to be transformed according to the organizational dynamics related to the NFRE process. Such dynamics use different levels of knowledge from the involved roles and knowledge transformation flows, especially when referring to tacit knowledge [11].

There are three roles defined in the TCER core; these are shown in Fig. 3 and described next:

NFR elicitor: The person in charge of the NFRE process, possessing a tacit knowledge identified in the high-levels of communication with the end users.

Technical user: Person or group of persons that form the software product development team. Their tacit knowledge is related to experience, abilities, and knowledge they apply in activities like design, development, testing, deployment, database administration, architectural definition, functional support, infrastructural support, and telecommunications. This role establishes a close relationship with the tacit knowledge of the NFR elicitor, needed to identify, clarify, and complement the NFR.

End user: Person or group of persons that will use the software product and that perform operational or management tasks within the business context of the organization.

Dalkir identifies five levels of knowledge: novice, beginner, competent, expert and master [43]. MERliNN combines the first two levels as beginner, given that they describe manifestations of lack of knowledge, implying a misuse of knowledge due to not reasoning with it or being unaware of its existence [11].

The integral vision of the TCER core consists in finding the four strategic scenarios for knowledge transformation that are proposed in MERliNN: 1) knowledge instauration; 2) knowledge configuration; 3) knowledge consolidation; 4) knowledge institutionalization. These four scenarios can be visualized in Fig. 3, which combines the SPEM-KF symbology with an adapted rich picture [44]. Each scenario uses the KM multiperspectives to guide the development of the technical activities associated to NFRE, allowing to identify and track the direction of tacit and explicit knowledge flows among the roles taking part in the process. Given that the scope of this document is restricted to tacit knowledge, only the knowledge Instauration scenario (delimited by a blue rectangle) is described.

B. Knowledge Instauration in TCER Core

The scenario of knowledge Instauration starts when the NFR elicitor (source) contextualizes the stakeholders (recipient) by sharing data and information about what the NFR are, their importance, effects, and possible ways to identify them in order to develop a software product (Fig. 4). At that stage, the end user or technical user possess a lower level of tacit knowledge than the elicitor with respect to the domain of NFR. Thereafter, during the process of NFRE, the user can omit aspects, needs, and expectations of the software product.

As the user's tacit knowledge increases by means of the socialization process, her understanding about the NFR, concepts and uses also increases (knowledge as a state of mind). Consequently, the user acknowledges the importance of the internalization of the NFR. As a result of this first process of knowledge instauration towards the NFR, the end user, i.e. each individual involved in the process, obtains a tacit knowledge established in her mental scheme, creating more possibilities for understanding and use of the NFR in the RE processes carried by the organization.

In the scenario of Instauration the stakeholder will express, often verbally, the acquired understanding as a state of her mind, i.e. new tacit knowledge. Obtaining a clearer definition of the NFR, propitiates the flow of tacit knowledge and a combination of this new knowledge with previously acquired knowledge, thus making an impact in the achievement of quality in the software products (knowledge as a capability). This description is presented in Table II.

TABLE II. DESCRIPTION OF THE SCENARIO OF KNOWLEDGE INSTAURATION

Scenario of knowledge instauration		
Flows of transformation tacit knowledge	***Roles***	***Technical stage of NFR***
Socialization, Exteriorization	NFR Elicitor End User	Identification
Description: This scenario implies the NFR elicitor or group of elicitors must make considerable efforts in the analysis of knowledge as a state of mind, as the stakeholders possess a low level, or even a lack of the required tacit knowledge for an effective identification of the NFR. In the socialization flow, the end users that take part in the NFRE process, confirm what they know and understand for a specific NFR. Meanwhile, the elicitor validates, corrects, complements, and confirms what the interviewees are expressing about their understanding of the NFR for the software product. The elicitor uses diverse mechanisms for registration of the identified NFR; i.e. make use of the tacit knowledge exteriorization flow.		
Techniques: Brainstorming, meetings for the establishment of concept equivalence, didactic workshops, role games, use of analogies, experience sharing, knowledge replication by means of narratives and role imitation.		

