Teaching Software Engineering from a Collaborative Perspective: Some Latin-American Experiences

Fáber D. Giraldo
System and Computer Engineering Department
University of Quindío
Armenia, Colombia
fdgiraldo@uniquindio.edu.co

César A. Collazos
IDIS Research Group
University of Cauca
Popayan, Colombia
ccollazo@unicauca.edu.co

Sergio F. Ochoa
Computer Science Department
University of Chile
Santiago, Chile
sochoa@dcc.uchile.cl

Sergio Zapata
Informatics Institute
National University of San Juan
San Juan, Argentina
szapata@iinfo.unsj.edu.ar

Gisela Torres de Clunie
Software Engineering Department
Technological University of Panama
Panama City, Panama
gisela.clunie@utp.ac.pa

Abstract— **Teaching software engineering has been recognized as an important challenge for computer science undergraduate programs. Instruction in such area requires not only to deliver theoretical knowledge, but also to perform practical experiences that allow students to assimilate and apply such knowledge. This paper presents some results of two Computer-Supported Collaborative Learning (CSCL) experiences that involved students of software engineering courses from four Latin American Universities. The obtained results were satisfactory and indicate the reported collaborative activity could be appropriate to address teaching software engineering.**

Keywords: Collaborative Activity, Computer Supported Collaborative Learning, Software Engineering, Collaborative Work, Collaborative Groups.

I. Introduction

Maybe, Software Engineering is one of the most critical area of Computer Science at academicals and industrial levels; professionals who ended their studies in Computer Science or Informatics Engineering can find a lot of opportunities and enterprise jobs, where offer do not answer the growth market demand. Trends such movement of software development offshore to the lowest-cost locations, growth and maturation of languages and platforms and the reliance on third-party components for core system functionality, mean that the mix of competencies required to successfully practice software engineering includes much more than just software development skills [17]. These issues have placed an increasing demand for software developers (and engineers) who are equipped not only to deal with the scientific and technical aspects of computing, but for those who have professional education and preparation for the practice of software engineering [20]. Changes in software technology and models for software development require commensurate change in the education of software engineers. Educational institutions themselves must be able to adapt quickly, both in the content of their offerings and in their ability to exploit new technology in support of education. Second, the educational institutions must prepare their graduates to accept responsibility for upgrading their own skills throughout their careers [18]. Teaching Software Engineering is complex due to develop of specific skills through theoretic and practical activities. Practice-based software engineering is a very best practice in the contemporary Software Engineering and Computer Science education [19].

Software engineering involves not only the software product and process, but also the team work. The software development process and the team work involve collaborative work as a key point. Therefore, a CSCL (Computer supported Collaborative Learning) strategy could be useful to address the challenge to teach and learn software engineering in undergraduate and graduate programs.

This paper presents a model that supports the teaching-learning process of software engineering topics using collaborative strategies among geographically dispersed groups of students and professors. The main task of modern higher education systems is preparing students for participation in an information society where knowledge is the most critical resource for social and economic development, where the creation of networks for knowledge sharing is a new skill. Effective communication, negotiation skills and creating new knowledge to critically assess information resources or a product, are transferable skills that higher education institutions must provide to their students.

Collaborative learning involves intellectual work together to pursue certain learning results. It may be better as far as learning partners give different perspectives to a problem or issue [1]. Information technologies based on Internet gives to students the opportunity to "talk" or interact with peers from different countries, and develop skills such as those mentioned above. An important part in this process is a reflection about learning a concept, skill or topic through a discussion with another. Learning together is a model used in higher education to promote reflection on learning, either through joint projects or helping others to understand learning support material. In the dialogue between students we can obtain multiple perspectives, creating a cognitive conflict, promotes the development of critical skills and capabilities to professional debate, objectivity, and discursive reflection [4].

The collaborative learning activity proposed in this paper can be used to teach almost any issues belonging to the software engineering domain. This activity has been evaluated in a preliminary way with students from four Latin American universities. The obtained results are highly encouraging.

Next section 2 presents related work. Section 3 describes the proposed pedagogical model. Section 4 presents the experimentation process. Section 5 shows and discusses the obtained results. Finally, section 6 presents the conclusions and future work.

II. RELATED WORK

Each CSCL activity follows a sketch that is performed by the students to reach a common goal. Typically the goal is related to acquisition of a particular knowledge or capability. These activities involve interactions among the participants through computers. In order to receive the full benefit of social learning, students must interact with each other, share information and coordinate their actions. Unfortunately, the research work in this area indicates that computer mediation creates potential obstacles to student-student interaction. Specifically, team members tend to experience a slower development of trust, cohesion, efficiency and knowledge sharing, when interact through computers. It usually impacts negatively on the effectiveness of the interaction among participants [2]; in our case, among students. Developers of collaborative learning supporting tools must be creative to promote effective student-student interactions [12].

