A Framework for Teaching Distributed Requirements Engineering in Latin American Universities

G. Sevilla, S. Zapata, F. Giraldo, E. Torres, C. Collazos

***Abstract*—**This work describes a framework for teaching of global software engineering (GSE) in university undergraduate programs. This framework proposes a method of teaching that incorporates adequate techniques of software requirements elicitation and validated tools of communication, critical aspects to global software development scenarios. The use of proposed framework allows teachers to simulate small software development companies formed by Latin American students, which build information systems. Students from three Latin American universities played the roles of engineers by applying an iterative development of a requirements specification in a global software project. The proposed framework involves the use of a specific purpose Wiki for asynchronous communication between the participants of the process. It is also a practice to improve the quality of software requirements that are formulated by the students. The additional motivation of students to participate in these practices, in conjunction with peers from other countries, is a significant additional factor that positively contributes to the learning process. The framework promotes skills for communication, negotiation, and other complementary competencies that are useful for working on GSE scenarios.

***Keywords*—**Requirements analysis, distributed requirements engineering, practical experiences, collaborative support.

I. Introduction

THE process of software development tends, with increasing emphasis, to be distributed globally where the protagonists of this process (stakeholders) are dispersed geographically. This modern form of work is called Global Software Development (GSD). There are a number of reasons for this scenario: cost reduction, better global use of the scarce availability of human resources, more hours of work available in relation to the different time zones, incentives to invest in emerging markets and a significant growth in software demand around the world [16]. It is acceptable to maintain that these new working contexts will involve adaptations in the forms and contents of the teaching of Software Engineering, seeking to prepare professionals for a globalized labor context.

It is known that the success of a software project depends on a large extent on a correct execution of the Software Requirements Engineering (SRE) process. Having a correct definition of the software requirements is key to obtaining a quality product that meets the needs and expectations of the customer/user. The need to have an SRE process within the software engineering process is imperative to obtain quality products [12]. SRE is fundamentally a communication process between a requirements specialist and the customer's stakeholder [1].

Gustavo Sevilla, Sergio Zapata, and Estela Torres are with the National University of San Juan, Argentina (e-mail: gsevilla@iinfo.unsj.edu.ar, szapata@iinfo.unsj.edu.ar, etorres@iinfo.unsj.edu.ar).

Faber Giraldo is with the University of Quindío, Armenia, Colombia (e-mail: fdgiraldo@uniquindio.edu.co).

Cesar Collazos is with the University of Cauca, Popayan, Colombia (e-mail: ccollazo@unicauca.edu.co).

The requirement elicitation stage is the most critical of all the phases in software development, since the mistakes made at this stage are more expensive and difficult to resolve owing to their impact upon the other stages [13], [23]. Likewise, the quality of the requirements improves to the extent that there is greater stakeholder participation. This participation gets difficult when the SRE process is carried out in a distributed environment of development, that is to say, when the actors are geographically dispersed. According to Damian's work [8], achieving the appropriate participation of system users and field personnel is one of the challenges identified in these scenarios and that are caused both by inadequate communication and time difference. Therefore, the need to pay attention to the teaching of SRE in distributed environments of software development. The teaching model should promote and facilitate the collaboration of those involved, being the requirements elicitation techniques and the tools of communication key elements to serve this purpose.

Most of the current teaching models of SRE do not consider these new distributed scenarios. Moreover, they rarely attain practical professional experience, and teaching is frequently centered on theory, while students' seldom get involved in real projects [5].

This work presents a framework for teaching distributed software engineering. The framework addresses the execution of software engineering practices with teams that are formed by students and professors of Latin American universities. The model is supported by collaborative strategies based in Wikis, videoconferences, and emails. The model is applied on software requirements elicitation practices and involves activities of communication and interaction between all the distributed participants of the software projects (i.e., requirement engineers, developers, users, etc.).

The organization of the article is as follows: Section II presents recent initiatives in the area of SRE in distributed contexts. In Section III, the teaching framework proposed is presented in detail, including a description of the experiences on which the framework was founded. This section also shows an analysis and discussion of the results obtained after the application of the proposed framework. Finally, Section IV offers the conclusions of the work.

II. SRE Education

SRE is a vital part of the Software Development life cycle

[6]. Research endeavors in software development have found that failures and deficiencies of software systems are often rooted in the requirements activities undertaken [9], [14], [29], [31]. One possible reason of this is the lack of appropriate skills and knowledge of those engaged in SRE activities [3]. For this reason, the improvement of SRE teaching in universities should be a concern. Software Requirements Engineering Education (SREE) is an important topic and needs to be further explored [21]; however, the gap between what the industry needs and what the graduates learn from SRE courses is still very large [27]. In [6], the deficiencies in an SRE curriculum have been reported, and authors have emphasized the importance of requirements engineering education in Software Engineering and Computer Science programs and have provided recommendations for its improvement.

Many studies have been carried out to evaluate and to propose models, guidelines, frameworks, and solutions for SREE. However, academics and practitioners are still searching for the effective methodology to adopt, in order to deliver high quality and relevant RE courses [21]. Despite the increasing interest in SREE, we found few studies that discuss and try to address the education of Distributed SRE. Most studies found were focused on the application of SRE in the industry, under colocalized contexts.