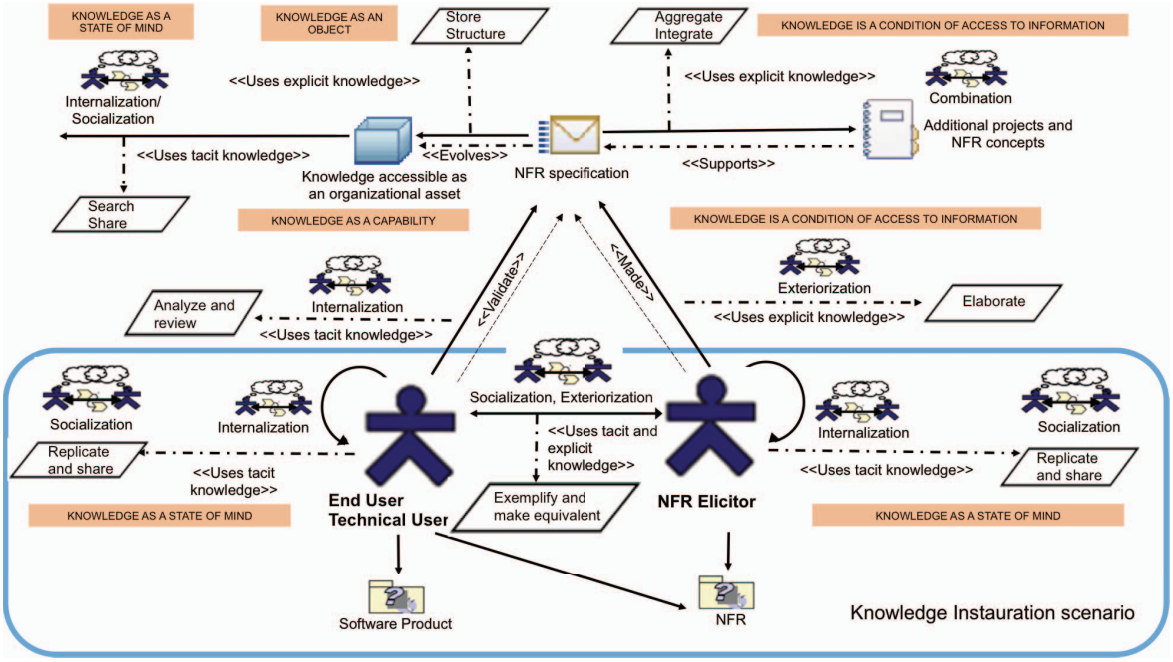

Fig. 3. Formal model for TCER core.

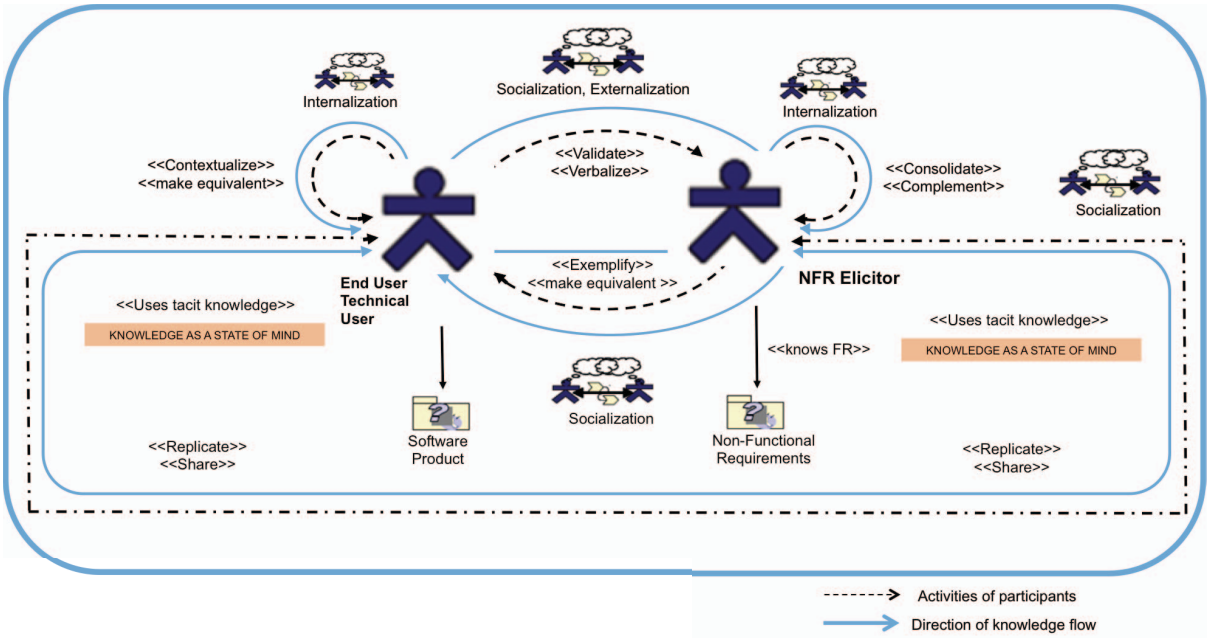

Fig. 4. The scenario of knowledge Instauration for NFRE.

