Information quality and quantity-based model to represent the appropriateness of software requirements elicitation techniques

Modelo para representar la adecuación de las técnicas de educción de requisitos software basado en cantidad y calidad de la información

Luis Aguirre–Arredondo, Dante Carrizo–Moreno*

Departamento de Ingeniería Informática y Ciencias de la Computación, Universidad de Atacama. Avenida Copayapu 485 – Copiapó. CP 1530000. Copiapó, Chile.

ABSTRACT: To capture information about the needs of stakeholders and the problem domain and to specify the requirements of planned software, developers can use several elicitation techniques from various sciences such as the social sciences or psychology. Each technique has different performance depending on the context in which it is applied. Therefore, to know which requirements elicitation technique is the most appropriate, it is necessary to understand the meaning of appropriate technique. Practitioners and researchers have differing views of the techniques suitability. This paper proposes a model to represent the appropriateness of elicitation techniques. The model uses an estimator calculated through the variables of requirements quantity and quality. To illustrate the proposal, it was validated with data from an experiment found in the related literature. Our work aims to unify an appropriateness construct, which can help standardize future empirical studies and thus facilitate the creation of a body of knowledge on requirements elicitation techniques.

RESUMEN: Para capturar información de las necesidades de los stakeholders y del dominio del problema, y luego conformar los requisitos del software a construir, los desarrolladores tienen a disposición una miríada de técnicas de educción, provenientes de diversas ciencias como las ciencias sociales o la sicología. Por su naturaleza, cada técnica tiene un desempeño diferente dependiendo del contexto en que aplica. Por lo tanto, para saber qué técnica de educción de requisitos es la más adecuada es necesario saber qué deberíamos entender por "técnica más adecuada". Tanto desarrolladores como investigadores, y entre investigadores, tienen visiones diferentes de la bondad de las técnicas. Este trabajo propone un modelo para representar el constructo de adecuación de las técnicas de educción. Este modelo utiliza un estimador de adecuación, el cual se calcula a través de las variables de cantidad y calidad de requisitos. El modelo propuesto fue validado con datos de un experimento encontrado en la literatura. Este modelo pretende unificar el constructo de adecuación, lo que puede contribuir a uniformar futuros estudios empíricos y así, facilitar la conformación de un cuerpo de conocimientos sobre las técnicas de educción de requisitos.

1. Introduction

In software engineering, there is agreement on the set of activities to be carried out at the requirements stage: requirements elicitation, requirements analysis, requirements specification, validation and verification requirements, and requirements management [1].

Elicitation occurs mainly in the initial stages of software development to capture relevant information

for determining software requirements. To do this, requirements engineers can use many techniques, many of them coming from sciences very different from software engineering, such as psychology or linguistics [2]. These techniques vary in performance depending on the context in which the elicitation happens in a software development project. Since requirements elicitation is a critical step for creating a quality software product, it is necessary to select the most appropriate technique for each elicitation session [3, 4].

However, there is no agreement about the meaning of the construct of appropriateness, which in the related literature is generally referred as the effectiveness of an elicitation technique. In a previous study, the researchers identified thirteen metrics by which researchers have

represented this construct in empirical and theoretical studies [5]. Furthermore, another subsequent study found different views between practitioners and researchers on the appropriateness construct [6].

For this reason, this research presents an approach to represent the construct of appropriateness for elicitation techniques. This model used constructs identified in a systematic mapping study as a starting point, which were then reduced to two fundamental variables representing the degree of a technique's suitability: that is, the quantity and quality of the elicited requirements information. The proposed model was validated using an experiment comparing elicitation techniques in two development environments: collocated and distributed.

This research aims to create a simplified and practical way to represent the appropriateness of elicitation techniques and to facilitate the correct selection of one at a given stage of a software development project. In addition, this proposal may help to standardize the response variables of empirical methods when studying the behavior of elicitation techniques.

The structure of this work is as follows: Section 2 gives a history of the problem and approaches to the solution, Section 3 presents the research methodology, Section 4 presents an analysis of the constructs found in the literature, Section 5 presents the construction of the model, Section 6 presents the model validation, Section 7 presents a discussion of results and Section 8 is the conclusion.

2. Background and related work

The software development process consists of five stages [1]: requirements, design, implementation (coding), testing, and maintenance. The requirements phase is one of the most crucial for the success of a software project. Davis [7] reported that if errors are not detected in the requirements phase, it will be 5 to 10 times more expensive to repair errors in the coding phase, and 100 to 200 times more in the maintenance phase. Therefore, correctly carrying out requirements specification is highly desirable to optimize the final cost and ensure the quality of a software product [8].

There are five established activities in the requirements phase: requirements elicitation, requirements analysis, requirements specification, validation and verification of requirements, and requirements management. Requirements Elicitation, also called "requirements capture," "requirements discovery," or "requirements acquisition," among other terms, deals with the origin of stakeholders' needs and how software engineers can

capture them [9]. This activity is a first effort to understand the problem that software must solve. It is fundamentally a human activity in which stakeholders are identified and the relationship between the development team and the client is set.