The work of the reference [11] proposes a framework to integrate the SRE research and education to improve educational performance. In [24] and [25], the authors have used an experimental approach for teaching SRE using a business game board that demonstrates the social/design problems, complexities, and richness of a software development organization in the distress of creating a new product. In [30], the authors have also designed and used a board game to introduce SRE good practices to small novice organizations. Authors of [4] have described classroom experiences, successes, and challenges in teaching the unknown reality of SRE.

Authors in [17] shared an experience of teaching SRE by integrating it in several courses and challenging the students with authentic cases taken from business practices in which they had to apply theories and train their competencies at a Business Information Technology Program at the University of Twente. Authors of [7] discussed the design and delivery of a master's course in SRE designed to overcome some of the issues that have caused research-practice gap. Students were encouraged to share their experiences in a peer learning environment to improve the potential for effective collaborations, whilst simultaneously developing the requirements engineering skill sets of enrolled students. All the mentioned articles present studies carried out in colocalized scenarios. There are few works done in SREE under distributed contexts which are the focus of this paper.

A gap exists about GSD since this is neither considered within the curricula, nor does it explicitly form a part of the body of knowledge [26]. GSD [15], [16] is one of the current challenges of teaching and training in the requirements elicitation process [10].

Reference [20] exposes an SREE work in an undergraduate SE course, where students were able to develop and improve their skills in documentation activities and requirements management, using a template-based approach to document requirements high quality.

The work of [10] reported an experience of teaching a course intended for preparing graduates for SRE activities in global customer-developer relationships. The course was taught in a collaboration of three universities in disparate locations, time zones, and culture. The students from the three locations played the roles of a client and a developer, and they experienced the iterative development of a requirements specification in global projects. Standard communication tools were used, and the students were free to choose them. The Wiki is not identified as one of the tools used for communication or registration of requirements. Another issue is that elicitation techniques are not considered, being important in the framework proposed in this article.

In [22], there is an important design of a GSE course that uses the Scrum methodology with adaptations for this scenario, combined with better GSE practices, for the teaching of critical skills in GSE, such as team work and distribution communication, using appropriate communication tools.

The article in [26] proposes a simulating environment which, by using virtual agents, will enable students and professionals to acquire a subset of the skills necessary for requirements elicitation in GSD such as: the elicitation of requirements based on a stakeholder's needs using an interview technique and computer mediated communications, the ability to work in an international context, and an understanding of the cultures and customs of other countries. This is a proposal that has not been implemented yet, so there are no results regarding the educational achievements of it.

The educational framework presented in this paper is based on educational experiences carried out by different Latin American universities regarding the teaching of GSD. Specifically, the framework involves elicitation techniques and communication tools. It focuses on the requirements engineering stage, obtaining valuable conclusions for the SREE.

III. The Proposed Framework

The framework proposed for the teaching of the distributed requirements engineering at the university level of education is based on experimental works carried out with this purpose.

At the time of carrying out the experimental works described in this section, the teaching of the SRE in computer science programs in most Latin American universities was framed in colocalized scenarios. Taking into account this described situation and the emergence of the new GSD scenarios, the experiments considered the need to investigate requirements elicitation techniques and adequate communication tools for the requirements engineering process in these scenarios. These experiments made it possible to propose this framework for the teaching of the SRE in university computer science programs.

A. Distributed Elicitation of Software Requirements: An Experimental Case between Argentina and Colombia

In this case [32], acquiring knowledge from the stage of distributed elicitation of software requirements in the new scenarios of GSD was intended, determining the more convenient techniques of elicitation of requirements to use. A controlled experiment with two factors was carried out jointly with professors from the National University of San Juan (Argentina), the University of Cauca (Colombia) and the University of Quindío (Colombia). The factors were: context of elicitation (distributed/colocalized) and elicitation techniques used (three different combinations of techniques). Thus, two similar experimental phases were executed. The first applied to a distributed software development context, while the second was applied in a traditional colocalized context. The experimental design was finally a 2x3 factorial design as shown in Table I.

The elicitation techniques used in this experiment are the most used by small software development companies in Argentina: interview, questionnaire, and brainstorming [2], which is also true throughout Latin America. Understanding that, in real situations, the combination of techniques is used in the elicitation process, three alternatives (factors) of combination techniques to experiment were defined: interview/questionnaire, interview/brainstorming, and interview exclusively.

TABLE I
EXPERIMENTAL DESIGN WITH TWO FACTORS

		Applied Elicitation Techniques		
		Technique 1	Technique 2	Technique 3
Elicitation Context	Stage 1: Distributed	Students groups Tec1-Dist	Students groups Tec2-Dist	Students groups Tec3-Dist
	Stage 2: Colocalized	Students groups Tec1-Coloc	Students groups Tec2-Coloc	Students groups Tec3-Coloc

Advanced students from the different universities mentioned played the role of requirements engineers, while professors acted as remote clients/users. The requirements engineers, forming elicitation teams, were required to prepare a Software Requirements Document (SRD) as the final product of their work. That document would subsequently be the subject of quality measurement. In order to obtain the SRD, the requirements engineers had an initial descriptive document that presented a general and preliminary overview of the problem. Then, they had to apply the combination of elicitation techniques that were randomly assigned to each elicitation team until obtaining the final SRD. To achieve the goal, they had to interact with the client/user of the system to be developed.