Technical stage is adapted from ISO 12207 and ISO 15504 standards. Identification of NFR is the phase where the NFR are made known to stakeholders in a way that clearly states the constraints and quality expectations of the software product.

VI. VALIDATION

In this section, a preliminary validation method for the MERLiNN framework is introduced. Validation is conducted through a single-embedded case study as proposed by [45] performing the activities proposed by [46]. The framework was applied in three projects (analysis unit - AU1, AU2, and AU3) in a two years old software development organization in Colombia, with 12 employees and 6 completed software projects.

A. Design of the case study

The objective of the case study was to analyze and validate the transformation of the tacit knowledge on NFR that the roles possess, and how this knowledge flows during the NFRE process. The mechanisms used are: (1) observation of aspects identified during the intervention; (2) conducting semi-structured interviews; and (3) a survey to gather qualitative and quantitative information about the NFR and stakeholders involved in the NFRE process.

Two types of variables were defined: participative and empirical, as suggested in [47]. The participative variable is related to the members of the technical team in the organization, who showed a legitimate interest in the correct usage of the work products offered by MERLiNN in their respective analysis units. On the other hand, the empirical variable was manifested in the guidance the knowledge worker, as KN and NFR advisor, provided to the different project leaders while seeking to execute the instauration process in a systematic and organized NFRE process. As a result, the subjects of the investigation were: (1) one expert adviser on NFRE for software products who would provide support in the contextualization of tacit knowledge, and guide the NFRE in the three selected analysis units; (2) three technical teams, one for each analysis unit; and (3) a group of operative end users and chiefs of area with whom the elicitors implemented the KM activities proposed by MERLiNN. It is important to note that the technical team had tacit knowledge pertaining to multiple roles assigned to the analysis units, as designers, architects, testers, and developers.

B. Intervention

The three analysis units had different subjects and problematics. It took seven weeks to carry the intervention process, gathering the required measurements (Table III) and detecting the differentiating aspects. All the analysis units follow the SCRUM methodology, without gathering or documenting NFR in their user stories.

In the case of knowledge flows, this case study helped demonstrate that: (1) the flow of socialization was used at the time the expert adviser imparted training on NFR to project leaders and members of the technical team of the three units of analysis. (2) the flow of externalization was used in the units of analysis through explicit information about: (i) Type of client organization, (ii) Number of internal and external persons involved in the development project, (iii) (iv) Interfaces to be taken into account for the development project, (v) Forms of knowledge management in the client organization to be used at the time of the FNS, (vi) Number and knowledege level on NFR of the stakeholders (vii) External stakeholders of a legal nature that could include or suggest restrictions on the software product, (viii) NFR of software products and (ix) Mechanisms to manage NFR knowledge in each project, (3) the flow of combination was permanently evidenced through the consultations made by the advisors to the expert adviser when they had some concern about the way in which they were carrying out the process of identification and specification of NR, and finally, (4) the flow of internalization occurred through the employees expressing their concerns about concepts related to NFR and the components of MERLiNN, which became every time more specific, demonstrating the obtainment of a greater understanding and dominion of these concepts on NFR by the elicitors. These processes of knowledge transformation allowed obtaining a significant amount of NFR in each one of the units of analysis involved (Table III).

TABLE III. MEASURES OBTAINED

Metrics	*AU1*	*AU2*	*AU3*
Number of NFR identified	20	63	49
Number of stakeholders involved in the NFRE process	4	1	28

This compilation of NFR shows that through the framework, the ambiguity of the NFRs that must be implemented in the information systems to be designed and constructed is diminished. Similarly, the use of framework processes could avoid requiring the reprocessing of NFR elicitation, since the information is condensed into the specific format provided by the framework for that purpose. Likewise, knowledge management enables the knowledge extracted through the elicitation process to be accessed by other projects as shown in Fig. 4 of the installation scenario. For the case study, this accessibility occurred with the use of the Drive tool, through which the organization published the NFR specification obtained for the intervened analysis units and additional information resulting from the executed elicitation process. During the intervention process, the occurrence of the four scenarios of MERLiNN was identified. For the scenario of knowledge instauration, the tacit knowledge socialization activities among the elicitors and technical leaders of the three analysis units were performed.