There are several frameworks that model the elicitation process and its execution. Christel and Kang's model [10] describes the five-step elicitation process in in a waterfall way, with the possibility of returning to previous stages. In another model [11], elicitation is represented by two views: longitudinal and sectional. The longitudinal view shows the elicitation process over time as an iterative activity consisting of a finite set of elicitation sessions. The sectional view of the model gives another perspective, showing the elicitation sessions as chained in a spiral with three moments: preparation, execution, and analysis. Preparation includes deciding on which elicitation techniques to use. For this, there are dozens of techniques available from very different areas such as sociology, linguistics, anthropology, and cognitive psychology [12, 13]. For requirements engineers, then, the problem of how to select the most appropriate requirements elicitation technique at a given time in the software development project appears. This moment of the project is specified by the particular context in which the elicitation process occurs (see Figure 1).
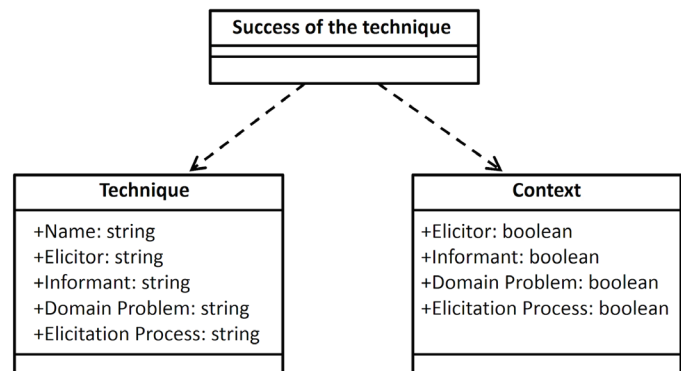

Figure 1 Problem of selecting elicitation techniques

Since elicitation techniques differ, each technique's ability to access information of the problem domain may depend on the contextual situation of the project [14], as shown in Figure 2. However, to know which requirements elicitation techniques are most appropriate, we must first know what we mean by the concept of appropriate. In scientific literature, the appropriateness concept is used in both theoretical and empirical study proposals as a way to compare elicitation techniques. However, there are no previous studies that suggest the need for a unique insight into the construct of performance.
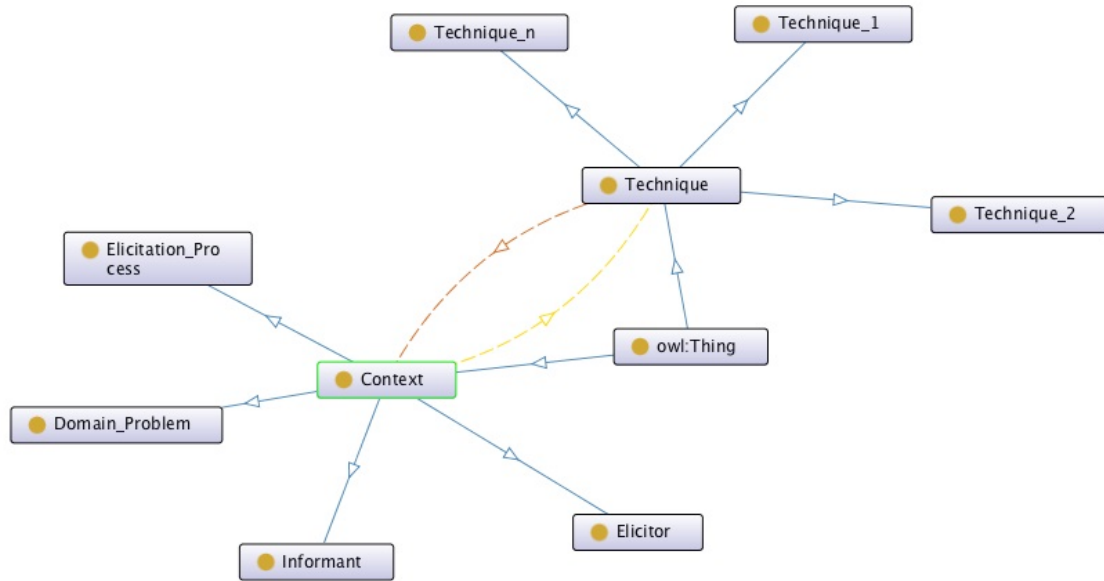
Figure 2 Relationship between elicitation techniques and context factors

Some reviews show the response variables used, but only as secondary information about their work [15]. For this reason, the authors of this paper carried out a mapping study to understand how the goodness of each elicitation technique was displayed. [5] identified 42 primary studies by searching the IEEEXplore, ACM DL, and SCIENCE DIRECT databases. They also conducted opportunistic searches on the Internet using the references of already-identified selected articles and publications.

From the 42 primary works, the researchers extracted 58 metric ways to assess the performance of the techniques.

This means that some works proposed more than one independent metric. These metrics were grouped into 13 constructs that can be seen in Figure 3. Since the constructs "Quality level of understanding" and "Quality of information elicited" belonged to the same concept, they were joined in a single construct called Quality. The same was done with the constructs "Quantity of information elicited" and "Quantity of knowledge elicited," which formed the construct Quantity. In turn, lower use constructs, such as "Productivity," "Utility," "Performance," and "Usability Specification," were categorized as "Others". Thus, there were 6 different constructs.
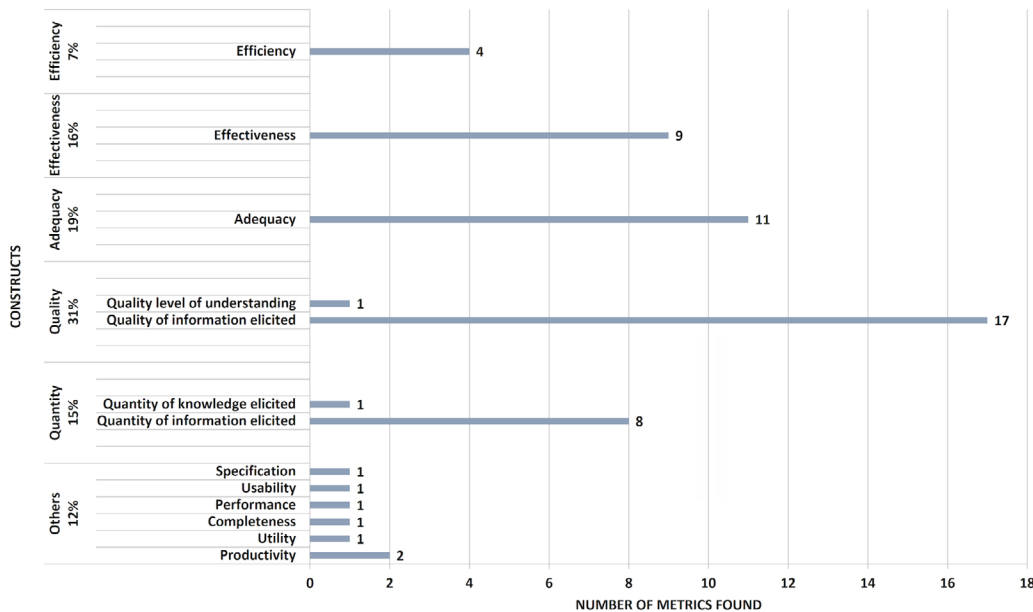

