*Journal of Universal Computer Science, vol. 23, no. 4 (2017), 385-403*
*submitted: 27/9/16, accepted: 29/3/17, appeared: 28/4/17* © *J.UCS*

# Selecting a Software Elicitation Technique According to Layers of Knowledge and Preciseness: A Case Study

**Maria-Isabel Sanchez-Segura**
(Carlos III University, Madrid, Spain
misanche@inf.uc3m.es)

**Fuensanta Medina-Dominguez**
(Carlos III University, Madrid, Spain
fmedina@inf.uc3m.es)

**Diana-Marcela Vásquez-Bravo**
(Carlos III University, Madrid, Spain
Diana.marcela.vasquez@gmail.com)

**Gustavo Illescas**
(Universidad Nacional del Centro de la Provincia de Buenos Aires, Tandil, Argentina
illescas@exa.unicen.edu.ar)

**Cynthya García de Jesús**
(Carlos III University, Madrid, Spain
cygaje@gmail.com)

**Abstract:** This paper presents a case study analyzing a set of software engineering elicitation techniques. The aim of the case study is to demonstrate that completeness and preciseness are two criteria to be incorporated into the set of existing parameters used to classify and select which elicitation technique to apply depending on the project context variables. Completeness refers to how well each elicitation technique elicits domain, task and strategic requirements, and preciseness refers to how many requirements a software engineer is able to elicit using each technique. Based on the results, we can state that completeness and preciseness perform differently for each analyzed technique. Therefore, these two criteria are necessary in order to improve elicitation technique selection. Also, the techniques used in this case study have been ranked according to the above-mentioned criteria, that is, which technique included in this study, is best suited for which requirements layer and which technique can be expected to elicit most requirements during the knowledge externalization phase.

**Keywords:** Elicitation techniques, layers of knowledge, software engineering, knowledge elicitation
**Categories:** D.2, D.2.0, D.2.1

## 1 Introduction

The software industry is knowledge intensive [Tiwari, 2012]. A software engineer's goal is to transform a customer's problem into a solution that has to be implemented as a software product or service. The first and critical phase during the development of a software product that satisfies customer needs is called elicitation, when the software engineer has to:

- Deal with the need that a customer has envisaged and which must be elicited using the right technique.
- Deal with experts in a specific field in which a software system is to be deployed to assure that the system operates like a human expert in the field.

Note that the elicitation process is not just a technical process prior to system construction; it should also be seen as a social process involving different individuals and groups of people with different levels of knowledge, abstraction and communication.

Requirements elicitation is considered to be a key requirements engineering activity. It is complex [Razali, 2012], and it has been proven that incorrect requirements elicitation leads to project failure [Kausar, 2010]. So, the software engineering discipline has worked hard to provide a set of elicitation techniques to manage the knowledge to be elicited. Knowledge elicitation involves acquiring and transferring knowledge from individuals (as it exists in the minds of experts in a specific domain) to an abstract and effective representation for the purposes of organization, modeling and ultimately statement in an understandable and reusable format. This means transforming tacit into explicit knowledge. There follows a definition of these two types of knowledge:

- Tacit knowledge: highly informal, personal, unverbalized, intuitive knowledge derived from experience.
- Explicit knowledge: formal and systematic knowledge that can be expressed without ambiguities in narrative descriptions, mental maps, diagrams, databases, etc.

The process of transforming tacit into explicit knowledge represented as concepts models and specified as best practices or lessons learned is what, in the field of knowledge management, Nonaka refers to as externalization [Nonaka, 2007]. So, in knowledge management terms, the software engineering elicitation process is equivalent to knowledge externalization through its transformation from tacit into explicit knowledge.

The externalization process, and by extension the elicitation process, is very complex because there are many factors that interfere in this process. Therefore, a lot of research has been conducted on the definition of techniques that can facilitate the elicitation process depending on the conditions in which the knowledge is gathered. It is commonly accepted that one technique cannot possibly account for all conditions, contexts and types of knowledge.

There is some interesting research (see Section 2 for more details) concerning the identification of criteria that are helpful for selecting the best elicitation technique [Tiwari, 2012][Razali, 2012][Kausar, 2010][Carrizo, 2014] [Carrizo, 2016].

Although all these papers present effective and validated proposals, none of them take into account the elicitation technique preciseness in terms of the number of requirements that the software engineer is able to elicit using each technique, against the expected number of requirements that could be elicited, neither the fact that there are different layers of knowledge to be elicited: strategic, domain and task knowledge [Schreiber, 1994][Wielinga, 1992].

- The layer of strategic knowledge is related to decision making, task selection and specialized knowledge.
- The layer of task knowledge is related to experience in the development of procedures, tasks and activities.
- Finally, the layer of domain knowledge is related to concepts, structures and their relationships within a specific domain.

For both of these reasons, we developed this case study in order to visualize the completeness and preciseness of at least one set of elicitation techniques useful in the externalization phase of a software project elicitation process. This case study is intended to discover whether or not a selected elicitation technique provides the same coverage of different types of requirements, accounting for different layers of knowledge, and the same preciseness regarding the number of elicited requirements. If the results reveal that each technique elicits different requirements for each knowledge layer and/or the preciseness varies from one technique to another, then two more criteria should be included in order to select the elicitation technique to be applied in each project: the knowledge layer for which the elicitor wants to elicit requirements and the expected preciseness of the elicited requirements.

The remainder of this paper is structured as follows. Section 2 focuses on the groundwork of this research, that is, work related to solving the problem of selecting the appropriate software engineering requirements elicitation techniques. Section 3 describes the selected software engineering elicitation techniques. Section 4 describes the case study carried out, that is, details the procedure enacted for the purposes of validation and discussion of the results. Finally, Section 5 outlines the conclusions and future work.

## 2 Related Works

Considering the diversity of existing elicitation techniques, software engineers need objective, clear and measurable criteria to analyze and decide which technique to use in the projects in which they participate. The elicitation technique classifications in the literature are more theoretical than practical. The analyzed papers identify factors such as the problem domain, solution domain, user types or characteristics or project phases in order to select the appropriate elicitation technique [Booch, 2008] [Browne, 2008] [Chung, 2000] [Hickey, 2003a][Hickey, 2003b] [Heninger, 1980] [Hull, 2005] [Jackson, 2001][Lauesen, 2002][Mylopoulos, 2001][Nuseibeh, 2000] [Paetsch, 2013][Pressman, 2015][Sawyer, 1999] [Van Lamsweerde, 2000][Wiegers, 1999] [Beecham, 2005] [Davis, 2002][Hickey, 2003c][Lauesen, 2002][Leffingwell, 2000][Maiden, 1996].

