

# COMPETENCIES DESIRABLE FOR A REQUIREMENTS ELICITATION SPECIALIST IN A GLOBAL SOFTWARE DEVELOPMENT

Miguel Romero

*University of Bío-Bío, Department of Computer Science and Information Technologies  
Avenida Andrés Bello s/n 3780000 Chillán, Chile*

Aurora Vizcaíno, Mario Piattini

*University of Castilla-La Mancha, Alarcos Research Group - Institute of Information Technologies & Systems  
Department of Information Technologies & Systems - Escuela Superior de Informática  
Paseo de la Universidad 4, 13071 Ciudad Real, Spain*

**Keywords:** Requirements Elicitation training, Global Software Development, Educational Environment.

**Abstract:** The global software development poses several challenges in software engineering, particularly in the elicitation stage, owing to the problems of communication and coordination which are caused when teams are geographically distributed. For a successful requirements elicitation in a global development environment, it is necessary to rely on professionals who are capable of confronting the challenges that arise in this environments such as: cultural differences, distributed communication and coordination. In order to develop in software engineers the skills suitable to face these challenges it is first necessary to discover which competencies or skills they should have or develop. In this work we describe an analysis carried out with this goal, therefore we propose a list of competencies desirable by a requirements elicitation specialist, which have been obtained from a review of the related literature. We also comment on certain useful strategies in the teaching of these competencies and propose the usage of a simulation environment for their development.

## 1 INTRODUCTION

Global Software Development (GSD) is an emerging paradigm in the software industry. GSD is characterized by stakeholders who are geographically distributed around the World. The principal advantages of GSD are that: it allows the exploitation of a 24-hour work day, it decrease costs, it capitalizes on global resource pools, and it is geographically closer to the end consumer (Cheng & Atlee, 2007). However, GSD also has disadvantages, principally with regards to coordination and communication between virtual teams (James D. Herbsleb, 2007). Nevertheless, GSD continues to grow. In fact, one finding of the report of ACM's Job Migration Task Force is that: "Globalization of, and offshoring within, the software industry are deeply connected and both will continue to grow.

Key enablers of this growth are information technology itself, the evolution of work and business processes, education and national policies" (from p.9, Aspray, Mayadas, & Vardi, 2006).

With the arrival of this paradigm, there is an increasing gap between what is taught in universities and what the software industry requires, owing to the fact that GSD demands a new set of competencies (Patterson, 2006).

One of the areas of software engineering which is most affected by the difficulties that emerge from GSD is that of requirements elicitation, because it is fundamentally a communication process between a requirements specialist and the customer's stakeholder (SWEBOK, 2004).

Given the importance of the elicitation stage, the main objective of this work is to identify the competencies that are necessary to develop said

stage in GSD. We also propose a simulation tool which assists in the training of these competencies.

In the following section we present the challenges of global software development which are reported in literature. Section 3 shows the competencies which were identified and Section 4 presents the various strategies that can be used to teach said competencies. Finally, Section 5 outlines our conclusions and our future work.

## 2 ISSUES IN GSD

The GSD paradigm presents several issues that affect all the phases of software development. The following list corresponds with the main challenges in GSD:

- *Cultural Difference.* Cultural differences can affect a GSD project in different ways including communication and coordination effectiveness, group decision-making and team performance (James D. Herbsleb & Moitra, 2001), (D. Damian, 2007), (D. E. Damian & Zowghi, 2002), (Raffo & Setamanit, 2005), (Setamanit, Wakeland, & Raffo, 2006).
- *Inadequate Communication.* In global environments communication is challenged by many factors. Cultural, along with language differences, distance, and time differences make it difficult for people to interact, so stakeholders must make a special effort to communicate effectively (Bellur, 2006), (D. Damian, Hadwin, & Al-Ani, 2006), (James D. Herbsleb & Moitra, 2001).
- *Time Difference.* Distribution over many geographically distanced sites introduces time difference as another factor. This can be seen as an advantage, because it presents the opportunity of working on an around-the-clock schedule, but it can also be a disadvantage when the tasks need an intensive collaboration between people in distanced sites and synchronous communication is difficult to implement or is not even possible during the normal work timetable (D. E. Damian & Zowghi, 2002), (Raffo & Setamanit, 2005), (Setamanit et al., 2006).
- *Knowledge Management.* People working on GSD projects need to share a lot of information about requirements, which comes from several sources at distanced sites. Without effective information and knowledge sharing mechanisms it is not possible to