C. Analysis

The socialization flow occurred when the knowledge worker trained the project leaders and part of the technical teams on NFR. Through the exteriorization flows, several data and information were obtained: 1) the number of stakeholders and their level of tacit knowledge on NFR; 2) external interests that could express or impose legal restrictions on the software product; 3) mechanisms of exteriorization to manage the NFR

knowledge in each project.

Throughout training, the NFR elicitors asked the advisor for a deeper level of exemplification, replication, and abstraction of the NFR using techniques defined for the scenario of knowledge instauration (Table I). Therefore, the technical team assigned to each project would be able to increase its tacit knowledge. The identified NFR were related to maintainability, portability, reliability, and compatibility of the system. These NFR categories demand a high level of technical knowledge, previous experience with databases related issues, data recovery mechanisms, and integration with other software tools. Therefore, there was a high probability for these requirements to miss being specified in the initial iterations of the process.

When the interviews were applied, an increase of the tacit knowledge was detected in the participants: 1) two of them manifested an increase of 3 levels of knowledge; 2) another two an increase of 2 levels; and 3) one participant increased 1 level. This increase in their knowledge level supplies the participants with more tacit knowledge to socialize and exteriorize in the posterior iterations of the project.

Within the validation plan, mechanisms were designed for: 1) validating the construct for the interview and survey's questions, allowing the traceability among the subject of study, the collected data, and data analysis; 2) validate the TCER core in MERliNN; and 3) a focus group-based validation of the TCER core.

VII. CONCLUSIONS

The lack of mechanisms of the Non-Functional Requirements (NFR) elicitation process and the non-standard procedures for establishment of them by the stakeholders and the software development organizations propitiates a loss of tacit knowledge for the correct definition of the characteristics of the software product. In this document, we have introduced general concepts regarding Knowledge Management (KM), knowledge multi-perspectives to explain how tacit knowledge flows within NFR elicitation process.

The MERliNN framework is based on multi-perspectives of KM that support the analysis of the flow of tacit knowledge within the phase of Engineering of Requirements. The TCER core guides and supports the MERliNN framework to be instrumented through processes, activities, tasks, work products, roles, execution guides, and templates for NFR.

MERliNN was implemented in a case study with three analysis units (projects) in a Colombian software development organization. The results show that the members of the technical work team increased their level of tacit knowledge, which enables them to socialize and externalize in later iterations. The NFR identified in the three analysis units were maintainability, portability, reliability, and compatibility of the system. Currently, a technological tool is being developed that supports the application of MERliNN framework.

ACKNOWLEDGMENT

This work is being supported by Universidad del Cauca (UNICAUCA) Colombia, and Universidad Autónoma de Baja California (UABC) México. Financial support was provided by PRODEP-SEP.

REFERENCES

[1] R. Anaya, A. Cechich, and M. Henao, A Model to classify knowledge assets of a process oriented development, In Software process improvement for small and medium enterprises: techniques and case studies, H. Oktaba and M. Piattini, Eds. New York: Information Science Reference, 2008, pp. 280-295.

[2] B. L., Flores Rios and O. M., Rodríguez Elías, Experience factory infrastructure as a basis for knowledge manegement in a software process improvement program, In Ingeniería de software e ingeniería del conocimiento: tendencias de investigación e innovación tecnológica en Iberoamérica, R. A. Aguilar Vera et al. Eds. México: Alfaomega, 2010, pp. 174-183.

[3] J. Capote, C. J. Llanten Astaiza, C. J. Pardo Calvache, A. de J., González Ramírez, and C. A., Collazos, "Gestión del Conocimiento como apoyo para la mejora de procesos software en las micro, pequeñas y medianas empresas," Revista Ingeniería e investigación, vol 28 (1), 2008, pp. 137-145,

[4] J. Mejia Miranda, M. A. Muñoz Mata, G. Rojo Cruz, H. E. Ramírez Luna, I. E. Tinajero Díaz, and J. M. García Ramírez, "Extracción del conocimiento tácito para la mejora de procesos software: una experiencia en un organismo gubernamental," Sistemas y Tecnologías de Información CISTI, vol 1, 2014, pp. 423-429.

[5] V. Ambrosini and C. Bowman, ''Surfacing tacit sources of success,'' International Small Business Journal, vol. 26 no. 4, pp. 403-31, 2008.