Some of the proposed elicitation methodologies, like Robertson's requirements methodology, include a process model that details the activities of requirements capture, with inputs, outputs and recommended techniques for each activity [Robertson, 1999].

Other authors have defined specific process models, establishing the use of scenarios for requirements elicitation [Hsia, 1994] [Holbrook, 1990]. There are also general top-down elicitation approaches [Pressman, 2015][Gause, 1989], but very few provide a process view approach.

Carrizo et al. [Carrizo, 2014] analyze and classify ten key papers in the field of elicitation techniques according to a set of nine aspects. After classification, they found that all existing papers failed to afford significant and systematic assistance for requirements engineering practitioners and concluded that a new framework was required for proper elicitation technique selection. Carrizo et al. reported very interesting research on project, elicitor and stakeholder criteria for deciding which elicitation technique to select. In some other works of Carrizo [Carrizo, 2015] [Carrizo, 2016] there are an extensive analysis of the literature around existing criteria for the selection of an elicitation technique and it can be confirmed that the criteria validated in this case study are not listed among the ones identified, and with this work we are going to demonstrate the importance to incorporate them to the list of existing ones.

Razali [Razali, 2012] provides a set of parameters to select the appropriate elicitation technique; analyst´s experiences, ease of use, speed, cost and stakeholder´s preferences, but the proposal, although promising, has only been partially tested.

Tiwari [Tiwari, 2012] bases elicitation technique selection on seven variables: type of stakeholder, social environment, nature of the system, type of user, scope of the system, analyst ability/skills, enacted methodological approach.

As a summary, we can affirm that there are two criteria missing in all the reviewed papers that we consider might be relevant for elicitation technique selection:

- Different layers of knowledge to be elicited: it is important not to forget that the elicitation process deals with knowledge, and knowledge can be stratified into three layers —strategic, domain and task knowledge—, which the elicitor must take into account.
- Number of elicited requirements: one technique could be more precise than the others so the technique used to develop the elicitation process will depend on the criticality of the project.

This case study is intended to demonstrate that these two criteria should be considered relevant for the selection of the appropriate elicitation technique.

## 3 Selected Software Engineering Elicitation Techniques for Knowledge Externalization

Based on the review of the literature on elicitation techniques, Table 1 summarizes some of the most widespread elicitation techniques, whose use has been empirically validated.

Column 1 Classification shows the categories used to group the requirements elicitation techniques provided by software engineering literature; Column 2 Techniques names each of these techniques; and, finally, Column 3 Knowledge Creation Phase connects the elicitation technique categories with the phase of the model for knowledge creation [Nonaka, 2007] in which they can be applied. Those phases are:

- Socialization (S): conversion of tacit knowledge into explicit knowledge through knowledge transfer from one person to another by means of social interaction and shared experiences among the members of the organization.
- Externalization (E): conversion of tacit knowledge into explicit knowledge represented as concepts, models and articulated as best practices or lessons learned.
- Combination (C): conversion of existing knowledge into new evolved explicit knowledge by categorizing, reclassifying and synthesizing existing explicit knowledge.
- Internalization (I): conversion of tacit knowledge into explicit knowledge so that new knowledge can be developed.

This classification is used to select the techniques that will be studied in this research. In order to delimit the scope of the case study, we will use a subset of techniques that are applicable in the knowledge externalization phase (marked with E), because it is the phase most akin to the elicitation process:

- Structured elicitation techniques neatly lead the knowledge acquisition process and guide the interviewee under certain circumstances. They also contribute to the clarification and classification of concepts. The selected techniques for this category are Structured Interviews and Product Patterns.
- Structured and unstructured elicitation techniques satisfy the need to elicit knowledge from people for each problem domain. The selected technique for this category is Brainstorming.
- Representation or mapping techniques benefit the externalization of knowledge by mind mapping and establishing correlations, hierarchies and diagrams of items of knowledge. The selected technique for this category is Concept Mapping.

For the purpose of this work, were analysed the performance of a set of techniques classified as suitable for externalization phase (E). A random selection was performed among all the techniques identified in Table 1 as suitable for the purpose of this work: The selected techniques are described next.

- Structured Interview: In a structured interview, the purpose of the interview and questions should be prepared beforehand. The questionnaire question design and clustering should be logical with respect to the actions or processes that have been identified in the organization and are to be explored. Interview preparation is very time consuming and can contain ambiguous language problems, but it is an excellent way to thoroughly examine particular situations.
- Product Pattern: The term product pattern, coined by Medina-Dominguez et al. [Medina-Dominguez, 2008], encapsulates the knowledge and experience related to the development of any software product, being understood as anything produced during the software life cycle. Product patterns facilitate knowledge representation since they correspond to a reusable generalization or abstraction that can be used as a starting point for future solutions [Medina-Dominguez, 2009]. Besides being an element that enables knowledge encapsulation and reuse, we have used product patterns here as a knowledge elicitation technique, enabling knowledge elicitation in the externalization phase of the model for knowledge creation [Nonaka, 2007].
- Brainstorming: This is a group technique that makes it possible to take into account all the opinions that spring to participants' minds. All the ideas about a topic or particular problem are analyzed and discussed in groups. This technique was created by Alex Osborne in 1941, when his search for creative ideas resulted in an

interactive unstructured group process, generating more and better ideas than could be produced by individuals working independently.

- Concept Mapping: A concept map is a diagram showing the relationships between different concepts. This technique is a justified option, because the tacit knowledge that resides in the minds of experts and an organization's employees can be converted into explicit knowledge using models or concepts or by specifying their relationships.