exploit GSD's benefits (D. E. Damian & Zowghi, 2002), (James D. Herbsleb & Moitra, 2001), (Huang & Trauth, 2007).

- *Language Difference.* Language differences may be a source of misunderstanding, especially when the common language between stakeholders is not their native language (Huang & Trauth, 2007), (Raffo & Setamanit, 2005), (Setamanit et al., 2006).
- *Trust.* Maintaining trust relationships is especially difficult in GSD environments because of the lack of informal and spontaneous communication (Babar, Verner, & Nguyen, 2007), (Bhat, Gupta, & Murthy, 2006), (Nguyen, Babar, & Verner, 2006).

These problems greatly affect communication between stakeholders. The requirements elicitation stage is fundamentally a communication process and is therefore greatly affected by GSD issues.

In order to train professionals in the process, who are capable of accomplishing top-quality requirement elicitation and of confronting the difficulties of GSD environments, teaching must be adjusted in its different dimensions: contents, learning tools, learning techniques, assessment strategies, learning outcomes and professional competencies. In this context, defining professional competencies for software engineers to work in GSD environments is fundamental, since it will allow us to have clear ideas about the professionals the industry needs, so as to define the updated contents for software engineering courses and learning tools which will help students to be more prepared to work in GSD.

The following section proposes a set of competencies, both generic and specific, for a requirements specialist working on a GSD Project.

## 3 COMPETENCIES FOR A GLOBAL REQUIREMENTS ELICITATION

Before describing the competencies for requirements elicitation in global environments, we believe it appropriate to explain the meaning of the term 'competencies'.

### 3.1 Definition of Competencies

In literature, the concepts of knowledge skills and competency are used to describe the attributes that a person should have if s/he is to play a part in a

profession. In this sense, as the ‘Tuning project’ points out: “Several terms: capacity, attribute, ability, skill, competency are used in an often interchangeable, and to some degree overlapping meaning. They all relate to the person and to what s/he is capable of achieving. But they also have more specific meanings. Ability, from the Latin ‘habilis’ meaning ‘able to hold, carry or handle easily’, leads to the word ‘habilitas which can be translated as ‘aptitude, ability, fitness or skill’... a competency or a set of competencies means that a person puts into play a certain capacity or skill and performs a task, in which s/he is able to demonstrate that s/he can do so in a way that allows evaluation of the level of achievement.” (from p.20, Tuning, 2007).

The concept of competencies is closest to the way in which companies evaluate professionals. Moreover, the concept is used in the process of adapting European universities to European Higher Education Area. It is for these reasons that we have decided to use this concept.

### 3.2 Competencies Identified

In order to define a list of competencies which are necessary for the requirements elicitation process, we have carried out the following process:

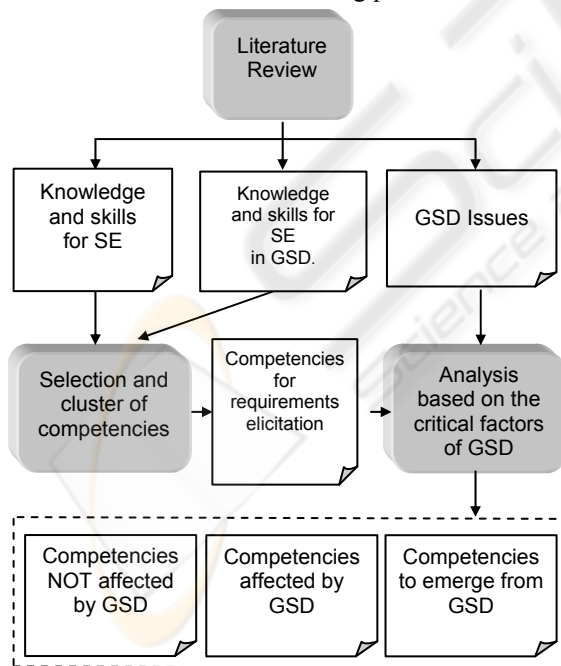


Figure 1: Process used to obtain the competencies.

We have first carried out a review of literature in search of papers reporting skills, knowledge or

competencies for software engineers in general, and also for GSD environments.

Based on these studies, we have developed an initial set of 189 elements (knowledge, skills or competencies) for software engineers. From this list the elements related to requirements elicitation were selected and the list was then reduced to 90 elements removing others such as: behavioral modeling (e.g. structured analysis, state diagrams, etc.), process quality and improvement, which were not relevant for our work. We next removed the elements which were repeated and those which were contained in elements other, thus reducing the list to 70 elements. A list of competencies for requirements elicitation is defined from this list of 70 elements. From the list of competencies defined, and by considering the critical factors for GSD reported in literature (see Section 2), we have examined the competencies one by one in order to discover what the impact of each critical factor upon the competencies is.

As a result of this analysis, we have obtained a list of competencies for the requirements elicitation process which we have placed in three groups:

1. The competencies that emerge due to the Global development of the software, and which are therefore not required in local software development environments. These competencies are listed in Table 1.

Table 1: Competencies which emerge from Global Software Development.

Generic Competencies
Knowledge of a second language (principally English Language skills) (Tuning, 2007), (Huang & Trauth, 2007), (Ita Richardson, Moore, Paulish, Casey, & Zage, 2007). The need to communicate with people who speak another language makes the knowledge of a second language (particularly English) vital.
Ability to work in an international context (operating in a virtual team environment, Virtual team skills) (Tuning, 2007), (Gorgone et al., 2002), (Vasudevan, 2006), (D. Damian et al., 2006).
Appreciation of diversity and multiculturalism (Tuning, 2007).
Ability to deal with multicultural environments (IEEE & ACM, 2004), (Minor & Armarego, 2005), (Ita Richardson et al., 2007), (Gorgone et al., 2002).
Understanding of cultures and customs of other countries (Tuning, 2007).
Understanding of Global awareness (Adya, 2006). Achieving this competency is particularly difficult owing to problems in knowledge management.

- The competencies required in co-located environments that are affected by the critical factors of GSD, which therefore implies that the manner in which to teach them needs to be adapted to consider the impact of GSD. These competencies are listed in Tables 2 (the generic ones) and 3 (the specific ones).

Table 2: Generic competencies for requirements elicitation which are affected by GSD.

Generic competencies
Computer mediated communication skills (Ita Richardson et al., 2007), (D. Damian et al., 2006). The geographic distance between the stakeholders means that this competency is more important in GSD, along with the difficulties of time difference.
Use of Communication protocols (Ita Richardson et al., 2007), (Huang & Trauth, 2007). Communication protocols change between cultures so, for example, ways of greeting others (i.e. with a kiss) may be a motive of conflict.
Communication skills, (timely responses, speed, recognizing the semantic gap) (Adya, 2006), (Ahamed, 2006), (Ita Richardson et al., 2007), (Minor & Armarego, 2005), (Callele & Makaroff, 2006). In the GSD environment greater effort is needed to achieve effective communication as this is affected by cultural problems, distance, language, and time differences.
Ability to Resolve Conflicts (Aken & Michalisin, 2007). This competency is affected by cultural differences and communication problems in GSD.
Critical and self-critical abilities (Tuning, 2007), (Aken & Michalisin, 2007), (Callele & Makaroff, 2006). Criticizing someone else's work when that person is from a different culture is much more difficult than when that person is from the same culture.
Ability to deal with uncertainty and ambiguity (in local and remote teams) (IEEE & ACM, 2004), (D. Damian et al., 2006). In remote teams, ambiguity is far greater owing to the knowledge management problems present in GSD.
Ability to interact with stakeholders (often not from the same culture) (IEEE & ACM, 2004), (Ghezzi & Mandrioli, 2005), (Aken & Michalisin, 2007). Cultural difference, inadequate communication, language difference and poor trust affect interaction with stakeholders.
Team and group communication skills (both oral and written, email, etc.) (IEEE & ACM, 2004). The development of this competency is particularly affected by GSD issues.
Teamwork skills (Dynamics of working in teams/groups) (Tuning, 2007), (IEEE & ACM, 2004), (Minor & Armarego, 2005), (Ahamed, 2006), (Aken & Michalisin, 2007).