The researchers who designed the experiment, prior to its execution and together with expert teachers, developed a Basic Requirements Document (BRD) which contained all the software requirements necessary to adequately satisfy the development of the requested system. This document, which was prepared and validated by expert professors, would be subsequently used as a reference for the evaluation of the SRD produced by students. In order to measure the quality of the resulting SRD, and that such measurement is an indicator of the effectiveness of the applied elicitation technique, the metric used in [18], [19] was chosen, adjusting it for this particular experimental case. In this way, the metric used in the present experiment involved four sub-indicators to be measured:

a) Adequacy Degree of the Document (ADD): It measures aspects of the organization of the document that arise as a result of the action of elicitation. For example: the classification of requirements in order of importance, feasibility, and/or implementation effort. This indicator takes values between 1 and 100.
b) Percentage of evolved requirements (ER): It measures the percentage of SRD requirements that are identified as an evolution of a software requirement that was already in the BRD. They are the requirements that needed a deeper and more thorough elicitation, which show a richer and more effective interaction between the requirements engineer and the client/user.
c) Percentage of requirements without defect (RWD): It measures the percentage of requirements that do not have precision defects, vagueness, ambiguity, etc. which means defects attributable to deficiencies in the elicitation process.
d) Percentage of supported requirements (SR): Percentage of requirements that are in the BRD are also found in the SRD produced by the requirements engineers. This sub-indicator is related to the completeness of the SRD produced. This indicator best represents a successful elicitation process. In case where this indicator reaches 100%, it means that all the necessary requirements for the application in development were discovered.

Integrating these four sub-indicators, each one with different weightings as shown in the following, the indicator of quality for the requirement specification document (QRSD) was obtained. This indicator indirectly reflects the effectiveness of the applied elicitation techniques.

$$QRSD = (0.05 * ADD + 0.3 * ER + 0.25 * RWD + 0.4 * SR) \quad (1)$$

The ER and SR sub-indicators are the ones that best reflect the effectiveness of an elicitation process, which is why they have a greater weighting in the QRSD indicator. At the end of each phase of the experiment, and according to the defined sub-indicators, a group of expert professors, which is external to the experiment, evaluated and qualified the SRD documents produced by the different elicitation groups. This process consisted in evaluating each SRD document presented by the groups of students taking as reference the BRD document prepared by the researchers and the metric defined above. It is important to note that prior to the start of the evaluation, a meeting with the experts was held in order to combine criteria for assigning values to each sub-indicator. The evaluation was made by a group of four expert professors, who first performed an individual review. A group review where they agreed on the final QRSD value of each SRD was subsequently done. The details of each of the two phases of the experiment are given below.

1) Experimental Phase Executed in a Distributed Environment

In this phase, 22 students and six professors participated. The students were randomly distributed in 11 groups of two students each. Each group forms an elicitation team. In turn, each group was randomly assigned a professor who would play the role of a client/user for such elicitation team. Also, at random, each group was assigned one of three combinations of techniques selected for this experiment. In this way, four elicitation groups that applied the interview technique were formed, four groups applied the interview-questionnaire combination technique; and finally, three groups applied the interview-brainstorming technique.

In order to simulate the distributed environment, communication between the requirements engineers and the clients/users was at all times via emails, chat, or IP videoconference. Both parties are free to opt for the specific application of their preference. It is important note that the requirement engineers and customers/users never interacted face to face. Different communication tools were used depending on the applied elicitation technique. For the interview and brainstorming, the subjects preferred to use videoconference. While for the questionnaire, it was mainly email and chat to a lesser extent. General-purpose standard communication applications were used to understand that these are the most widely used and most preferred by small software companies that are faced for the first time with the construction of distributed or global software.

This phase of the experiment took about ten days to complete. Table II shows the results obtained by each group regarding the quality of the resulting requirements documents.

TABLE II
RESULTS OF THE EXPERIMENTAL DISTRIBUTED PHASE

Group	Techniques	ADD	ER	RWD	SR	QRSD
1	Interview+Brainstorming	100.00	89.47	95.26	44.00	73.01
2	Interview+Questionnaire	80.00	94.74	87.11	42.00	70.80
3	Interview	90.00	81.48	92.96	46.00	70.36
4	Interview+Questionnaire	70.00	80.00	81.00	30.00	59.58
5	Interview+Brainstorming	70.00	65.38	89.23	60.00	69.25
6	Interview	90.00	93.94	98.18	46.00	75.40
7	Interview	100.00	55.56	84.81	32.00	55.42
8	Interview+Questionnaire	100.00	92.59	99.07	80.00	89.29
9	Interview+Brainstorming	80.00	80.00	92.25	32.00	63.66
10	Interview+Questionnaire	80.00	90.00	91.50	66.00	80.08
11	Interview	80.00	86.96	86.30	36.00	65.86

2) Experimental Phase Executed in a Colocalized Environment

In this phase, 18 advanced students participated along with five professors. Similarly, as described in the aforementioned phase, the students were divided into nine groups: three groups of elicitation that applied the interview technique, three groups applied the combination interview-questionnaire technique, and three groups applied the interview-brainstorming technique.