[6] B. L. Flores-Rios Pino, F. J., Ibarra-Esquer, J. E., González-Navarro, F. F. and Rodríguez-Elías, O. M., "Análisis de Flujos de Conocimiento en Proyectos de Mejora de Procesos Software bajo una perspectiva multi-enfoque," Revista Ibérica de Sistemas e Tecnologias de Informação, 14, 2014, pp. 51-66.

[7] M. Alavi and D. E. Leidner, "Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues," MIS quarterly, 25 (1), 2001, pp. 107-136.

[8] K. Olmos and J. Rodas, "KMoS-RE: knowledge management on a strategy to requirements engineering," Requirements Eng., 19, 2014, pp. 421-440.

[9] D. Pandey, U. Suman, and A. Ramani, "An effective requirement engineering process model for software development and requirements management," in ARTCom, Oct 2010.

[10] L. Chung, J. Cesar and P. Leite, "On Non-functional Requirements in Software Engineering," In Conceptual modeling: Foundations and applications, pp. 363-379, 2009.

[11] S. L. Buitrón Ruiz, Marco de trabajo para la Elicitación de Requisitos No Funcionales basada en la Gestión de Conocimiento MERliNN. Universidad del Cauca. Masters Thesis. Colombia, 2016.

[12] T. Davenport and P. Prusak, Conocimiento en acción: Como las organizaciones manejan lo que saben. Traducido del Inglés por Moreno J.M. Buenos Aires 1ª Ed: Pearson Education, S.A., 2001.

[13] I. Nonaka and H. Takeuchi, La organización creadora del conocimiento: Cómo las compañías japonesas crean la dinámica de la innovación. México: Oxford University Press, 1999.

[14] C. Chen, Z. Zhang, L. Chen and L. Yongbo, "Research on the factors of the process of the tacit knowledge in knowledge service enterprises," Open Cybernetics & Systemics Journal, vol. 9, 2015, pp. 1776-1782.

[15] H. Itami and T. Roehl, Mobilizing Invisible Assets, Cambridge, Harvard University Press, 1987.

[16] S. A. Carlsson, "Towards an understanding and conceptualization of knowledge managing within the context of inter-organizational networks," Third European Conference on Organizational Knowledge, Learning, and Capabilities (OKLC 2002), 2002.

[17] M. B. Lloria, "A review of the main approaches to knowledge management," Knowledge Management Research & Practice, vol. 6 No. 1, 2008, pp. 77-89.

[18] B. L. Flores Rios. Modelo de evolución de la gestión de conocimiento en MiPyMEs, de acuerdo con el nivel de madurez en un programa de mejora de procesos de software. UABC, 2016. México.

[19] I. Tuomi, "Data is More Than Knowledge: Implications of the Reversed

Hierarchy for Knowledge Management and Organizational Memory," in Proceedings of the Thirty-Second Hawaii International Conference on Systems Sciences, IEEE Computer Society Press, Los Alamitos, CA, 1999.

[20] A. Gupta and V. Govindarajan, "Knowledge flows within multinational corporations," Strategic Management Journal, vol. 21, 4, 2000, pp. 473-496.

[21] J. M. Guo and Y. L. Wang, "Context modeling for knowledge flow," Proceedings of 2008 IEEE International Conference on Information Reuse and Integration, IEEE IRI-2008, Institute of Electrical and Electronics Engineers Computer Society, Piscataway, NJ 08855-1331, USA, 2008, pp. 330-335.

[22] H. Zhuge, A knowledge flow model for peer-to-peer team knowledge sharing and management, Expert Systems with Applications, vol. 23 (1), 2002, pp. 23–30.

[23] D-R, Liu and C-W, Lin, "Modeling the knowledge-flow view for collaborative knowledge support," Knowledge-Based Systems, vol. 31, 2012, pp. 41-54.

[24] B. H. Hansen and K. Kautz, "Knowledge mapping: a technique for identifying knowledge flows in software organisations," in Lecture Notes in Computer Science, vol. 3281, T. Dinsoyr, Ed., Heidelberg:Springer Berlin, 2004, pp. 126-137.

[25] F. J. Pino, M. Piattini and G. Horta Travassos, "Managing and developing distributed research projects in software engineering by means of action-research," Revista Facultad de Ingeniería Universidad de Antioquia, 68, 2013, pp. 61-74.

[26] T. H. Al Balushi, P. R. F. Sampaio and P. Loucopoulos, "Eliciting and prioritizing quality requirements supported by ontologies: a case study using the ElicitO framework and tool," Expert Systems, vol. 30 (2), 2013, pp. 129–151.