Currently there are proposals for including collaborative models in teaching of several knowledge areas, such as Artificial Intelligence [10], Programming [13] and Expert Systems [3]. There are also initiatives that incorporate new teaching-learning strategies for software engineering courses. For example, Manjarres et al. [9] have specified a participatory approach to conduct a software engineering undergraduate course in Spain. The participatory approach involves a practical activity consisting of the development of a project based on free software. Such project involves analysis, design and implementation of an application for management partners and volunteers for the Engineers without Borders (ISF) organization. There students are integrated into the development team of the organization, capable of performing their collaboration by distance interaction. This collaboration supposes the practice in different engineering techniques and internalization of the intrinsic values to free software development paradigm [9]. Mesa and others have proposed a strategy for teaching software engineering from the perspective of PBL (Problem-Based Learning), making a coordinated effort for the proper execution of project [11]. However, none of these initiatives consider working with people geographically dispersed using collaborative models, which is the basis of our proposal.

III. PEDAGOGICAL COLLABORATIVE MODEL

Our proposed model pretends to promote learning through collaborative work in a scenario where different participants (students, teachers and mediators) are geographically distributed. In order to validate the benefits of the proposed model we have performed the same experimental practice compared with a traditional model of teaching-learning, called ad-hoc. To summarize, the working hypothesis that this article attempts to show is that through a distributed collaborative instructional process based on a structured technique such as Jigsaw[1], instructors can achieve better results than applying a non-collaborative instructional process (ad hoc) distributed. Model involves the following types of actors:

- *Expert Professor*: He/she is responsible for imparting knowledge and defines the practical activities to be performed by students. It is usually a professor that belongs to one of the participating institutions.
- *Mediator professors*: They are monitoring activities of students. They resolve basic technical questions formulated by students, and issues related to development of the educational process. At least there is one teacher facilitator for each participating university.
- *Students*: They are the central actors and final recipients of the teaching-learning process. They belong to courses of Software Engineering courses from several participating universities.

A. Phases of the Model

During the first phase the expert instructor introduces a topic or subject, in both cases (2008 and 2009) topics were *effort estimation in a software development project*, and *usability evaluation techniques*. Expert (professor) guides the subject through video-conferencing to geographically dispersed students.

In the second phase, students must do a practical activity in a distributed way, which is related to the subject exposed. Groups are organized by selecting students from different universities. It is recommended that groups are conformed of at least one member of each participating institution. In this

[1] www.jigsaw.org

stage groups follow a particular dynamic of work under the monitoring of mediators teachers.

In the third phase of model, teachers assess the work of the practical activities performed by students in the previous phase. This assessment usually involves an individual and group rating of students. Also in this phase professors evaluate the teaching-learning process, usually through surveys from students, in order to obtain indications that help to provide feedback and improve the process. In the experience of year 2008 (*software effort estimation*) professors used a practice called *local practice* before *distributed practice* exposed in second phase. Mail goal of this activity was preparation of students for distributed practice, resolving an exercise similar to proposed by expert but in local interaction with their partners in the same geographic location.

B. Operation of collaborative groups

In 2008, students was free for choose the best way for interacting with their geographical partners. In 2009, under the proposed model, groups of students work as collaborative groups in order to obtain the benefits of learning this way of working [21]. Groups adhere to a dynamic and structure of work similar to established by JIGSAW technique. In this instance, expert professor distributes material to students and goals of practical activities. These practical activities are divided into many parts as members have the group, so each student is responsible for one of these parties and performs his-her task individually. After, students form pairs of specialists conformed by participants from different groups, but the same university, which were assigned the same piece of work. These pairs discuss their work, and eventually may correct or adjust the results for their individual work. This instance of specialization is performed in a co-located way. Finally each student returns to his-her original group and he-she must explain and discuss the results of their work to their peer group. This activity is repeated with each team member, therefore, to end this process of knowledge sharing, all members could learn about the different specialties thematic involved in each of practical activities. This activity is performed in a distributed way for students from several universities. At the end, each group should consolidate results achieved by their members and agree on a single group result, it must be delivered to expert professor. This could be, for example, a report group.