The communication between requirements engineers and customers/users was always on-site. Ten days included the execution of this phase approximately. Table III shows the results obtained by each group of students, discriminating the results of each one of the four sub-indicators (ADD, ER, RWD and SR).

TABLE III
RESULTS OF THE COLOCALIZED EXPERIMENTAL PHASE

Group	Techniques	ADD	ER	RWD	SR	QRSD
1	Interview+Questionnaire	92.50	85.71	91.25	36.00	67.55
2	Interview+Brainstorming	75.63	86.36	87.73	52.00	72.42
3	Interview	76.88	84.38	88.28	56.00	73.63
4	Interview	92.50	95.65	95.00	42.00	73.87
5	Interview+Brainstorming	75.00	93.33	91,67	52.00	75.47
6	Interview +Questionnaire	71.88	100.00	90.24	62.00	80.95
7	Interview	73.13	100.00	98.48	64.00	83.88
8	Interview +Questionnaire	74.38	95.56	85.33	76.00	84.12
9	Interview+Brainstorming	80.63	93.75	93.85	72.00	84.42

The average results of the QRSD can be seen in Tables IV and V, grouping them by the elicitation techniques applied in the two elicitation processes done. In the other words, they are distributed and colocalized, respectively.

TABLE IV
AVERAGE RESULTS OF THE DISTRIBUTED PHASE

Techniques	Average QRSD
Interview	66.76
Interview + Questionnaire	74.94
Interview + Brainstorming	68.94

TABLE V
AVERAGE RESULTS OF THE COLOCALIZED PHASE

Techniques	Average QRSD
Interview	77.12
Interview + Questionnaire	77.54
Interview + Brainstorming	77.44

The experimental data obtained suggest that the combination of more effective requirements elicitation techniques in a distributed context would be interview-questionnaire, which reaches a level of effectiveness that exceeds in 9% the next most effective combination of techniques, which was interview-brainstorming. It is also noted that the interview technique, used in a unique way, is the least effective of the techniques evaluated in this distributed context. In addition, according to the preliminary results obtained, it can be noticed that, in these new distributed scenarios, the average efficiency of the traditional techniques of elicitation would be 10% lower than the average of those same techniques applied in colocalized environments.

B. The Use of a Wiki in SRE under Distributed Contexts

The experience that is presented in [28] reports a controlled experiment conducted at the National University of San Juan (Argentina) using a Wiki in the SRE process in distributed software development scenarios. The Wiki used was Softwiki, a project funded by the Ministry of Education and Research of the German Government and supported by the universities of Duisburg-Essen and Leipzig.

In this second experiment, it was intended to teach students of the computer science program to use a Wiki (specifically designed for the SRE process), evaluating its performance in distributed software development scenarios in order to determine its incorporation into the teaching of the distributed SRE.

This experiment, as already mentioned, was carried out in a university context, where the subjects of the experiment were advanced students and professors of a degree course in computer science of the university. There were nine SRE groups or teams, each made up of two students playing the role of requirements engineers and one teacher, who served as the client/user. The work scenario was distributed. The requirements engineers were remotely located with respect to the clients/users. The assignment of students to the groups was randomly done, taking into account that the students had similar previous experience and knowledge profiles. The assignment of teachers was also random.

In order to simulate the distributed environment, communication between requirements engineers and clients/ users was always done by email, chat, videoconference, or wiki. At no time was there face-to-face interaction.

In the framework of the experiment executed, the requirements engineers, role that was assigned to the students, had to elicit information regarding the requirements of the software to be implemented. For this, they interacted with the clients; role that was assigned to the professors. The software to be developed, object of the SRE process, was a small/ medium size information system intended for the management of administrative information. The same problem was posed to all SRE groups that participated in the experiment. The information elicited was registered in the wiki and progressively and cooperatively converted into well-formulated software requirements.

At the beginning of the experiment, the requirements engineers were provided with an initial descriptive document that exposed a vague and general view of the problem, and from there, they had to start with the requirements engineering process until a definitive software requirement document (SRD) was obtained, using the communication tools mentioned. Requirements engineers specified in the Wiki tool, along with the client, the functional and nonfunctional requirements of the SRD, a document that would subsequently be subject to quality measurement.

As for the only requirements elicitation technique employed in this study, one of the most commonly used by small software companies in Argentina was chosen [2]. Prior to the execution of the experiment, the research team developed a Reference Software Requirements Document (RSRD), which contained all the necessary software requirements to adequately satisfy the development of the requested system. This RSRD would later be used as a reference for the SRD evaluations produced by the requirements engineers of each group.