Figure 3 number of metric found by construct

In a later study [6], these representations were subjected to evaluation practitioners through a survey of 14 students in Master's in Information Technology and Innovation. These students were software engineers with experience in low and similar software requirements specification (between 2 and 3 years) who had been instructed in requirements elicitation techniques. The survey consisted of giving values to the constructs representing the success or goodness of the technology. The rating for each of the constructs ranged from 1 (lowest grade) to 5 (higher grade).

Figure 4 shows the researchers and practitioners' views. Comparing the graphics reveals that there is a confused vision of how practitioners evaluate the performance of techniques, since there is no clear trend for how to do it. Moreover, for researchers, appropriateness is determined mainly by quality.
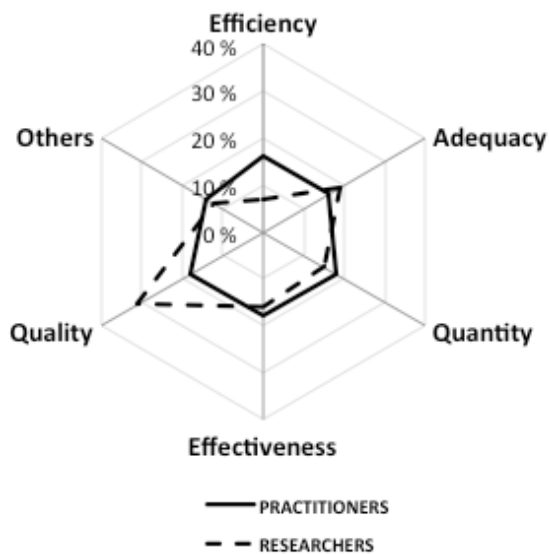

Figure 4 comparison of views of practitioners and researchers

3.   Methodology

As seen above, there is no uniformity regarding how to measure the appropriateness of elicitation techniques.

This evidence is relevant since it has implications for the results of empirical research, especially in efforts to generate a body of knowledge for creating guidelines for requirements engineers. This is particularly important in areas like Evidence-based Software Engineering [16], where the aggregation of evidence is performed by generalizing concepts. The greater the diversity of constructs used the more generalization is required, which means a significant loss of prescriptive information. This then prompts the question: is it possible to generate a model to represent the appropriateness of requirements elicitation techniques?

To answer this question required the analysis and interpretation of data. In addition, this research proposes a formal definition for each construct found. The construction of the model was executed based on empirical work, particularly the data obtained in the analysis. Finally, to validate the proposed model, a paper from the literature containing the data required by the model was used to apply the model. Figure 5 shows the methodology followed for the proposed model.
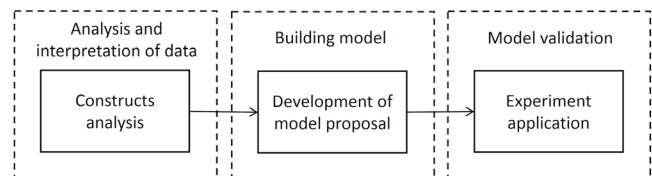
Figure 5 methodology for modeling the appropriateness of elicitation techniques

4.   Analysis of appropriateness constructs

This section presents each of the relevant constructs that have been used by researchers and practitioners to measure the performance of requirements elicitation techniques. These are: quantity, quality, effectiveness, efficiency, and adequacy. A summary of some definitions of authors and their generality for purposes of this research is shown in Table 1. In the context of the requirements elicitation process, these constructs are explained as follows.

Table 1 Definitions of constructs

Construct	Author's definitions	Unified definition
Quantity	Number of objects in the entity, number of construct of higher order and number of literal levels (Chen et al., 2002). Number of use cases identified, Number of dependencies between use cases (Laporti et al., 2009). Number of functional requirements, number of procedural requirements (Moore & Shipman, 2000). Number of functional and non-functional requirements (Sutcliffe, 1997). Number of subgoals{ elicited (Sen & Hemachandran, 2010). Number of objectives, processes, task and level of information (Browne & Rogich, 2001). Number of non-redundant system functions (Duggan & Thachenkary, 2003). Number of interactions between developers and clients in successful projects (Keil & Carmel, 1995). Number of signals (Fowlkes et al., 2000).	Measurement of techniques performance to capture different magnitude of the information
Quality	Quality of the information elicited (complete, coherent and unambiguous requirements) (Al-Salem & Abu Samaha, 2007). Degree in which assessment procedures must be consistent, complete and clear (Hevner & Mills, 1995). Percentage of requirements appropriate to the context (Vieira et al., 2012). Degree in which specifications are accurate, understandable, complete, consistent, unambiguous and relevant (Duggan & Thachenkary, 2003). Quality of Software Requirements Specification (Lloyd et al., 2002). Proportion of intrinsic attributes and extrinsic attributes elicited (Rugg et al., 1992).	Measure of guarantee that the specified requirements correspond to the stakeholders' needs.
Effectiveness	Relationship between the number of real needs identified and the total of the known requirements (Vieira et al., 2012). Qualitative efficacy (Lloyd et al., 2002). Number of criteria identified per transcript page (Agarwal & Tanniru, 1990).	Ability to achieve the effect that is desired or expected, i.e. it is a metric that shows the coverage of requirements captured in an elicitation process
Efficiency	Saving time both in preparation and acquisition (Chen et al., 2002). Time used to obtain the number of use cases (Laporti et al., 2009). Time needed to obtain the resulting requirements (Boulila et al., 2011). Number of unique solutions generated divided by session time (Duggan & Thachenkary, 2003). Number of facts/time and number of rules/time (Chao & Salvendy, 1995).	Optimal use of the resources available to achieve the desired objectives, i.e. the requirements captured in an amount of time.
Adequacy	Stakeholder acceptance, easy application, graphical output, rapid implementation, surface learning curve, high maturity, scalability (Sadiq et al., 2009). Parameter weighting: number of actors involved (X1), level of interaction (X2), quality of requirements collected as consequence of a particular technique (X3), effectiveness of time (X4), profitability (Kausar et al., 2010). Subjective estimation depending on: Application domain, Requirements engineer type, Information resources, User participation, Requirements properties (Tsumaki & Tamai, 2006). Subjective estimation, depending on: Purpose of the requirements, type of knowledge, internal filtration of knowledge, observable phenomena, acquisition context, interdependence method (Maiden & Rugg, 1996). Subjective evaluation depending on the behavior, of each technique, of the problem domain (Wagner et al., 2003). Subjective estimation, dependent on: attributes Size, Complexity, Volatility requirements, Time constraints, Cost constraints (Jiang & Eberlein, 2007).	Assessment based on experience considering certain aspects or characteristics involved in the elicitation process