**Nonaka knowledge creation phases**

**S: Socialization, E: Externalization, C: Combination, I: Internalization**

| **Classification** | **Techniques** | **Knowledge Creation Phase** |
|---|---|---|
| Structured Elicitation Techniques | Card sorting | E, C, I |
| | Scenarios | |
| | Structured Interview | |
| | Product Pattern | |
| | Critical success factor | |
| Unstructured Elicitation Techniques | Unstructured Interview | S, E, I |
| | Brainstorming: | |
| Formal analysis Techniques | Text analysis | C |
| | Repertory grid | |
| Representation or mapping techniques | Multi-dimensional scaling | E, C |
| | Concept mapping | |
| Observation-based techniques | Task analysis | S, I |
| | Participant observation | |
| | Behaviour analysis | |
| | Protocol analysis | |
| | Prototyping | |

*Table 1: Classification and Summary of the Characteristics of Elicitation Techniques.*

## 4 Case Study Description

The goal of this section is to compare the completeness and the preciseness of the requirements elicited using the selected elicitation techniques. The research method that we used to do this was the case study [Yin, 2013] [Runeson, 2009].

Table 2 summarizes the indicators to be measured in this case study. The values of the RESRi, RETRi and REDRi indicators for each technique applied in requirements elicitation are calculated as a ratio of the units of each type of specific requirement (domain, strategic or task knowledge) that a software engineer is able to elicit from the client using a technique to the number of requirements of the respective type that a software engineer is expected to elicit. Table 4 summarizes these indicators for this case study.

| EXTERNALIZATION PHASE | | | | |
|---|---|---|---|---|
| **Completeness** | | | **Preciseness** | |
| **RESRi** *(Ratio of Elicited Strategic Requirements)* | **RETRi** *(Ratio of Elicited Task Requirements)* | **REDRi** *(Ratio of Elicited Domain Requirements)* | **REi** *(Requirements Elicited)* | **Pi** *(Preciseness)* |
| Strategic requirements layer.<br><br>Extent to which technique i facilitates the description of strategic requirements<br><br>Related to decision making, task selection and specialized knowledge | Task requirements layer.<br><br>Extent to which technique i facilitates the description of task requirements<br><br>Related to experience in the development of procedures, tasks and activities | Domain requirements layer<br><br>Extent to which technique i facilitates the description of domain requirements<br><br>Related to concepts, structures and their relationships within a specific domain | Total amount of requirements acquired in the elicitation session with technique i | Percentage of requirements that a software engineer is able to express using technique i over the total number of requirements expected to be elicited |

*Table 2: Indicators to measure the selected techniques*

There follows an example of each type of requirement taken from one of the projects involved in this case study concerning the management of a transactive memory system:

- **Domain requirements example (REQUIREMENT- D01):** The content stored in the transactive memory system (TMS) should be dynamic, so it should be continuously updated.
- **Task requirements example (REQUIREMENT – T01):** The TMS members should be able to add non-structured knowledge to the system, like paper notes, podcast or videos.
- **Strategic requirements example (REQUIREMENT – S01):** No TMS member can modify a version that he or she has not created, so any modification made in the knowledge capsule should be saved as a different version.

The above-mentioned indicators are represented by the following formulas:

**REDRi= REi_DRi *100 /ERE_DRi**
**RESRi= REi_SRi *100 /ERE_SRi**
**RETRi= REi_TRi *100 /ERE_TRi,**

where

**REi_DRi** = number of **R**equirements **E**licited with each technique for the **D**omain **R**equirements layer
**REi_SRi**= number of **R**equirements **E**licited with each technique for the **S**trategic **R**equirements layer
**REi_TRi**= number of **R**equirements **E**licited with each technique for the **T**ask **R**equirements layer
**ERE_DRi**= number of **E**xpected **R**equirements **E**licited for the **D**omain **R**equirements layer
**ERE_SRi**= number of **E**xpected **R**equirements **E**licited for the **S**trategic **R**equirements layer
**ERE_TRi**= number of **E**xpected **R**equirements **E**licited for the **T**ask **R**equirements layer

The REDRi, RESRi and RETRi indicators are expressed as a percentage and are calculated for each of the four selected techniques and for each one of the 10 participants in the case study (see Table 4).

The preciseness of the externalized knowledge or elicited requirements in terms of software engineering is calculated by the Pi indicator. This indicator is calculated as a ratio of the requirements that a software engineer is capable of eliciting using each of the techniques. In order to calculate the Pi values for each technique, we have to calculate the values for the REi variable. REi (requirements elicited) represents the requirements resulting from the elicitation session using technique i for each subject.

The Pi ratio is calculated as the percentage of requirements elicited over the total number of requirements expected to be elicited using technique i. To do this, we have to count the number of requirements that the software engineer understands using each technique and the number of requirements that the software engineer is expected to elicit during the elicitation session.

The expected requirements from an externalization session for each project in this case study are equivalent to a constant value of 15 requirements. These 15 requirements have been divided into three layers: specialized or decision-making knowledge describing plans (strategic requirements), procedural knowledge about how to do tasks and/or activities (task requirements), and concepts, objects, relationships and structures, actions or events about a domain (domain requirements). To analyze this indicator, we established the ratio of elicited requirements as Pi=(REi*100)/15.

The case study time frame illustrated in Figure 1 indicates when each technique was applied in the first phase, that is, the execution phase. The second phase included data collection and results analysis.

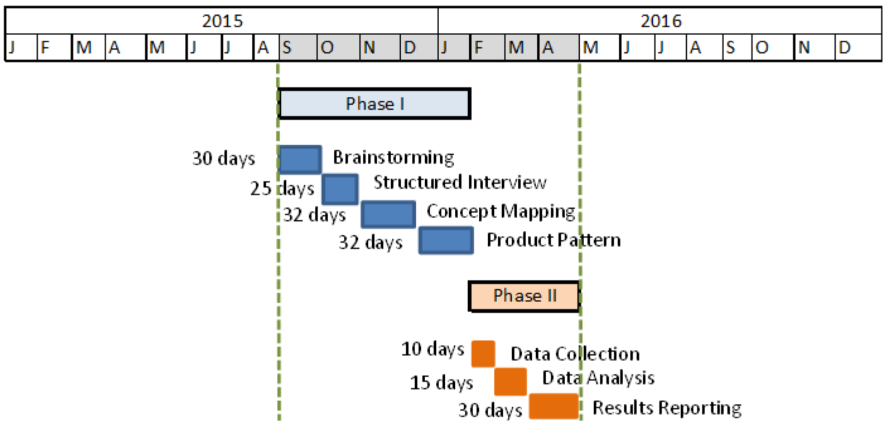


*Figure 1: Gantt chart of the case study phases*

### 4.1 Techniques

The techniques to be examined as candidate techniques useful for externalizing requirements, selected as discussed in Section 3, are: structured interviews, product patterns, brainstorming and concept mapping.