To verify the contribution of the wiki usage in order to obtain a higher quality SRD, only three quality attributes of RSRD were considered: Completeness, Non-Ambiguity, and Precision. To satisfy these attributes, it is necessary a reflexive interaction (of meditated responses, not instantaneous) based on textual support (to analyze mainly ambiguity and precision) between the requirements engineers and the clients. These characteristics make one think that the wiki would be a valid tool for that process. For this purpose, three indicators were used for these attributes:

a) The RSRD Completeness Indicator (q1) determines the extent to which the requirements present in the RSRD are supported in the SRD of the group. The result is a ratio (values are given between 0 and 1). As this value approaches 1, the degree of completeness is greater.
b) The Non-Ambiguity of Requirements Indicator (q2) determines the degree of non-ambiguity of RSRD requirements. The result is a ratio between the requirements evaluated as unambiguous and the total requirements of the SRD. As this value approaches 1, the greater the degree of non-ambiguity of the SRD produced.
c) The Accuracy of Requirements Indicator (q3) determines the degree of accuracy of the RSRD requirements. The result is a ratio between the requirements evaluated as accurate and the total requirements of the SRD. As this value approaches 1, the greater the degree of accuracy of the SRD produced.

A group of experts from the UNSJ, foreign to the experiment, evaluated the SRD produced by the students in order to determine these indicators. Prior to the evaluation, the experts met to agree and unify the weighting and assessment criteria of each one of the indicators.

Table VI shows the results obtained by each SRE group regarding three quality attributes of the Software Requirements Document: Completeness, Absence of Ambiguity and Precision. Another index measured is the degree or level of use of the wiki tool done by each group. This index was obtained from the own log records of the wiki tool.

By doing an analysis of correlations between the Use of Wiki Index and each one of the indicators q1, q2 and q3, diverse conclusions or indications can be reached. Thus, it turned out that there is no evidence that the increased use of the wiki promotes more complete and less ambiguous requirements documents in distributed environments. However, there was a high level of correlation between the Use of Wiki Index and q3, which raises the suspicion that the greater use of the wiki improves the accuracy of RSRD.

Table VII shows the use of communication tools on the different days of the experiment, consolidating the data of all SRE groups. The emails column records the number of emails exchanged between requirements engineers and customers/ users. In the Wiki Versions column, the total number of requirements new versions recorded in the wiki tool is counted per each day of the experiment. In the next column called Wiki Comments, the total number of comments related to the requirements that were registered in the wiki tool by the requirements engineers and the clients/users are counted. These last two columns give evidence of the use of the wiki. Finally, the VC column counts the total minutes of use of the videoconference communication tool.

TABLE VI
OBTAINED RESULTS

Group	Use of Wiki Index	Compleness q1 Index	Non- Ambiguity q2 Index	Precision q3 Index
G1	1.725	0.500	0.800	0.650
G2	1.529	0.540	0.941	0.765
G3	2.345	0.520	0.840	0.800
G4	2.300	0.460	0.700	0.750
G5	2.020	0.600	0.800	0.520
G6	1.618	0.500	0.618	0.382
G7	1.750	0.780	1.000	0.538
G8	1.339	0.520	0.731	0.346
G9	1.283	0.520	0.652	0.174

It should be considered that the first four days were filled with activities of initial interaction (presentation, approximation, etc.) between the parties. It was agreed to carry out the first interview in that initial period. That is why it can be said that the elicitation activity actually began on the fifth day. Taking into account the aforementioned, it is inferred from the observation in Table VII that videoconferences have been mainly made in the first sections of the SRE process. This supports the hypothesis that this communication tool is mainly used to begin the general approach of the problem, trying to be an option to determine the generalities of the problem, its scope, without going into detail.

TABLE VII
OBTAINED RESULTS

Day	Emails	Wiki Versions	Comments Wiki	VC
1	1	0	0	0
2	5	0	0	0
3	3	0	0	0
4	0	0	0	0
5	13	0	0	48
6	11	0	0	0
7	7	83	0	132
8	1	43	17	0
9	3	111	15	84
10	1	4	3	47
11	0	0	0	0
12	5	20	48	0
13	17	42	43	0
14	18	149	125	18
15	3	19	17	0
16	2	25	21	0

Unlike videoconferences, it is noticed that the activity in the wiki tool is stipulated with greater emphasis in the final sections of the SRE process. This adds to the belief that the wiki is a tool that helps in the search for accuracy of software requirements, an action that is usually executed right at the end of the software requirements process.

The email exchange activity is the only activity that maintains at least one moderate activity throughout the SRE process. It is likely that this is due to the fact that this tool is used as a means of communication of actions and synchronization tasks between the requirements engineers and the customers/users (initial contact, transfer notifications of tasks control, tasks validation requests, final agreement regarding the requirements document, etc.), being these actions present throughout the whole SRE process.

C. Integral Analysis of the Experiments

In the first experiment, it is noticed that the combination of requirements elicitation techniques Interview along with Questionnaire are an effective alternative for SRE environments where the client/user and the requirements engineer are geographically distant. It is noted that the interviews were used at the beginning of the requirements elicitation process, when a more general notion of the client/ user's needs regarding the system to be developed was required while the questionnaire technique was used in order to obtain more specific or focal information.