- Quantity is defined as a portion of a magnitude or a number of units. It represents a clear measurement of a techniques' performance, which allows the comparison of techniques that capture different magnitudes (mathematical properties related to size) of information [17, 18].
- Quality is defined as the property, or set of inherent properties, that allows for judgment of value. Software engineering aims to create products with high quality standards. The quality concept is found across software development, and must therefore be present in requirements engineering to ensure that the requirements specified are consistent with stakeholders' needs. Thus, it is expected that techniques should be aimed at capturing quality information [19, 20].
- Effectiveness is defined as the ability to achieve a desired or expected effect; that is, it is a metric that shows the ability of a process to meet a stated goal. Effectiveness can be measured simply by comparing the results achieved and the results expected (RA/RE). When comparing elicitation techniques' efficiencies, the expected result is the same, so the construct is reduced to a comparison of the quantity of information [21, 22].
- Efficiency is the ability to have someone or something available to achieve a certain effect. In this case, we are looking for the optimal use of available resources to achieve desired objectives. Efficiency can be measured by the results achieved and the resources needed (RA/RN), where the resources are a measure of time or cost. When making a comparison among elicitation techniques' efficiencies, the available resources are the same, so the construct is reduced to a comparison of the quantity of information [23, 24].
- Adequacy is defined as providing, arranging, or appropriating something else. In our case, adequacy is adapting something because preexisting conditions have changed. Therefore, it is difficult to standardize as a criterion for adaptation. This construct represents the output or outcome we wanted to establish in this investigation. In short, we wanted to find the baseline for formalizing the adequacy construct [25, 26].

From the above, it can be concluded that the solution to the problem of how to measure the success or goodness of a requirements elicitation technique depends on the quantity and quality of the information captured. As can be seen in Figure 6, the constructs efficiency and effectiveness can be explained and reduced to Quantity construct. This, since for the purposes of comparing elicitation techniques, the totality of the requirements and the availability of resources (time) is common for the techniques involved. The Quality of the requirements is another necessary construct in the ontology to approximate the measure that represents the performance of the techniques. Finally, both quantity and quality represent ontologically the subjective concept of adequacy.
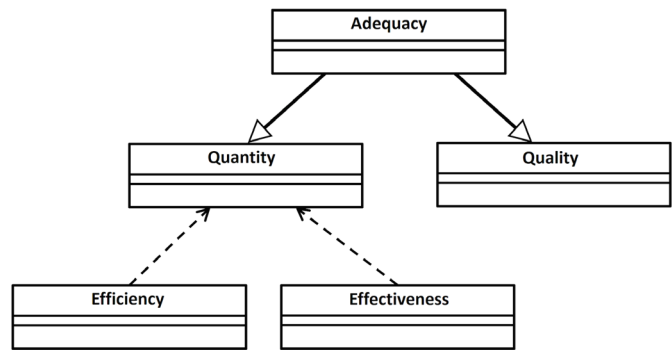

Figure 6 Constructs reduction

In this case, we refer to the quantity of requirements obtained, and not to the amount of information captured by an elicitation technique. This is because an elicitation technique can elicit much information, but not all data collected is useful, since, for example, in an interview the informant can talk about issues that are not useful for building the software system. As the informant can give superfluous information, it is difficult to measure the total possible information.

Thus, it seems more accurate to say that an elicitation technique can capture a certain number of requirements that are a subset of a finite number of requirements. For example, if an experiment is run, you can have a gold standard comprising all requirements to be specified. However, when we talk about quality, we are talking about requirements that meet desirable attributes related to completeness, accuracy, vagueness, and ambiguity, among others.

5. Building the model

Due to the difference in the nature of elicitation techniques, it is possible to expect their performance to be better in some situations than others; that is, certain factors of the project context influence the behavior of elicitation techniques and, therefore, the outcome of the process. Thus, it is necessary to represent the degree to which a particular technique is suitable for the application in a given context. In this paper, we propose using an appropriateness estimator, represented by A_{T_i} (Appropriateness of the technique i). This measurement will take a value between 0 and 100, which represents the percentage of appropriateness of the technique, as shown in equation (1).

$$0 \leq A_{T_i} \leq 100 \qquad (1)$$

In the previous section, it was stated that the criteria representing the success or goodness of the technique are mainly quantity (Q) and quality (K) constructs. Therefore, A_{T_i} will depend on these variables. These variables will also be measured in percentages, and so, Q and K have a maximum value of 100.