[27] A. L. de Araújo, L. M. Cysneiros, and V. M. B. Werneck, "NDR-Tool: Uma Ferramenta de Apoio ao Reuso de Conhecimento em Requisitos Não Funcionais," in Anais do WER14, Pucón, Chile, 2014.

[28] N. Larburu, R. G. Bults, and H. J. Hermens, "Making medical treatments resilient to technological disruptions in telemedicine systems," 2014 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), June 2014, pp. 285-288.

[29] Y. Terawaki, "Supporting of requirements elicitation for ensuring services of information systems used for education," Human Interface and the Management of Information. Interacting with Information, 2011, pp. 58-65.

[30] L. Teixeira, V. Saavedra, C. Ferreira, J. Simões, B. Sousa Santos, "Requirements Engineering Using Mockups and Prototyping Tools: Developing a Healthcare Web-Application," In: Yamamoto S. (eds) Human Interface and the Management of Information, Information and Knowledge Design and Evaluation. HIMI 2014. Lecture Notes in Computer Science, vol 8521, pp. 652–663, 2014.

[31] P. Loucopoulos, J. Sun, L. Zhao, and F. Heidari, "A systematic classification and analysis of NFRs," 19th American conference on information systems (AMCIS 2013), Chicago: USA AIS 2013.

[32] D. Ameller, C. Ayala, J. Cabot, and X. Franch, "How do software architects consider non-functional requirements: An exploratory study," In Requirements Engineering Conference (RE), 2012 20th IEEE International, 2012, pp. 41-50.

[33] J. Helming, M. Koegel, F. Schneider, M. Haeger, C. Kaminski, B. Bruegge, and B. Berenbach, "Towards a unified Requirements Modeling Language," 2010 Fifth International Workshop on Requirements Engineering Visualization, pp. 53–57, 2010.

[34] B. Wei, Z. Jin, and L. Liu, "A formalism for extending the NFR Framework to support the composition of the goal trees," In Software Engineering Conference (APSEC), 2010 17th Asia Pacific, 2010, pp. 23-32.

[35] X. Song, Z. Duan, and C. Tian, C. Non-Functional Requirements Elicitation and Incorporation into Class Diagrams. Intelligent Information Processing V, pp. 72-81, 2010.

[36] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE), vol. 8, 2008, pp 71–80.

[37] E. Serna-Montoya, Estado actual de la investigación en requisitos no funcionales, vol. 16 (1), pp. 225-246, 2012.

[38] M. A. Jabar and F. Sidi, "The effect of organizational justice and social interdependence on knowledge sharing," International Conference on Information Retrieval & Knowledge Management, 2012, pp. 64-68.

[39] E. Amaru Galvis, Modelo de Referencia de procesos de gestión de conocimiento para organizaciones de desarrollo de software de Colombia. vol. 1. Universidad Nacional de Colombia, Bogotá, 2013.

[40] A. de J González, C. Z. Joaquí and C. A. Collazos, "KARAGABI KMMODEL: Modelo de referencia para la introducción de iniciativas de gestión del conocimiento en organizaciones basadas en conocimiento," Ingeniare - Revista chilena de ingeniería, vol. 17(2), 2009, pp. 223-235.

[41] S. Ratchev, "Knowledge-enriched Requirement Specification for One-of-a-kind Complex Systems," Concurr. Eng., vol. 13, no. 3, Sep. 2005, pp. 171–183.

[42] K. Olmos, J. Rodas and L. F. Fernández, "Requirements engineering knowledge model for informal structured domains," Int J Comp Commun Eng. vol. 2(1), 2013, pp. 75–77.

[43] K. Dalkir, Knowledge management in theory and practice. Elsevier. Amsterdam. 356, p. 2005.

[44] O. M. Rodríguez-Elias and A. I. Martínez G., Diseño de sistemas y estrategias de gestión del conocimiento: Un enfoque metodológico orientado a procesos y flujos de conocimiento, Editorial Académica Española, 236 p., 2011.

[45] R. K. Yin. Case study research. Design and methods. Newbury Park: Stage Publications, 2013.

[46] Runeson and Martin Host. Guidelines for conducting and reporting case study research in software engineering. Empirical Software. Eng. vol. 14(2), April 2009, pp. 131-164.

[47] W. L. French and C. H. Bell. Organization development: Behavioral science interventions for organization improvement London. Pearson, 6th ed., 1999.