The findings of the first experiment are compatible with those found in the second one. In the latter, in addition to showing that the wiki tools help to obtain a more precise requirements document, it is also observed that videoconferences are used at the beginning of the SRE process. Furthermore, the wiki reaches its greatest use towards the end of the process; and finally, emails are used throughout the whole process, mainly to communicate coordination activities among remotely located actors.

D. Framework Definition for Teaching Distributed Requirements Engineering

Based on the experimental evidence obtained and presented in the previous section concerning the requirements elicitation techniques and communication tools used in GSD scenarios, we propose in this article a framework for the teaching of SRE in a GSD scenario where the client/user is remotely located with respect to the requirements engineer. The proposed framework can be seen in Fig. 1.

The framework proposes, based on the experiences made, the use of requirements elicitation techniques and communication tools in different phases of the SRE process. The proposal tries to maintain a continuous interaction with the client/user during the whole process, having the same different degree of protagonism in the different phases.

In this sense, the wiki, specifically configured for the SRE process, is a communication and repository tool that records the progress of the process. Specifically, the software requirements established with its associated comments. The Wiki is accessed by both the requirements engineer and the client/user. The information that is generated in each phase of the process is stored in the wiki that supports the process model.

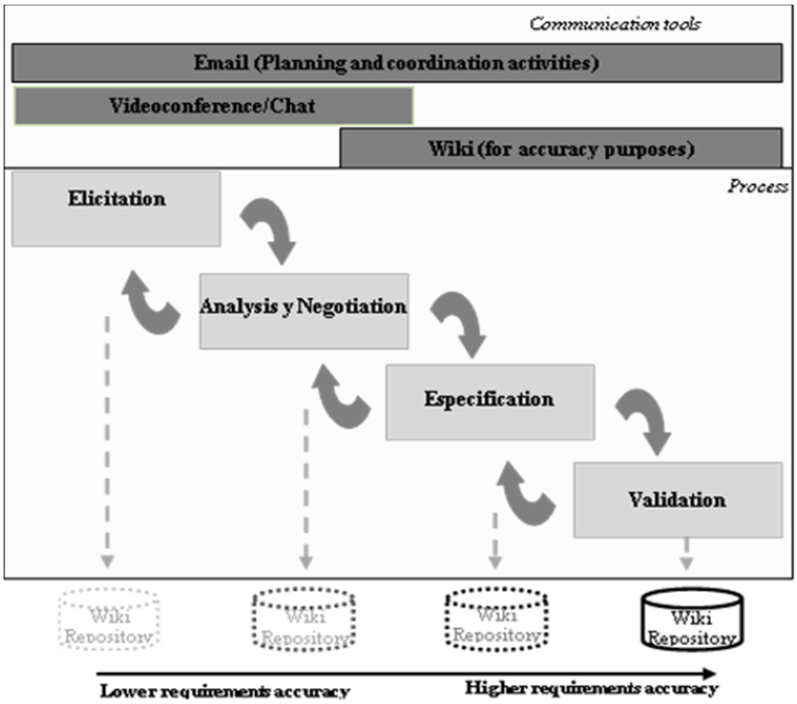

Fig. 1 Framework for teaching Distributed SRE

Based on the experimental evidence presented in this paper, it is proposed that the asynchronous communication provided by the emails is used throughout the whole SRE process, in all phases. Normally, emails support the exchange of planning and coordination messages between the parts of the process. For example, video-conference requests, termination of activities notifications, alarms of expiring tasks, and dissemination of aspects related to the progress of the project, etc. It is for this reason that the use of emails appears throughout the whole process.

At the end of the process, the set of software requirements that will make up the DRS core of the system will remain registered in the Wiki Repository.

In the proposed framework, the SRE process is divided in four different phases:

a) Elicitation Phase: The framework establishes that in this first phase of the process, the interview technique must be done through videoconference. This first approach between the client/user and the requirements engineer aims to obtain a general definition, not a precise one, of the problem to be solved. This is why, as the experiments show, the unstructured interviews are adequate for this phase. To this end, the videoconference is a communication tool of relatively high communicational richness that adequately supports a remote interview. As the knowledge of the problem progresses, the use of questionnaires to obtain a greater accuracy can be made. They can be managed appropriately through an asynchronous communication tool such as the email or web tools of specific use for questionnaires such as Google Forms.
b) Analysis and Negotiation Phase: The framework establishes that this phase of the process aims to analyze the clarity, pertinence, and technical feasibility of the software requirements. For this purpose, it is necessary to interact with the client/user to negotiate aspects related to effort and technical feasibility of the implementation of the requirements. In this phase, interviews must be done through videoconferences. As in all phases, the advances in terms of definition of software requirements are recorded in the Wiki Repository. At the end of this second phase is when the wiki tool as a means of asynchronous communication between the client/user and the requirements engineer begins to take preponderance.
c) Specification Phase: This phase of the process aims to establish a set of well-formulated software requirements. In other words, requirements with sufficient quality attribute to ensure the success of the later stages of the software development process. The framework proposes the use of the wiki tool in this phase. With it, the requirements engineer can review and edit the requirements to improve its quality while the client/user can verify, modify, and validate the changes. In this phase, a type of asynchronous, textual, reflexive, and focal interaction between the parties begins, so videoconferences are no longer useful, turning the wiki into the right tool. This phase requires more detailed work on the requirements, especially in order to give them precision. This is where the wiki tool brings great benefit as evidenced by the experimental data mentioned in this paper.
d) Validation Phase: Finally, in this phase, both parts of the process, client/user and requirements engineer, validate the set of software requirements identified and exposed

clearly in the Wiki Repository. Again, this tool supports these tasks of integral revision and validation.