### 4.2 Participants

Experts from the software engineering group (IDIS) at Colombia's Universidad del Cauca participated in this case study. This research group specializes in the search for mechanisms to improve the software industry, effectively influencing the region's and the country's competitiveness. All the participants had the same experience in the software project elicitation phase, totalling from six to eight years. The participants will be referred to throughout the paper as "software engineers".

IDIS was ranked within the top 10 systems engineering and software research groups in Latin America, as classified by a study of Latin America by CEELAM, a European business consulting company [IDIS, 2008].

In order to analyze the techniques application and results, two of the paper authors, who are senior software engineering experts, defined the four projects in which the software engineering techniques to be analyzed were applied, limiting the scope of each project to 15 requirements, they acted as "clients" and later the other two authors analyzed the results of this research, so the fact to act as clients do not affect the analysis of the data because there where different people.

### 4.3 Documentation

The case study participants received general information about the project to be elicited, and user guide for each technique. The participants received no additional documentation concerning knowledge on the domain in which they were working.

Additionally, all documented evidence generated by applying the techniques was collected.

### 4.4 Training

The software engineers participating in the case study did not receive any training related to the problem domain for the four projects that were part of this case study. The training provided at this stage was related to the characteristics, use and application of the selected elicitation techniques in order to establish a single and general knowledge elicitation technique baseline.

The objective of this activity was to acquaint the participants with the four techniques that were to be applied in knowledge externalization to assure that they used and evaluated these techniques to the best of their ability.

To do this, the participants received an email containing information about all four techniques (use characteristics and technique application) and, based on this information, used these specific techniques to externalize knowledge from two Spanish experts acting as clients.

### 4.5 Methodology

Figure 2 is a diagram illustrating steps of the methodology applied to develop the case study.

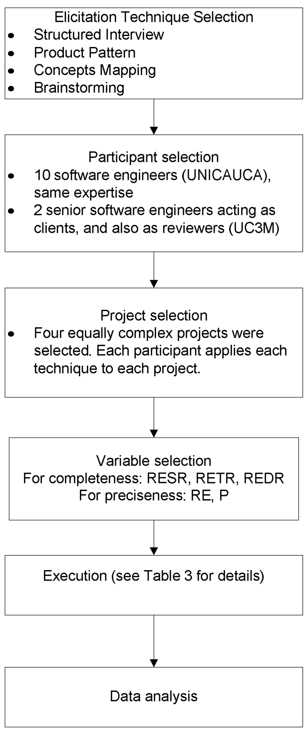


*Figure 2: Case study development*

### 4.6 Data Collection

The indicators used to analyze the performance of the elicitation techniques are: completeness and preciseness. Table 4 in Section 8 Annex 1, presents these indicators as well as the data gathered.

### 4.7 Analysis and Results

An OLAP (OnLine Analytical Processing) technique was used to conduct the analysis [Chaudhuri, 1997][Han, 2011]. This technique can objectively identify relationships among indicators by analyzing the data collected for the case study indicators. About the kiviat charts in next sections, they represent the mean value in the four projects, for each software engineer involved in the case study.

#### 4.7.1 Requirements Completeness

In this section, we present the completeness analysis of each selected technique.

a. Analysis of the results of the structured interview technique

Figure 3 shows a kiviat chart illustrating the extent to which this technique facilitates the elicitation of requirements by type. The values for the application of this technique tend to be higher for the description of task requirements (RETRi).For the domain and strategic requirements levels related to concepts, relationships and structures and strategies, respectively, the structured interview technique has some weaknesses highlighted by the lower levels of the REDRi and RESRi indicators. So, this technique does not identify all the domain and strategic requirements.

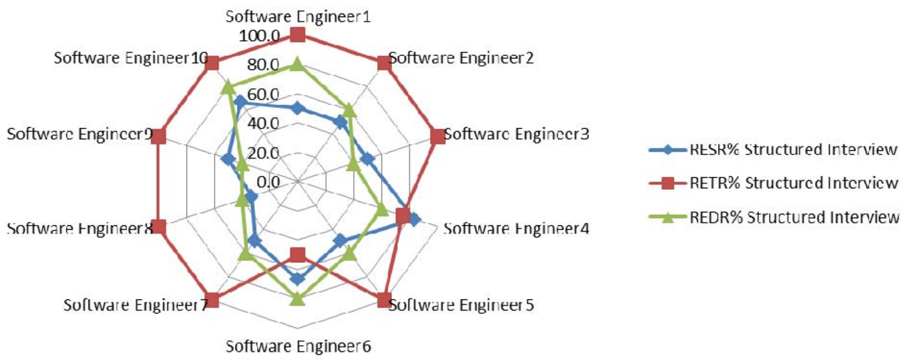


*Figure 3: Coverage of structured interview technique*

b. Analysis of the results of the product pattern technique

Figure 4 shows that there is not a marked difference for any of the three knowledge types. By contrast, we can deduce from the similarity of the results for each knowledge type that the product pattern technique is not dependent on the type of requirements to be externalized and behaves homogeneously for externalizing domain, strategic and task requirements elicitation.

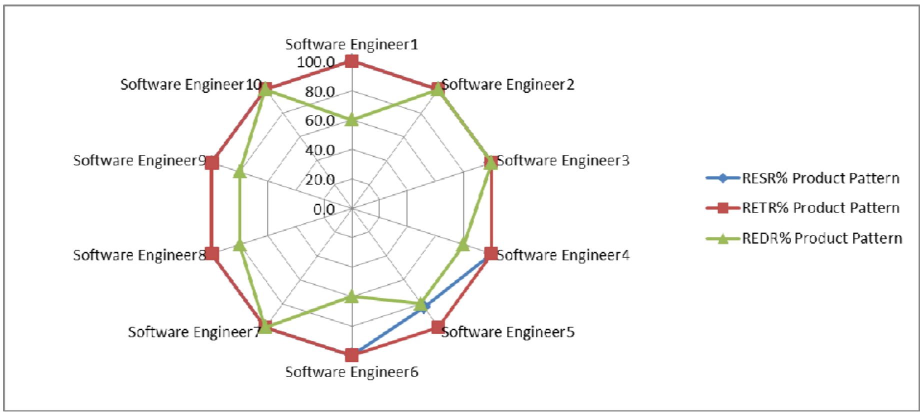


*Figure 4: Product pattern technique coverage*

c. Analysis of the results of the brainstorming technique

The brainstorming technique, Figure 5, presents higher values for the RESRi indicator. With respect to the REDRi indicator (domain knowledge layer related to concepts, structures and their relations within a specific domain) and the RETRi indicator (task knowledge layer related to experience in the performance of procedures, tasks and activities), the values for the brainstorming technique were found to be close to each other but much lower than for the RESRi indicator.