IV. EXPERIENCES APPLYING THE MODEL

In 2008, research teams have made a first application of model in an experience where topic used was effort estimation of software projects; in order to students acquire knowledge and skills in this type of techniques. We have worked with students from three geographically separated universities: University of Quindío (Colombia), University of Chile (Chile) and National University of San Juan (Argentina). Students are in the final years of the degree programs of Systems and Computer Engineering or Computer Science from their respective universities.

In this experimentation, the session class could not be performed simultaneously in the three universities, because there were some technological problems generated by ISPs. A Professor of the University of Chile gave the distance class to 19 students from the National University of San Juan (Argentina). This online class was conducted using the VoIP functionality of the Skype software and Conference XP. These platforms allow share slides, desktop sharing, working remotely, and provide support for teacher interaction with the student group [8]. Then, the same teacher repeated the lecture to students in Colombia with participation of 22 students in the University of Quindío (Colombia). The working group of University of Quindío used the network RENATA as the basis for collaborative virtual environment with the University of Chile, site from which originated the session. Academic Networks, as in the case of RENATA, facilitates collaborative academic work, to share information, access to laboratory equipment, high-volume data transfer, development of applications and support distributed processing complex experiments are crucial for research . Similarly, these networks facilitate communication and teamwork among researchers who are geographically dispersed in different regions, thus facilitating the development of joint projects of academic nature, science and technology. Finally, Professor gave the class in traditional way (expository and co-localized) to students at the University of Chile. This group consisted of 21 students from an undergraduate program in Computer Science.

Distributed practice (i.e. the second phase of the model) was performed through distributed working sessions, in both modes (synchronous and asynchronous). The research team from the University of San Juan identified 21 groups of work, each group consisting of 3 students, one student from each country involved in the experience. Formation of the groups was randomly, respecting the above condition. Students were to solve a practical but with a greater level of complexity that presented by Professor of the University of Chile. Students were free to choose the communication tools that they used. Each student was provided with contact information of other members of their group, so that students could communicate with each other through email and instant messaging systems. Then, researchers from the San Juan National University sent the specification of distributed practice agreed at the preparation phase.

In general, during 2008 students were highly motivated to interact with peers from other universities and, beyond the constraints of mediated communication technologies; the motivation was maintained throughout the process. Students used, at their option, multiple media such as chat, VoIP, email, and video-conferencing. Students had to "negotiate" with each other to arrive at agreements about technologies used for communication, ways to solve the problem, communication schedules, and distribution of tasks, among others. These instances of negotiations forced them to implement skills and competencies few used by them, but very importance in their training. Finally, researchers performed an evaluation of the experience, which is the last phase of the model, which consisted of satisfaction surveys and co-peer evaluations.

In 2009, this experience has involved undergraduate students in Software Engineering courses from several institutions: 23 students from University of Cauca (Colombia), 36 students from the University of Quindío (Colombia), 56 students from the Technological University of Panama (Panama), and 20 students from the National University of San Juan (Argentina). During the experience, researchers have contrasted distributed collaborative model proposed (JIGSAW) versus a non-distributed model of collaborative teaching (Ad-hoc), which was based mainly on traditional interaction patterns. In this model, ad-hoc groups of students followed work and coordination forms freely chose by them.

As indicated by the model in its first phase, process began with the presentation of a topic based on usability evaluation, showing techniques of inquiry, inspection, test and mechanisms for evaluating accessibility in interactive environments. This session was conducted by a Professor from University of Cauca (Colombia). Professor gave theoretical and practical material relating to each evaluation technique. Presentation was delivered in the same time, local to host university of expert teacher, and videoconferencing for the rest of the participating universities through the Microsoft LiveMeeting. This class was taught to all students who participate in the experiment, without distinguishing between groups of students collaborative and ad-hoc groups. As part of second phase of the model, students were distributed in collaborative groups and ad hoc groups (control groups). Collaborative groups adhered to the process explained before (JIGSAW approach), while ad-hoc groups were free to organize as they want. In this experiment involved 17 collaborative groups and 16 ad-hoc groups. Each group consisted of 4 or 5 students, where there was at least one member of each participating university. Task assigned to students was to conduct a usability test form Web site of the Technological University of Panamá (www.utp.ac.pa). For the experiment we used the e-learning environment ***AulaNet***, at the ***UTPVirtual Center*** of the Technological University of Panama. For the development of activities various services were enabled in this environment.

At the end of the activity, each group presented a report evaluating the web site using some approach exposed in the class. Mediator professors in each participating university contributed to this process resolving some questions generated during activity. Finally, as part of the third phase of the model, students answered a survey about the executed experience, which was evaluated again by researchers to gain feedback and adjusting the process. In this phase expert professor also evaluated quantitatively the reports of student groups.