Having cleared which variables are concerned with the appropriateness estimator, we proceeded to perform an empirical analysis, where four options were assessed to relate variables:

- Addition: if quantity takes a value of 100, and quality takes a value of 100, the sum gives a result of 200. This being greater than 100, the value is outside the range of evaluation. Although intuitively the addition seems easier for comparing the appropriateness of different techniques, it is not good at all because the value of the estimator must vary between 0 and 100.

- Subtraction: if we calculate the appropriateness estimator by subtraction, techniques with equal values for quantity and quality would always yield the minimum value (0), which is totally incorrect.

- Division: regardless of which concept is the numerator or denominator, the model should allow the comparison of different techniques. For example, if a technique captures 100% of the requirements with 50% quality adjustment, it should agree with another technique that captures 50% of the requirements but with 100% quality. Division operation does not guarantee this standard.

- Multiplication: the commutative property of multiplication can overcome the disadvantage of division. Similarly, the multiplication of quality and quantity can be normalized to a percentage, maintaining the appropriateness estimator in the defined range. That is, by multiplying the maximum values that Quantity and Quality can take, it gives us 10,000. Therefore, if we divide 10,000 by 100, we get the maximum value that the appropriateness estimator can take (100). On the other hand, if the quantity or quality value takes the minimum value (0), the estimator is 0, which sounds logical, because if the value of the quantity is 0, it means that the technique obtained 0 requirements. So, the value of the estimator should be 0. In addition, where quality is 0, we can say that none of the requirements obtained has quality, strictly speaking, since having an incomplete or ambiguous requirement does not work, because the value of the estimator ought to be 0.

Therefore, the best way to represent A_{T_i}, depending on quantity and quality, would be as shown in equation (2).

$$\frac{Quantity_{T_i} * Quality_{T_i}}{100} \qquad (2)$$

Resulting equation (3).

$$A_{T_i}\left(Q_{T_i}, K_{T_i}\right) = \frac{Q_{T_i} * K_{T_i}}{100} \qquad (3)$$

First, this proposal is based on the grounds that the quantity of requirements obtained is a percentage of the total requirements. For example, an elicitation technique can collect 80% of the total requirements of the project to develop.

On the other hand, the measurement of the quality of requirements is only related to the quantity of the requirements obtained, because it is not possible to assess the quality of a requirement that has not been captured. Note that the quality of requirements is considered as a percentage of the average quality of each requirement. Continuing the above example, where 80% of the requirements are captured, if a 50% quality of requirements is obtained, it would be about 50% of the 80% of the requirements captured.

Applying the function $A_{T_i}(Q_{T_i}, K_{T_i})$ with data from the previous example, we would have $A_{T_i}(Q_{T_i}, K_{T_i})$ is equal to 40%. The development of this operation is shown in equation (4).

$$A_{T_i}\left(Q_{T_i}, K_{T_i}\right) = \frac{Q_{T_i} * K_i}{100} = 40\% \qquad (4)$$

where Q = 80%; K = 50%

Moreover, the quantity and quality of a technique depend on aspects of the environment of the requirements elicitation process; i.e., changes in these variables depends on the contextual aspects of the process.

We used four factors to determine the contextual issues that influence the effectiveness of elicitation techniques:

- Elicitor: The person who makes the elicitation. The literature sometimes uses other names, such as analyst or requirements engineer, to refer to this role.

- Informant: The person who has the relevant information. Informants can be clients, users, and any person with an interest in software development: more generally called stakeholder.

- Problem Domain: Knowledge area hosting the problem.

- Elicitation Process: Activities and environment in which the process is done.

These factors were derived through a systematic and nonsystematic review of the scientific literature conducted by Carrizo, Dieste and Juristo [14]. In this proposal, for each of these factors, a Favorable or Unfavorable value is assigned, depending on the context. The context consists of a set of attributes representing desired and undesired characteristics for successful elicitation. However, it is not within the scope of this paper to present the method of determining the Favorable or Unfavorable values from these attribute sets.

For example, for the Elicitor factor, if a requirements engineer is very experienced and well trained in elicitation techniques, a favorable value is assigned. However, if the requirements engineer ignores techniques elicitation, an unfavorable value is assigned.

Contextual factors are represented as follows:

f_e = Rating of Elicitor factor.

f_i = Rating of Informant factor.

f_{dp} = Rating of Domain Problem factor.

f_{pe} = Rating of Elicitation Process factor.

The contextual situation is represented by $C(f_e, f_i, f_{dp}, f_{pe})$, which has sixteen potential configurations: C(Favorable, Favorable, Unfavorable, Favorable), or C(Unfavorable, Favorable, Favorable, Unfavorable).

Therefore, each requirements elicitation technique has an appropriateness measured by the quantity and quality associated with each of the sixteen different contextual situations. The scope of this investigation does not include calculating values for quantity and quality of all the techniques in all configurations. This is because calculating the quantity and quality associated with each technique and context requires major empirical work. This assessment may lead to much more extensive research to be carried out in future work. However, this study used a literature search to find empirical studies that show how we can create this body of knowledge.

From the above, we can deduce that for each elicitation technique, settings of $C(f_e, f_i, f_{dp}, f_{pe})$ will exist whose appropriateness value is maximized, as denoted in equation (5).

$$\forall\, T_i \exists\, C_j f / max\left\{ A_{T_i}\left(Q_{T_i}, K_{T_i}\right)\right\} \tag{5}$$

To calculate the function $C(f_e, f_i, f_{dp}, f_{pe})$, it is important to know how appropriate a technique is in a certain context. Thus, we have equation (6) and equation (7).

$$Q_{T_i}\left(C_j\right) = Q_{T_i}C\left(f_e, f_i, f_{dp}, f_{pe}\right) \tag{6}$$

$$K_{T_i}\left(C_j\right) = K_{T_i}C\left(f_e, f_i, f_{dp}, f_{pe}\right) \tag{7}$$

Substituting into equation (3), we get equation (8)

$$\frac{Q_{T_i}\left(C_j\right)*K_{T_i}\left(C_j\right)}{100} = \frac{Q_{T_i}\left(f_e, f_i, f_{dp}, f_{pe}\right)*K_{T_i}\left(f_e, f_i, f_{dp}, f_{pe}\right)}{100} \tag{8}$$

So, finally, the proposed model to represent the appropriateness of requirements elicitation techniques is shown in equation (9):

$$
\begin{aligned}
& 0 \le Q \le 100\ \wedge\ 0 \le K \le 100 \\
& ; K \exists\ if\ Q \exists \\
& C_j = C(f_e, f_i, f_{dp}, f_{pe})
\end{aligned}
\tag{9}
$$

A_{T_i} : Appropriateness of technique i.