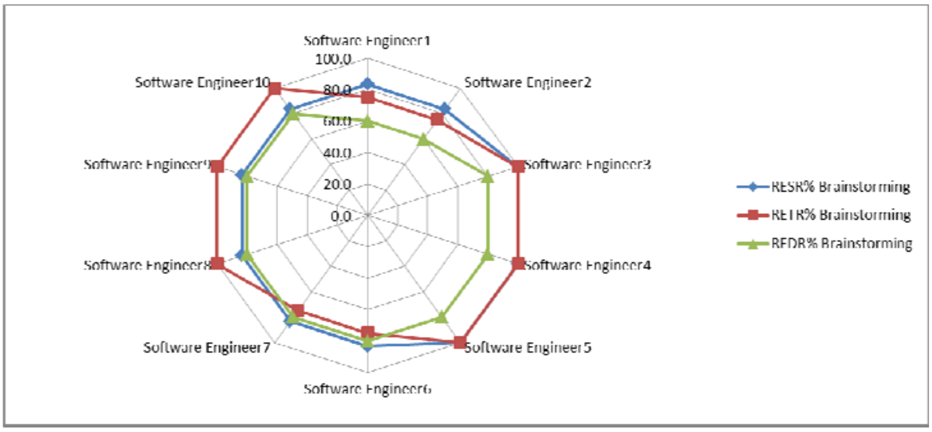


*Figure 5: Brainstorming technique coverage*

d. Analysis of the results of the concept mapping technique

Figure 6 shows that the values for the concept mapping technique are higher for the REDRi indicator. The features of the technique could explain these results, because concept mapping facilitates the identification, clustering and relationship of concept.

With respect to the values of the RESRi (strategic requirements layer related to decision making, task selection and expert knowledge) and RETRi (task requirements layer related to experience in the performance of tasks and activities at the procedural level) indicators, the values of the concept mapping technique were found to be close to each other but much lower than for the REDRi indicator.

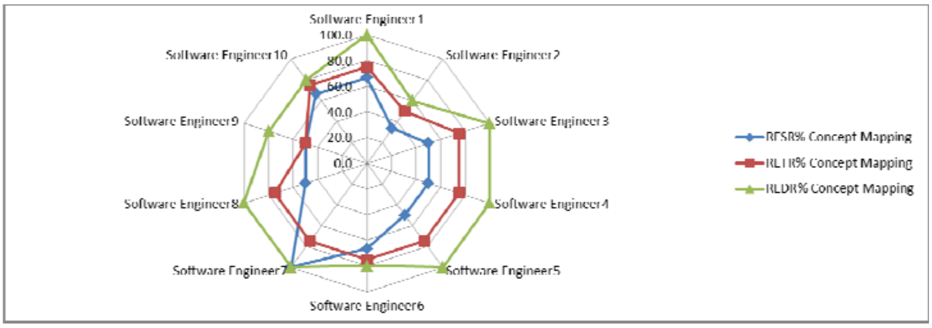


*Figure 6: Concept mapping technique coverage*

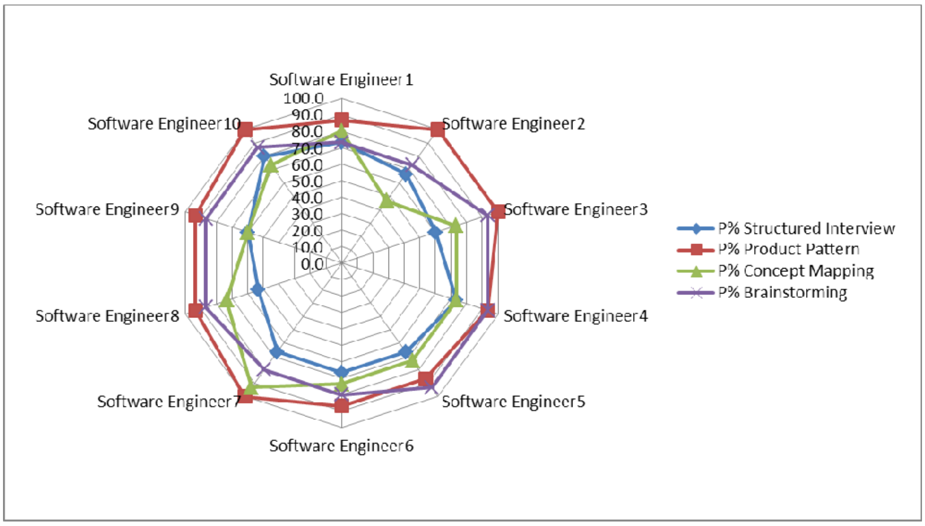


*Figure 7: Elicitation technique preciseness*

### 4.7.2 Preciseness of elicitation techniques

Pi represents the total percentage of requirements externalized using each of the techniques analyzed in this case study. Figure 7 illustrates the result of the preciseness for each technique used in knowledge externalization from the 10 case study participants.

Note in Figure 7 that the Product Pattern technique accounts for a larger area than the other techniques and therefore scores highest for the preciseness indicator with an average number of requirements identified by the 10 participants of 94%, followed by Brainstorming with an average number of identified requirements of 84.7%, by Concept Mapping with an average number of identified requirements of 72% and, finally, by Structured Interview with an average of identified requirements of 66.7%.