It is very important to emphasize that collaborative groups (CG) had a working way default and explicitly based on JIGSAW[2] model. Each team member had assigned a particular technique for evaluating the usability of user interfaces. Therefore, each member had to study individually assigned technique, becoming a specialist in it, and apply it to evaluate the usability of the website chosen. Then the students of the same technical specialists are "gathered" to share expertise. Finally, collaborative groups of students returned to "meet", now with a specialist in each technique, and they produced a single technical report which was agreed between them. In each instance students use standard communication tools such as discussion forums, chat, email, IP videoconferencing, and instant message. The ad-hoc groups (GAD) served as control groups in our experiment. Unlike collaborative groups, they were free to organize and coordinate work in the like they chosen. These groups received the same initial class and they were asked for the same final technical report like collaborative groups. Specific activities were not required to executed work; each of these groups organized their own way to obtain the final result. Mediator professors followed up students' activities throughout all process.

[2] www.jigsaw.org

V. OBTAINED RESULTS

In 2008, we can observe that results of academic practices were good and satisfaction of all students was high. However, we had some technological difficulties and the students had some problems to agree in order to work in a synchronous way. Initially all three countries had different times, and Chile during the experience has changed its time zone. As a general conclusion, students in the programs in Informatics and Computer Science from three universities expressed, by the anonymous satisfaction survey, to be very pleased with the learning experience, which enriched their education process through interaction with peers from other universities separated geographically. This experience, additionally to the value of collaborative learning and the intensive interaction between students from different cultural and social contexts, can also be considered as a distributed laboratory to control experiments in software engineering practices.

Particularly in 2009, at the end of experimentation process, professors evaluated results obtained by the groups, and found that collaborative groups performed better than Ad-hoc groups. However, the difference was not significant in terms of the skills of group reports; average score in the collaborative groups was 93.53 versus 91.8 of Ad-hoc groups (on a scale of 0-100). From findings that researchers have discovered we can determine that having a working strategy involves initial adequate performance. So those collaborative groups, in which the strategy was imposed by teachers early, had a good overall performance. Supporting this similar perception, ad-hoc groups that do well performance are those who have defined themselves the working strategy. Hence, it is possible to assume that if group works with an ad-hoc or collaborative approach, has no direct implication on the work result. Rather, it is possible to conclude that groups were organized early, did better than those who did not. The Ad-hoc groups had more messages exchanges than collaborative groups. This is understandable because possibly collaborative groups, who had defined the strategy work, do not need discuss this topic. However, Ad-hoc groups need it. The most important theme is to have a common element: to define an appropriate strategy, commitment and a common goal, as is presented by Collazos et al [15].

VI. CONCLUSION AND FUTURE WORK

Software Engineering and its related professions are constantly changing. This obligate us not only to train professionals in order to manage a series of knowledge and current technologies, but to constantly refine curriculum which includes this profession. Working in groups is a mechanism that allows to students to gain experience in building software.

Experience in teamwork should be complemented with contemporary trends imposed to software, since it become one of the sectors most affected by the phenomenon of globalization and market opening, in addition to the geographical distribution of customers and need for establish principles of industrialization for build software, and internationalization of best practices, standards, architectures and technological platforms. It is necessary to strengthen local teaching of software engineering by providing regional experiences in software construction applied to collective efforts in teaching-learning software [16].

Technological changes and organization of people involved redesigning pedagogical models where work can be accomplished in different scenarios with people geographically dispersed. Evidence from the experiences presented in this paper must be corroborated and expanded by new applications of the proposed model. For future experiments we will be consider topics suggested by students in the anonymous surveys in order to perform theoretical and practical distributed sessions around common exercises, both teachers and students.

In this paper we have exposed how to teach Software Engineering topics through Computer Support Collaborative Work principles, to students from several countries in Latin American. Nowadays, we are implementing a Latin American Co-laboratory on eXperimental Software Engineering Research (LACXSER), that count on the support of several organizations, such as LACCIR, Colciencias (Colombia) and Education Ministry from Argentina. The next steps are to continue learning from these experiences and to increase the number of universities participating in this initiative.