Q_{T_i} : Quantity of Requirements obtained from technique i.

K_{T_i} : Quality of Requirements obtained from technique i.

C_j : Context.

$(f_e, f_i, f_{dp}, f_{pe})$: Contextual factors.

$$A_{T_i}\left(Q_{T_i}(C_j), K_{T_i}(C_j)\right) = \frac{Q_{T_i}(C_j)*K_{T_i}(C_j))}{100}$$

6.   Validating the model

To validate the proposed model, we searched the literature for experimental studies that present data about the quantity and quality of requirements. However, no publication was found with that data. Because of this, a new literature search was conducted. This time, the identification of studies was conducted by searching the SCOPUS database and congresses Workshop em Engenharia Requirements (WER), Experimental Software Engineering Latin American Workshop (ESELAW), and International Workshop on Empirical Requirements Engineering (EMPIRE). Opportunistic Internet searches were also conducted.

These searches identified a single work [27]. This study was a controlled experiment with two factors: elicitation context (distributed/collocated), and elicitation techniques used (3 different combinations of techniques). Two similar experimental phases were run: the first was applied to the context of distributed software development, while the second was applied in a collocated traditional context. For this, eleven groups were randomly conformed to applied techniques in the distributed environment, and nine groups were applied to the collocated environment. Each group consisted of two students (in the role of requirements engineers) and one professor (in the role of a stakeholder).

Note that this experiment assumes that the software development company or organization is small, the software being developed is of small/medium size, the application domain to which the software belongs is information systems administration, and standard communication software tools are available (IP videoconferencing, email, chat, and forums). In addition, the elicitation techniques used in this experiment are those most used in small software companies: interview, questionnaire, and brainstorming. Understanding that in real situations, combinations of techniques are used in the elicitation process, three alternative (factors) combining techniques were defined in the experiment: Interview, Interview + Questionnaire, and Interview + Brainstorming.

The results of the experimental phase executed in a distributed environment can be seen in Table 2, while the results of the experimental phase performed in a collocated environment can be seen in Table 3.

Table 2 Results in distributed environment, generated from data of primary study

Group	Techniques	Percentage of quantity	Percentage of quality
1	Interview + Brainstorming	89.47 %	95.26 %
2	Interview + Questionnaire	94.74 %	87.11 %
3	Interview	81.48 %	92.96 %
4	Interview + Questionnaire	80.00 %	81.00 %
5	Interview + Questionnaire	65.38 %	89.23 %
6	Interview	93.94 %	98.18 %
7	Interview	55.56 %	84.81 %
8	Interview + Questionnaire	92.59 %	99.07 %
9	Interview + Brainstorming	80.00 %	92.25 %
10	Interview + Questionnaire	90.00 %	91.50 %
11	Interview	86.96 %	86.30 %

Table 3 Results in collocated environment, generated from data of primary study

Group	Techniques	Percentage of quantity	Percentage of quality
1	Interview + Questionnaire	85.71 %	91.25 %
2	Interview + Brainstorming	86.36 %	87.73 %
3	Interview	84.38 %	88.28 %
4	Interview	95.65 %	95.00 %
5	Interview + Brainstorming	93.33 %	91.67 %
6	Interview + Questionnaire	100.00 %	90.24 %
7	Interview	100.00 %	98.48 %
8	Interview + Questionnaire	95.56 %	85.33 %
9	Interview + Brainstorming	93.75 %	93.85

6.1. Applying the model in the distributed environment

Determination of contextual factors' values

The Values for contextual factors in a distributed environment are:

f_e = assessment of Elicitor factor is unfavorable, since although the engineer was instructed in requirements elicitation techniques, they did not have enough experience to apply them.

f_i = assessment of Informant factor is unfavorable, since stakeholders did not know the problem domain at all.

f_{dp} = assessment of the problem domain factor is favorable, because the software to be developed is an administrative information system, a relatively known system that addresses a more familiar type of information.

f_{pe} = assessment of Elicitation Process factor is unfavorable, as it happens in a distributed environment in which it is generally more difficult to carry out elicitation.

Thus, the function C is: $C(f_e, f_i, f_{dp}, f_{pe}) = C$ (*Unfavorable, Unfavorable, Favorable, Unfavorable*).

Calculating appropriateness estimator

Table 4 summarizes the results obtained from the appropriateness estimator for each technique in the context C(*Unfavorable, Unfavorable, Favorable, Unfavorable*). The values of Q and K are the average data obtained from Table 2.

Table 4 Results of appropriateness estimator

Techniques	Value of Q	Value of K	Appropriateness estimator
Interview	79.49%	90.56%	71.99%
Interview + Questionnaire	89.33%	89.67%	80.10%
Interview + Brainstorming	78.28%	92.25%	72.21%

Comparing the appropriateness of the Interview, Interview + Questionnaire, and Interview + Brainstorming techniques, we discovered that the Interview + Questionnaire technique has a better performance for the context C(*Unfavorable, Unfavorable, Favorable, Unfavorable*).