### 4.7.3 Summary of analyzed variables for the selected elicitation techniques

Table 3 summarizes the indicators and variables defined to measure the completeness and preciseness of requirements during requirements elicitation.

| Elicitation Technique | Completeness (Average) | | | Rank for REDR | Rank for RESR | Rank for RETR | Preciseness (Average) | | Rank for P |
|---|---|---|---|---|---|---|---|---|---|
| Structured Interview | REDR: 60% | RESR: 55% | RETR: 92.5% | 4 | 4 | 2 | RE: 10 | P: 66.7% | 4 |
| Product Pattern | REDR: 84% | RESR: 98.3% | RETR: 100% | 2 | 1 | 1 | RE: 14.1 | P: 94% | 1 |
| Brainstorming | REDR: 76% | RESR: 83.3% | RETR: 90% | 3 | 2 | 3 | RE: 12.7 | P: 84.7% | 2 |
| Concept Mapping | REDR: 90% | RESR: 58.3% | RETR: 70% | 1 | 3 | 4 | RE: 10.8 | P: 72% | 3 |

*Table 3: Elicitation techniques ranking*

Column 1 lists the name of the elicitation technique. Column 2 shows the epistemological knowledge layers used to calculate the Requirements Completeness indicator, listing, for each technique, the average number of the requirements elicited by all 10 software engineers (as a percentage) for each of the REDR, RETR and RESR variables. Column 3, 4 and 5 specify the rank of each technique for the domain, strategic and task requirements completeness indicator, respectively (the rank position was calculated considering the average for each technique). Column 6 denotes the variables used to calculate the Requirements Preciseness indicator (indicator explained in Section 4). RE indicates the total average number of requirements acquired in the elicitation session using each technique (calculated using the total number of individuals). On the other hand, P stands for the total percentage of requirements elicited by each individual.

Finally, Column 7 presents the ranking of each technique by the Requirements Preciseness indicator (the rank position was assigned by the Pi indicator value).

## 5 Conclusions and future work

The data analysis reveals that not all the elicitation techniques analyzed in this case study perform equally in the process of eliciting requirements from different layers of knowledge. If this has happened with the analyzed techniques, it can be assumed that different elicitation techniques perform differently depending on the type of the requirements —domain, task or strategic— to be elicited. Consequently, if a software engineer selects a technique without taking into account its overall preciseness (number of requirements that it is able to elicit), as well as its completeness (number of identified domain, task and strategic requirements), he or she is likely to miss a considerable number of requirements merely on the grounds of poor elicitation technique selection.

This case study even when it is not aimed to provide formal empirical evidence about the effects of the criteria discussed, provides preliminary information to consider and further discuss such criteria during the identification of elicitation techniques. Based on the findings of this paper, completeness and preciseness should be added to the existing criteria for consideration in the guidelines on the selection of the best requirement elicitation technique for each project.

Focusing on the analyzed techniques, the results of this case study show that the Concept Mapping technique performs is good as eliciting requirements in the domain knowledge layer. This technique could be a justified choice under circumstances where the knowledge to be elicited has to be converted into explicit knowledge using models and concepts.

We have found that the Brainstorming technique is appropriate for working with specialized requirements. Brainstorming helps to acquire and represent strategic level requirements. This technique scores highest for the strategic requirements coverage indicator.

We can conclude that more requirements are elicited using Product Patterns than using any of the other techniques under this study, so this is a good technique to be used when it is not clear the kind of requirements to be elicited.

As regards the Structured Interview technique, the values of the requirements completeness indicator for the task requirements layer were higher than for the domain and strategic requirements indicators. This could be because the technique is better at eliciting requirements about procedures, tasks and/or explanations of events. This is also an appropriate technique when more particulars have to be elicited about specific topics in order to get detailed requirements about a particular domain, as interviews are flexible and can be tailored to the needs of the researcher and the required detail level.

As a future work, this case study could be extended to the remainder of software engineering techniques classified in Table 1 under the socialization, combination and internalization phases of Nonaka and Takeuchi's model for knowledge creation [Nonaka, 1995].

## References

[Beecham, 2005] Beecham, S., Hall, T., and Rainer, A.: 'Defining a requirements process improvement model. Software Quality Journal, 2005, 13(3), 247-279.

[Booch, 2008] Booch, G., Maksimchuk, R. A., Engel, M. W., Young, B. J., Conallen, J., and Houston, K. A.: 'Object-oriented analysis and design with applications', 2008, (3). Addison-Wesley.

[Browne, 2008] Browne, G. J., and Rogich, M. B.: ‘An empirical investigation of user requirements elicitation: Comparing the effectiveness of prompting techniques’. Journal of Management Information Systems, 2001, 17(4), pp.223-250.

[Carrizo, 2014] Carrizo, D., Dieste, O., and Juristo, N.: ‘Systematizing requirements elicitation technique selection’. Information and Software Technology, 2014, 56(6), pp. 644-669. http://doi.org/10.1016/j.infsof.2014.01.009

[Carrizo, 2015] Carrizo, D., Influential contextual attributes in the requirements elicitation process:a comprehensive literature review. Ingeniare. Revista chilena de ingeniería, vol. 23 Nº 2, 2015, pp. 208-218

[Carrizo, 2016] Carrizo, D., Ortiz, C. Aguirre, L. What do researchers mean by “the right requirements elicitation techniques”?. Ingeniare. Revista chilena de ingeniería, vol. 24 Nº 2, 2016, pp. 263-273

[Chaudhuri, 1997] Chaudhuri, S., and Dayal, U.: ‘An overview of data warehousing and OLAP technology’. ACM Sigmod record, 1997, 26(1), pp. 65-74.

[Chung, 2000] Chung, L., Nixon, B., Yu, E., and Mylopoulos, J.: ‘Non-functional requirements’. Software Engineering, 2000.

[Davis, 2002] Davis, A., and Hickey, A.: ‘Requirements Researchers: Do We Practice What We Preach’. Requirements Engineering Journal. 2002

[Gause ,1989]Gause, D., and Weinberg G.: ‘Exploring Requirements: Quality before Design’, 1989, Dorset House Publishing. New York.

[Han, 2011] Han, J., Kamber, M., and Pei, J.: ‘Data mining: concepts and techniques’, 2011. Elsevier.

[Heninger, 1980] Heninger, K.: ‘Specifying Software Requirements for Complex Systems: New Techniques and Their Application’. IEEE Trans on Software Eng, 1980, 6(1), pp. 2-12.

[Hickey, 2003a] Hickey, A., and Davis, A.: ‘A Tale of Two Ontologies: A Basis for Systems Analysis Technique Selection’. Americas Conference of Computer Information Systems (AMCIS), 2003.