E. Discussion

The framework proposed for the teaching of SRE to university students presents great potential for training in the most used elicitation techniques, as well as in the communication tools, and in the solving of real problems in distributed scenarios. This was reflected in the high values of the Intensity of Use and Satisfaction dimensions obtained from the processing of the surveys carried out to the participating students (Figs. 2 and 3). To obtain these graphs, a descriptive statistical analysis was performed for each dimension.

From Fig. 2, it is deduced that the results obtained are satisfactory as 75% of the values are greater than or equal to 4. As Fig. 3 shows, 81.3% of the results obtained are greater than or equal to 4, which means that they are highly satisfactory.

It was also evidenced that the application of the elicitation techniques and communication tools proposed in the framework would positively influence the quality of the SRD produced by the requirements engineers (students). On the other hand, the proposed framework, according to the opinions of students and professors who participated in the experience, could be improved by including the work products used and generated in each phase of the SRE process.

Another weakness of the framework is the lack of elements to deal with problems inherent to cultural differences that arise in these distributed contexts.

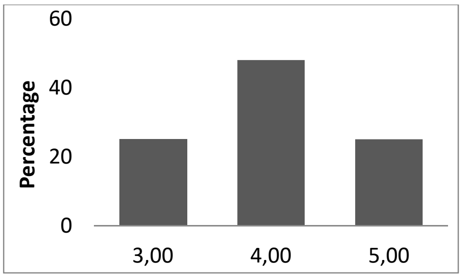

Fig. 2 Frequency graph corresponding to the Intention of Use dimension

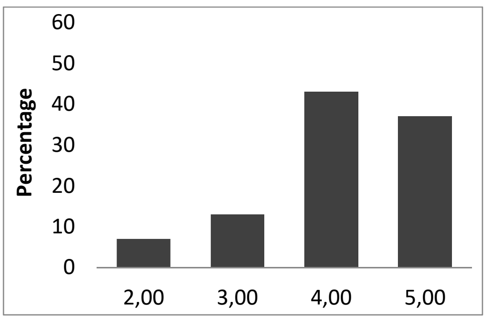

Fig. 3 Frequency graph corresponding to the General Satisfaction dimension

IV. Conclusions

In this work, we present a framework for the teaching of SRE for distributed scenarios.

The conception of the framework is based on controlled experiments carried out in university environments of Latin American countries. The framework proposes a method of teaching that incorporates both adequate techniques of software requirements elicitation and validated tools of communication, crucial for this type of global scenarios.

The additional motivation of students to participate in these practices in conjunction with peers from other countries is a significant additional factor that positively contributes to the learning process. The use of standard communication tools, easily available in Latin American universities, makes the proposed framework feasible to implement. Although the results have been satisfactory in a first controlled experiment, it must be replicated in order to reach stronger assertions.

Acknowledgment

This work was partially funded by the project entitled "Collaboration and Trust in Global Software Engineering", code UNSJ2018. We also would like to thank students and professors of Latin American Universities that participated on in this study.

References

[1] A. Abran, J. W. Moore, P. Bourque, R. Dupuis and L. L. Tripp, Guide to the Software Engineering Body of Knowledge (SWEBOK) version 3 public review. IEEE, 2013. ISO Technical Report ISO/IEC TR 19759.

[2] L. Antonelli and A. Oliveros, "Fuentes utilizadas por desarrolladores de software en argentina para elicitar requerimientos." *In WER (2002)*, pp. 106–116.

[3] J. Armarego, and O. Minor, "Studio learning of requirements: towards aligning teaching to practitioner needs," *in REET'05 (1st International Workshop on RE Education and Training)* (2005), REET.

[4] R. J. Barnes, D.C. Gause, and E. C. Way, "Teaching the unknown and the unknowable in requirements engineering education," *in Requirements Engineering Education and Training, 2008. REET'08,* 2008, IEEE, pp. 30–37.

[5] U. Bellur, "An academic perspective on globalization in the software industry," *in Computer Software and Applications Conference, 2006, COMPSAC'06. 30th Annual International (2006),* vol. 1, IEEE, pp. 53–54.

[6] B. Berenbach, "A hole in the curriculum. In Proceedings of the 1st International Workshop on Requirements Engineering Education and Training," *(REET-05)*, Paris, France (2005).

[7] A. M. Connor, J. Buchan and K. Petrova, "Bridging the research-practice gap in requirements engineering through effective teaching and peer learning," *in Information Technology: New Generations, 2009, ITNG'09, Sixth International Conference on (2009)*, IEEE, pp. 678–683.