6.2. Applying model to collocated environment

Determination of contextual factors values

The Values for contextual factors in a collocated environment are:

f_e = assessment of Elicitor factor will be unfavorable, since although the engineer was instructed in requirements elicitation techniques, they did not have enough experience to apply them.

f_i = assessment of Informant factor will be unfavorable, since stakeholders did not knew the problem domain at all.

f_{dp} = assessment of the problem domain factor will be favorable, because the software to develop is an information system of administration, which is a relatively known system and addresses a more familiar type of information.

f_{pe} = assessment of Elicitation Process factor is favorable, as it happens in a collocated environment, which is generally more traditional and familiar for carrying out the elicitation.

Thus, the function C will be: $C(f_e, f_i, f_{dp}, f_{pe}) = C$ (*Unfavorable, Unfavorable, Favorable, Favorable*).

Calculating appropriateness estimator

Table 5 summarizes the results obtained by from the appropriateness estimator for each technique in the context C(*Unfavorable, Unfavorable, Favorable, Favorable*). The values of Q and K are the average data obtained from Table 3.

Table 5 Results of appropriateness estimator

Techniques	Value of Q	Value of K	Appropriateness estimator
Interview	93.34%	93.92%	87.66%
Interview + Questionnaire	93.76%	88.94%	83.39%
Interview + Brainstorming	91.15%	91.08%	83.02%

Comparing the appropriateness of the Interview, Interview + Questionnaire, and Interview + Brainstorming techniques, we discovered that the Interview technique has better performance for context C(Unfavorable, Unfavorable, Favorable, Favorable).

6.3. Comparing performance of techniques depending on context

We can also compare how the performance of requirements elicitation techniques depends on the context. Table 6 shows that all techniques perform better in context C(Unfavorable, Unfavorable, Favorable, Favorable) than in context C(Unfavorable, Unfavorable, Favorable, Unfavorable).

Table 6 Comparison of techniques performance depending on context

Techniques	Appropriateness estimator	
	C(Unfavorable, Unfavorable, Favorable, Unfavorable)	C(Unfavorable, Unfavorable, Favorable, Favorable)
Interview	71.99%	87.66%
Interview + Questionnaire	80.10%	83.39%
Interview + Brainstorming	72.21%	83.02%

7. Discussion

Having a rule of thumb that allows you to choose the right elicitation technique at any given moment in a project is desirable for practitioners, especially novices. This is mainly because the context of an elicitation process can vary from one project to another, and even from one elicitation session to another within the same project. This variation of context implies that techniques can differ in performance and therefore, for each technique there is a context in which it develops better. This situation was demonstrated by the validation of the model, which showed that each technique or technique pair has a different suitability. However, it should be noted that, for the performance ratio of the techniques to be comparable, the same quantity and quality calculation metric had to be used.

In this case, for Quantity of requirements we used the metric Percentage of evolved requirements, which measured the percentage of requirements in the Software Requirements Document that were identified as an evolution of a basic software requirement. These are the requirements that needed deeper and more thorough elicitation, and are thus evidence of richer interaction between the requirements engineer and the stakeholder. For the Quality of the requirements, we used the metric Percentage of requirements without defect, which measures the percentage of requirements that do not have defects of precision, vagueness, ambiguity, etc. That is, these are defects attributable to deficiencies in the elicitation process.

There exist different ways of measuring each construct of quantity and quality. The proposed model provides flexibility in this matter, so that each development team can define its own metrics and can conform to its own manual on elicitation techniques. What is important is that the same metric is used in such a way that the relative difference in performance of each technique is kept uniform.

The case used in the validation allowed us to obtain measurements of relative performance between the techniques considered in the experiment. In no way does this mean that the Interview and Questionnaire techniques perform best in a distributed environment or that the Interview technique performs best in a collocated

environment. This is because the experiment configured only one of the possible contextual situations for each case. In addition, there are dozens of other techniques that could perform better in those contextual situations.

8.   Conclusions

Due to the nature of requirements elicitation techniques, each has different behavior depending on the context in which it is applied. In the literature, we found a significant amount of work on the use of requirements elicitation techniques; however, there is no research about establishing a unique way to measure performance or to show the great diversity that exists. In addition, a survey conducted of professional software developers revealed no homogeneity in the measurement of performance of requirements elicitation techniques.

This fuzzy vision influences the results of empirical research, mainly in efforts to generate a body of knowledge to help create guidelines for software development professionals, where evidence aggregation is done through the generalization of concepts. The greater the variety of constructs used, the more generalization is required, which means a significant loss of prescriptive information.

To overcome this problem, this paper proposed a model to represent the appropriateness of requirements elicitation techniques. This model uses an estimator, which is calculated through the variables quantity of requirements and quality of requirements. The model was validated with an empirical study from the literature. Unfortunately, we found only one paper with applicable empirical data. This means that there is a lack of rigor in the communication and scientific diffusion of these investigations. Certainly, this is a limitation of the research.

The results of this validation showed that the model of this research allows us to compare which requirements elicitation technique performs better in a given context and how various elicitation techniques work in different contexts, although more validation studies and model evaluations of more elicitation techniques are necessary.

In short, this model can be used by software development professionals to study one elicitation technique in the context in which it is applied or to compare various elicitation techniques in the same context. In addition, this study offers researchers a model to standardize the dependent variable (response variable), which can be used in future empirical studies on the effectiveness of requirements elicitation techniques.

In the future, since there are no useful empirical works, we must carry out empirical studies to perform a sensitivity analysis of the model and generate a repository with appropriateness records for elicitation techniques in different potential contexts. Furthermore, this information can lead to a tool that supports practitioners in making decisions regarding which technique to choose in each case.

9.   References

1. P. Bourque and R. E. Fairley, *Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0*, 3rd ed. CA, USA: IEEE Computer Society Press, 2014.

2. S. Robertson, "Requirements trawling: techniques for discovering requirements," I*nternational Journal of Human-Computer Studies*, vol. 55, no. 4, pp. 405-421, 2001.