[Hickey, 2003b] Hickey, A., and Davis, A.: ’Elicitation technique selection: how do experts do it?’. In Requirements Engineering Conference, 2003.Proceedings. 11th IEEE International, pp. 169-178.

[Hickey, 2003c] Hickey, A., and Davis A.: ‘Requirements Elicitation and Elicitation Technique Selection: A Model for Two Knowledge-Intensive Software Development Processes’. Proceedings of the 36th Hawaii International Conference on System Sciences – HICSS. IEEE Computer Society. 2003. DOI: 10.1109/HICSS.2003.1174229

[Holbrook, 1990] Holbrook H: ‘A Scenario-Based Methodology for Conducting Requirements Elicitation’. ACM SIGSOFT Software Engineering Notes, 1990, pp. 95-104.

[Hsia, 1994] Hsia, P., Samuel, J., Gao, J., Kung, D., Toyoshima, Y., and Chen, C: ‘Formal Approach to Scenario Analysis’. IEEE Software, 1994, 11(2), pp. 33-41

[Hull, 2005] Hull, E., Jackson, K., and Dick, J.: ‘.Requirements engineering’, 2005, (3). London: Springer.

[IDIS, 2008]IDIS: Investigación y desarrollo en Ingeniería del Software (2008). Available at: http://www.unicauca.edu.co/idis/news/2008/6/13/Grupo-IDIS-fue-catalogado-entre-los-10-

mejores-grupos-de-investigacion-en-el-tema-de-Ingenieria-de-Sistemas-y-del-Software-en-Latinoamerica

[Jackson, 2001] Jackson, M.: 'Problem Frames', Addison-Wesley, 2001.

[Kausar, 2010] Kausar, S., Tiriq, S., Riaz, S., and Khanum, A.: 'Guidelines for selection of elicitation techniques'. 6th International conference on Emerging technologies (ICET 2010), 2010, pp. 265-269- http://doi.org/10.1109/ICIT.2010.5638476.

[Lauesen, 2002] Lauesen, S.: 'Software Requirements: Styles and Techniques', Addison-Wesley, 2002

[Lauesen, 2002] Lauesen, S.: 'Software Requirements: Styles and Techniques', 2002, Addison-Wesley.

[Leffingwell, 2000] Leffingwell, D., and Widrig, D.: 'Managing Software Requirements', 2000, Addison-Wesley.

[Maiden, 1996] Maiden, N. A. M., and Rugg, G.: 'ACRE: selecting methods for requirements acquisition'. Software Engineering Journal, 1996, 11(3), 183-192.

[Medina-Dominguez, 2008] Medina-Dominguez, F., Sanchez-Segura, M., Mora-Soto, A., and Amescua A.: 'Patterns in the field of software engineering'. Encyclopedia of Information Science and Technology 2nd.Edition.Idea Group Publishers. 2008.

[Medina-Dominguez, 2009] Medina-Dominguez, F., Sanchez-Segura, M., Mora-Soto, A., and Amescua A.: 'Reverse engineering and software products reuse to teach collaborative web portals: a case study with final-year computer science students'. Journal of IEEE Transaction on Education, 2009

[Mylopoulos, 2001] Mylopoulos, J., Chung, L., Liao, S., Wang, H., and Yu, E.: 'Exploring alternatives during requirements analysis. Software', IEEE, 2001, 18(1), pp.92-96.

[Nonaka, 1995] Nonaka, I., and Takeuchi, H.: 'The Knowledge Creating Company: How Japanese Companies Create the Dynamics of Innovation', 1995, Oxford University Press, New York.

[Nonaka, 2007] Nonaka, I.: 'The Knowledge-creating company'. Harvard Business Review, 2007, pp. 162-171. Doi: 10.1016/S0048-7333(97)80234-X

[Nuseibeh, 2000] Nuseibeh, B., and Easterbrook, S.: 'Requirements engineering: a roadmap'. In Proceedings of the Conference on the Future of Software Engineering, 2000, pp. 35-46. ACM.

[Paetsch, 2013] Paetsch, F., Eberlein, A., and Maurer, F.: 'Requirements engineering and agile software development. In 2012 IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2013, pp. 308-308. IEEE Computer Society.

[Pressman, 2015] Pressman, R: 'Adaptable Software Model, Preparing for Software Requirements Elicitation' http://www.rspa.com/checklists/reqelicit.html, accessed March 2017

[Razali, 2012] Razali, F. A. And R.: 'A practical guide to requirements elicitation techniques Selection – An empirical study'. Middle-East Journal of Science Research, 2012, 11(8), pp. 9.

[Robertson, 1999] Robertson S., and Robertson J.: 'Mastering the Requirements Process', 1999. Addison-Wesley.

[Runeson, 2009] Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. Empirical software engineering, 14(2), 131.

[Sawyer, 1999] Sawyer, P., Sommerville, I., and Kotonya, G. International Conference on Product Focused Software Process Improvement. ed. / M. Oivo; P. Kuvaja. 1999. p. 222-236 (VTT Symposium; Vol. 195). 1999

[Schreiber, 1994] Schreiber, G., Wielinga, B., Hoog, R., Akkermans, H., and Van de Velde, W.: 'CommonKADS: A Comprehensive Methodology for KBS Development', IEEE Expert, 1994, 9(6).

[Tiwari, 2012] Tiwari, S., Rathore, S. S., and Gupta, A.: 'Selecting requirements elicitation techniques for software projects'. CSI 6th International conference on software engineering, (CONSEG), 2012. http://doi.org/10.1109/CONSEG.2012.6349486.

[Van Lamsweerde, 2000] Van Lamsweerde, A.: 'Requirements engineering in the year 00: A research perspective'. In Proceedings of the 22nd international conference on Software engineering, 2000, pp. 5-19.ACM.

[Wiegers, 1999] Wiegers, K.: 'Software Requirements', 1999, Microsoft Press.

[Wielinga, 1992] Wielinga, B.J., Schreiber, A., and Breuker J.A: 'KADS: A Modelling Approach to Knowledge Engineering, Knowledge Acquisition, 1992, 4(1).

[Yin, 2013] Yin, R.K. Case Study Research: Design and Methods, 2013, 5 ed. Sage.