[8] D. Damian & D. Zowghi, "RE challenges in multi-site software development organisations," *in Requirements Engineering 8.3*, 2003, 149-160.

[9] D. Damian and J. Chisan, "An empirical study of the complex relationships between requirements engineering processes and other processes that lead to payoffs in productivity, quality, and risk management," *in IEEE Transactions on Software Engineering, 32*, 7, 2006, 433–453.

[10] D. Damian, A. Hadwin and B. Al-Ani, "Instructional design and assessment strategies for teaching global software development: a framework," *in Proceedings of the 28th international conference on Software engineering*, 2006, ACM, pp. 685–690.

[11] A. M. Davis, A. M. Hickey and A. Chamillard, "Moving beyond the classroom: Integrating requirements engineering research & education to improve practice," *in Proceedings of the 1st international workshop on requirements engineering education and training, REET,* 2005, vol. 5, Citeseer, pp. 78–87.

[12] A. Duran Toro, *Un entorno metodológico de ingeniería de requisitos*

para sistemas de información, 2000.

[13] R. W. Ferguson and G. Lami, "An empirical study on the relationship between defective requirements and test failures," *in Software Engineering Workshop*, 2006. SEW'06. 30th Annual IEEE/NASA (2006), IEEE, pp. 7–10.

[14] S. Group, *Chaos summary 2009*. Online report. Accessed June 20 (2009).

[15] J. D. Herbsleb, "Global software engineering: The future of socio-technical coordination," *in 2007 Future of Software Engineering (2007), IEEE Computer Society*, pp. 188–198.

[16] J. D. Herbsleb and D. Moitra, "Guest editors' introduction: Global software development," *in IEEE Softw*. 18, 2 (Mar. 2001), 16–20.

[17] C. Huijs, K. Sikkel and R. Wieringa, "Mission 2 solution: Requirements engineering education as a central theme in the bit programme," *in 1st International Workshop on Requirements Engineering Education and Training (REET'05)*, 2005.

[18] W. J. Lloyd, *Tools and techniques for effective distributed requirements engineering: An empirical study*. PhD thesis, Virginia Tech, 2001.

[19] W. J. Lloyd, M. B. Rosson and J. D. Arthur, "Effectiveness of elicitation techniques in distributed requirements engineering," *in Requirements Engineering, 2002.* Proceedings. IEEE Joint International Conference on (2002), IEEE, pp. 311–318.

[20] D. Monett Diaz, S. Kujat, M. Hartmann, "Teaching Students Collaborative Requirements Engineering. Case Study Red:Wire," *in World Academy of Science, Engineering and Technology.* International Journal of Computer, Electrical, Automation, Control and Information Engineering. (2016), Vol.10, No. 5). S. 176 – 783.

[21] S. Ouhbi, A. Idri, J. L. Fernandez-Aleman and A. Toval, "Requirements engineering education: a systematic mapping study," Requirements Engineering 20, 2 (2015), 119–138.

[22] M. Paasivaara, C. Lassenius, D. Damian, P. Räty & A. Schröter, "Teaching students global software engineering skills using distributed scrum," *in Software Engineering (ICSE),* 2013 35th International Conference on (pp. 1128-1137). IEEE.

[23] P. Rajagopal, R. Lee, T. Ahlswede, C. C. Chiang and D. Karolak, "A new approach for software requirements elicitation," *in Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks. SNPD/SAWN 2005. Sixth International Conference on (2005), IEEE, pp. 32–42.

[24] G. Regev, D. C. Gause and A. Wegmann, "Requirements engineering education in the 21st century, an experiential learning approach*," in International Requirements Engineering*, 2008. RE'08. 16th IEEE (2008), IEEE, pp. 85–94.

[25] G. Regev, D. C. Gause and A. Wegmann, "Experiential learning approach for requirements engineering education," Requirements engineering 14, 4 (2009), 269–287.

[26] M. Romero, A. Vizcaino and M. Piattini, "Teaching requirements elicitation within the context of global software development," *in Computer Science (ENC)*, 2009 Mexican International Conference on (2009), IEEE, pp. 232–239.

[27] L. Scheinholtz, "What are employers really looking for*?," in Proceedings of the Requirements Engineering Education and Training Workshop (REET-07)*, 2007.

[28] G. Sevilla, S. Zapata, E. Torres and C. A. Collazos, "Using wikis as collaborative strategy to support software requirements elicitation," *in Computing Colombian Conference (9CCC),* 2014 9th (2014), IEEE, pp. 54–61.

[29] J. M. Smith, *Troubled IT projects: prevention and turnaround*, vol. 3. Iet, 2001.

[30] R. Smith and O. Gotel, "Re-o-poly: A game to introduce lightweight requirements engineering good practices," *in International Workshop on Requirements Engineering and Training*, India Habitat Center, New Delhi (2007).

[31] I. Sommerville and J. Ransom, "An empirical study of industrial requirements engineering process assessment and improvement," *ACM Transactions on Software Engineering and Methodology (TOSEM) 14*, 1, 2005, 85–117.

[32] 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 Informatica (CLEI),* 2012 XXXVIII Conferencia Latinoamericana En (2012), IEEE, pp. 1–7.