3. A. M. Hickey and A. M. Davis, "Requirements elicitation and elicitation technique selection: model for two knowledge-intensive software development processes," in *Proceedings of the 36th IEEE Annual Hawaii International Conference on System Sciences,* HI, USA, 2003, pp. 96-105.

4. N. R. Darwish, A. A. Mohamed and A. S. Abdelghany, "A Hybrid Machine Learning Model for Selecting Suitable Requirements Elicitation Techniques," *International Journal of Computer Science and Information Security*, vol. 14, no. 6, pp. 1-12, 2016.

5. D. Carrizo, C. Ortiz and L. Aguirre, "What do researchers mean by "the right requirements elicitation techniques"?," *Ingeniare, Revista chilena de ingeniería*, vol. 24, no. 2, pp. 263-273, 2016.

6. D. Carrizo, "Comparison of Research and Practice Regarding What We Mean by "The Right Software Requirements Elicitation Technique"," in *2016 10th International Conference on the Quality of Information and Communications Technology (QUATIC),* Lisbon, Portugal, 2016, pp. 79-82.

7. A. M. Davis, *Software requirements: objects, functions, and states, Revised Edition.* 2nd ed., New Jersey, USA: Prentice-Hall, Inc., 1993.

8. A. Hickey and A. Davis, "The role of requirements elicitation techniques in achieving software quality," in *International Workshop on Requirements Engineering: Foundations for Software Quality (REFSQ),* 2002.

9. K. Pohl, *Requirements Engineering: Fundamentals, Principles, and Techniques.* 1st ed., Springer Publishing Company, Incorporated, New York, USA. 2010.

10. M. G. Christel and K. C. Kang, "Issues in requirements elicitation," Software Engineering Inst., Pittsburgh, USA, Tech. Rep. No. CMU/SEI-92-TR-12, Sept. 1992.

11. D. Carrizo, "Contextual dynamic of the software requirements elicitation," *Revista Facultad de Ingeniería Universidad de Antioquia,* no. 69, pp. 34-52, 2013.

12. B. Nuseibeh and S. Easterbrook, "Requirements engineering: a roadmap," in *Proceedings of the Conference on the Future of Software Engineering,* New York, USA, 2000, pp. 35-46.

13. C. Potts, "Seven (plus or minus two) challenges for requirements research," in *Proceedings of the 6th international workshop on Software specification and design,* Como, Italy, 1991, pp. 256-259.

14. D. Carrizo, O. Dieste and N. Juristo, "Systematizing requirements elicitation technique selection," Information and Software Technology, vol. 56, no. 6, pp. 644-669, 2014.

15. O. Dieste and N. Juristo, "Systematic review and aggregation of empirical studies on elicitation techniques," *IEEE Transactions on Software Engineering*, vol. 37, no. 2, pp. 283-304, 2011.

16. B. A. Kitchenham, T. Dyba and M. Jorgensen, "Evidence-Based Software Engineering," in *Proceedings of the 26th International Conference on Software Engineering*, Washington D.C., USA, 2004, pp. 273-281.

17. G. J. Browne and M. B. Rogich, "An Empirical Investigation of User Requirements Elicitation: Comparing the Effectiveness of Prompting Techniques," *Journal of Management Information Systems*, vol. 17, no. 4, pp. 223-249, 2001.

18. A. M. Sen and K. Hemachandran, "Elicitation of Goals in Requirements Engineering Using Agile Methods," in *34th Annual IEEE Computer Software and Applications Conference Workshops (COMPSACW),* Seoul, South Korea, 2010, pp. 263-268.

19. S. R. Costa, D. Viana, R. do Nascimento and T. Conte, "Using Empirical Studies to evaluate the REMO Requirement Elicitation Technique," in *SEKE 24th International Conference on Software Engineering and Knowledge Engineering,* San Francisco, USA, 2012, pp. 33-38.

20. A. P. Massey and W. A. Wallace, "Focus groups as a knowledge elicitation technique: an exploratory study," *IEEE Transactions on Knowledge and Data Engineering,* vol. 3, no. 2, pp. 193-200, 1991.

21. V. Laporti, M. R. Borges and V. Braganholo, "Athena: A collaborative approach to requirements elicitation," *Computers in Industry*, vol. 60, no. 6, pp. 367-380, 2009.

22. N. Boulila, A. Hoffmann and A. Herrmann, "Using Storytelling to record requirements: Elements for an effective requirements elicitation approach," in *Fourth International Workshop on Multimedia and Enjoyable Requirements Engineering-Beyond Mere Descriptions and with More Fun and Games (MERE),* Trento, Italy, 2011, pp. 9-16.

23. C. J. Chao and G. Salvendy, "Impact of cognitive abilities of experts on the effectiveness of elicited knowledge," *Behaviour & Information Technology,* vol. 14, no. 3, pp.174-182, 1995.

24. D. M. Vásquez, M. I. Sánchez, F. Medina, and A. Amescua, "Guideline to Select Knowledge Elicitation Techniques," in *4th World Summit on Knowledge Society,* Mykonos, Greece, 2011, pp. 374-384.

25. L. Jiang and A. Eberlein, "Selecting Requirements Engineering Techniques Based on Project Attributes--A Case Study," in *14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems ECBS'07,* Tucson, AZ, USA, 2007, pp. 269-278.

26. Z. Zhang, "Effective Requirements Development-A Comparison of Requirements Elicitation techniques," in Software Quality Management XV: *Software Quality in the Knowledge Society,* E. Berki, J. Nummenmaa, I. Sunley, M. Ross and G. Staples (Ed.) British Computer Society, 2007, pp. 225-240.

27. S. Zapata, E. Torres, G. Sevilla, L. Aballay and M. Reus, "Effectiveness of traditional software requirement elicitation techniques applied in distributed software development scenarios," in *XXXVIII Conferencia Latinoamericana en Informática (CLEI),* Medellín, Colombia, 2012, pp. 1-7.