# Annex 1

Table 4 summarizes the collected data. SE = Software Engineer

| | | | | Completeness | | | | | | | | | Preciseness | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | RE_SR | ERE_SR | RESR% | RE-TR | ERE-TR | RETR% | RE_DR | ERE_DR | REDR% | RE | P=(RE*100)/15 |
| 1 Structured Interview | Project 1 | Client1 | SE1 | 3 | 6 | 50,0 | 4 | 4 | 100 | 4 | 5 | 80 | 11 | 73,3 |
| | | | SE2 | 3 | 6 | 50,0 | 4 | 4 | 100 | 3 | 5 | 60 | 10 | 66,7 |
| | | | SE3 | 3 | 6 | 50,0 | 4 | 4 | 100 | 2 | 5 | 40 | 9 | 60,0 |
| | | | SE4 | 5 | 6 | 83,3 | 3 | 4 | 75 | 3 | 5 | 60 | 11 | 73,3 |
| | | | SE5 | 3 | 6 | 50,0 | 4 | 4 | 100 | 3 | 5 | 60 | 10 | 66,7 |
| | | | SE6 | 4 | 6 | 66,7 | 2 | 4 | 50 | 4 | 5 | 80 | 10 | 66,7 |
| | | | SE7 | 3 | 6 | 50,0 | 4 | 4 | 100 | 3 | 5 | 60 | 10 | 66,7 |
| | | | SE8 | 2 | 6 | 33,3 | 4 | 4 | 100 | 2 | 5 | 40 | 8 | 53,3 |
| | | | SE9 | 3 | 6 | 50,0 | 4 | 4 | 100 | 2 | 5 | 40 | 9 | 60,0 |
| | | | SE10 | 4 | 6 | 66,7 | 4 | 4 | 100 | 4 | 5 | 80 | 12 | 80,0 |
| 2 Product Pattern | Project 2 | Client1 | SE1 | 6 | 6 | 100,0 | 4 | 4 | 100 | 3 | 5 | 60 | 13 | 86,7 |
| | | | SE2 | 6 | 6 | 100,0 | 4 | 4 | 100 | 5 | 5 | 100 | 15 | 100,0 |
| | | | SE3 | 6 | 6 | 100,0 | 4 | 4 | 100 | 5 | 5 | 100 | 15 | 100,0 |
| | | | SE4 | 6 | 6 | 100,0 | 4 | 4 | 100 | 4 | 5 | 80 | 14 | 93,3 |
| | | | SE5 | 5 | 6 | 83,3 | 4 | 4 | 100 | 4 | 5 | 80 | 13 | 86,7 |
| | | | SE6 | 6 | 6 | 100,0 | 4 | 4 | 100 | 3 | 5 | 60 | 13 | 86,7 |
| | | | SE7 | 6 | 6 | 100,0 | 4 | 4 | 100 | 5 | 5 | 100 | 15 | 100,0 |
| | | | SE8 | 6 | 6 | 100,0 | 4 | 4 | 100 | 4 | 5 | 80 | 14 | 93,3 |
| | | | SE9 | 6 | 6 | 100,0 | 4 | 4 | 100 | 4 | 5 | 80 | 14 | 93,3 |
| | | | SE10 | 6 | 6 | 100,0 | 4 | 4 | 100 | 5 | 5 | 100 | 15 | 100,0 |
| 3 Concepts Mapping | Project 3 | Client2 | SE1 | 4 | 6 | 66,7 | 3 | 4 | 75 | 5 | 5 | 100 | 12 | 80,0 |
| | | | SE2 | 2 | 6 | 33,3 | 2 | 4 | 50 | 3 | 5 | 60 | 7 | 46,7 |
| | | | SE3 | 3 | 6 | 50,0 | 3 | 4 | 75 | 5 | 5 | 100 | 11 | 73,3 |
| | | | SE4 | 3 | 6 | 50,0 | 3 | 4 | 75 | 5 | 5 | 100 | 11 | 73,3 |
| | | | SE5 | 3 | 6 | 50,0 | 3 | 4 | 75 | 5 | 5 | 100 | 11 | 73,3 |
| | | | SE6 | 4 | 6 | 66,7 | 3 | 4 | 75 | 4 | 5 | 80 | 11 | 73,3 |
| | | | SE7 | 6 | 6 | 100,0 | 3 | 4 | 75 | 5 | 5 | 100 | 14 | 93,3 |
| | | | SE8 | 3 | 6 | 50,0 | 3 | 4 | 75 | 5 | 5 | 100 | 11 | 73,3 |
| | | | SE9 | 3 | 6 | 50,0 | 2 | 4 | 50 | 4 | 5 | 80 | 9 | 60,0 |
| | | | SE10 | 4 | 6 | 66,7 | 3 | 4 | 75 | 4 | 5 | 80 | 11 | 73,3 |
| 4 Brainstorming | Project 4 | Client2 | SE1 | 5 | 6 | 83,3 | 3 | 4 | 75 | 3 | 5 | 60 | 11 | 73,3 |
| | | | SE2 | 5 | 6 | 83,3 | 3 | 4 | 75 | 3 | 5 | 60 | 11 | 73,3 |
| | | | SE3 | 6 | 6 | 100,0 | 4 | 4 | 100 | 4 | 5 | 80 | 14 | 93,3 |
| | | | SE4 | 6 | 6 | 100,0 | 4 | 4 | 100 | 4 | 5 | 80 | 14 | 93,3 |
| | | | SE5 | 6 | 6 | 100,0 | 4 | 4 | 100 | 4 | 5 | 80 | 14 | 93,3 |
| | | | SE6 | 5 | 6 | 83,3 | 3 | 4 | 75 | 4 | 5 | 80 | 12 | 80,0 |
| | | | SE7 | 5 | 6 | 83,3 | 3 | 4 | 75 | 4 | 5 | 80 | 12 | 80,0 |
| | | | SE8 | 5 | 6 | 83,3 | 4 | 4 | 100 | 4 | 5 | 80 | 13 | 86,7 |
| | | | SE9 | 5 | 6 | 83,3 | 4 | 4 | 100 | 4 | 5 | 80 | 13 | 86,7 |
| | | | SE10 | 5 | 6 | 83,3 | 4 | 4 | 100 | 4 | 5 | 80 | 13 | 86,7 |

*Table 4: Data gathered*