ACKNOWLEDGMENTS

This work is partially funded by the project entitled "*Fortalecimiento de la Red de Investigación Aplicada en Ingeniería de Software Experimental*", in *Proyectos de Fortalecimiento a Redes Interuniversitarias* II call of the Ministry of Education of Argentina. Authors thank to CINTEL- Colciencias (Colombia), for partial funding of this work by the research project entitled "*Red Latinoamericana de Investigación Aplicada en Ingeniería de Software Experimental*", grant IF-007-09 (Call Colciencias 487 - RENATA 2009), and also LACCIR Grant R1209LAC003.

REFERENCES

[1] Alavi, M., Computer-mediated collaborative learning: An empirical evaluation. MIS Quarterly, 18(2), 150-174. 1994.

[2] Collazos, C., Ochoa, S., Mendoza, J., Improving the schemes of evaluation through collaboration processes (In Spanish). Revista Educación y Educadores 10(1). 2007.

[3] Cuneo, C., Mariño, M., Collaborative Environment in the Education of Expert Systems (In Spanish). Comunicaciones Científicas y Tecnológicas. 2005.

[4] Falchikov, N., Learning together: Peer tutoring in higher education. London: Routledge Falmer. 2001.

[5] Granollers, T., MPIu+a. Proccess Model of Usability and Accessibility Engineering (In spanish). Departament de Llenguatges i Sistemes Informàtics, Universidad de Lleida. 2004.

[6] Dick, B. Simmons, Software Engineering Education in the New Millennium. Proceedings of the 30th Annual International Computer Software and Applications Conference (COMPSAC'06). IEEE Press.2006.

[7] Kagan, S., Cooperative Learning Resources for Teachers. San Juan Capistrano, CA: Resources for Teachers. 1989.

[8] http://www.renata.edu.co/index.php/buenas-practicas/36-buenas-practicas/155-laboratorio-de-ingenieria-de-sistemas-y-computacion-de-uniquindio-pionero-en-uso-de-renata.html

[9] Manjarres, A., Arias, M., Gaudioso, E., Transverse Competencies in software engineering teaching (In Spanish). VI Jornadas de Redes de Investigación en Docencia Universitaria, Universidad de Alicante. 2008.

[10] Mariño, S., Virtual Environment Design of Teaching-Learning for a Artificial Intelligence Course (In Spanish). Quaderns Digitals. Revista electrónica No.3. 2008.

[11] Mesa, J., Alvarez, J., Villanueva, J., Cos, F., Update of Teaching-Learning Methods in Courses of Engineering Project Management (In Spanish). Formación Universitaria, 1(4), pp.23-28. 2008.

[12] Orvis, K., Lassiter, A., Computer-Supported Collaborative Learning: Best Practices and Principles for Instructors. Information Science Publishing, Hershey, New York. 2007.

[13] Redondo, M., Mendes, A., Ortega, M., Collaborative Planning of Design for Programming Learning (in Spanish) , Taller Internacional de Software Educativo, Chile. 2001.

[14] Sangwan, R., Bass, M., Mullick, N., Paulish, D., Kazmeier, J., Global Software Development Handbook. Auerbach Publications. ISBN 0-8493-9384-1. 2007.

[15] Collazos, C., Guerrero, L., Pino, J., Ochoa, S., Improving the Use of Strategies in Computer-Supported Collaborative Processes. Proceedings of the 9th International Workshop on Groupware. Grenoble, France. Springer Verlag LNCS, 2806, pp.356-370. 2003.

[16] Lund, M., Zapata, S., Aballay, L., Herrera, M., Torres, E., Collazos, C., Giraldo, F., Ochoa, S., Evaluation of collaborative instructional process to software engineering in distributed learning environment (In Spanish). Revista Avances en Sistemas e Informática Vol 6-No.2. 2009.

[17] Hawthorne, M., Dewayne, E. Software Engineering Education in the Era of Outsourcing, Distributed Development, and Open Source Software: Challenges and Opportunities. Proc. of the 27th Int. Conf. on Software Engineering (ICSE). St. Louis, USA. Pages: 643 - 644. 2005.

[18] Bareiša, E., Karčiauskas, E., Mačikėnas, E., Motiejūnas, K., Research and Development of Teaching Software Engineering Processes. Proc. of the International Conference on Computer Systems and Technologies. Bulgaria. 2007.

[19] Carver, J., Jaccheri, L., Morasca, S., Shull, F., Issues in Using Students in Empirical Studies in Software Engineering Education. In: Ninth International Software Metrics Symposium (METRICS'03). 2003.

[20] Bagert, D., Hilburn, T., Hislop, G., Lutz, M., McCracken, M., Mengel, S., Technical Report CMU/SEI-99-TR-032 ESC-TR-99-002: Guidelines for Software Engineering Education Version 1.0. 1999.