Table 3: Specific competencies for requirements elicitation which are affected by GSD.

Specific Competencies
Knowledge of Advanced elicitation techniques (e.g. ethnographic, knowledge elicitation, etc.) (IEEE & ACM, 2004). Elicitation techniques were developed for local environments, and it is therefore necessary to review their use in global environments
Application of Elicitation Techniques (e.g. interviews, questionnaires/surveys, prototypes, etc.) (IEEE & ACM, 2004), (SWEBOK, 2004).
Ability to Identify real requirements (Young, 2006). Communication problems and cultural differences affect this competency.
Requirements elicitation skills (Minor & Armarego, 2005), (Young, 2006), (Callele & Makaroff, 2006). This competency is affected by all the issues in GSD.
Comprehension of requirements change control and change notification (Young, 2006). This competency is particularly affected by knowledge management problems.
Understanding of Requirements Tracing (SWEBOK, 2004). As with the aforementioned competency, this is also affected by problems of knowledge management.
Comprehension of Success Factors of GSD (Minor & Armarego, 2005).

- The competencies that are not affected by the critical factors of GSD and what should be taught on courses independently if the requirement elicitation process will be carried out in a collocated or distributed environment. For example: application of criteria for good requirement, detection of elicitation sources, classification requirements skills, elementary computing skills, working under pressure, and the capacity to learn. This group of competencies is outside the scope of this work and is therefore not presented.

### 3.3 Main Sources of Bibliographic Competencies Identified

The main sources of this study are the following: SWEBOK (SWEBOK, 2004), whose philosophy defines the body of knowledge generally accepted for software engineers (the completion of a 4 year degree course), It does not take into account aspects of GSD in the definition of knowledge. It is, however, a good starting point.

SE2004 (IEEE & ACM, 2004), presents the body of knowledge for software engineering that is required to be taught in a degree program. SE2004 delivers relevant information on how to teach these

skills, indicating which of them are essential and which are desirable, along with suggesting means to structure the curriculum. As with the SWEBOK, this has a broad consensus, but neither refers to aspects of GSD. While both studies are developed by ACM and IEEE, the body of knowledge presented in SWEBOK differs in part to that presented in SE2004. For example: "Layers / levels of requirements" appears in SE2004 but not in SWEBOK while "emergent properties" appears in SWEBOK but not in SE2004.

The Tuning project (Tuning, 2007) presents a study of the generic competencies it also indicates which are the most and least important, both for the academic world and for industry. Unfortunately, this study does not include software engineering. This means that the hierarchy of competencies contained in this project is not valid in its entirety for our field. For example, the competencies needed to understand the cultures and customs of other countries, appreciate diversity and multiculturalism, and the ability to work in an international context are among those least valued by the employers and graduates surveyed in the Tuning project (Tuning, 2007). This is because the groups surveyed are from fields of knowledge which have not been so highly affected by globalization as has computing. It would, therefore, be useful to replicate the study in our field in order to properly classify the generic skills according to our needs.

The empirical study of Damian, "outlines a set of emerging areas of competencies that a curriculum needs to emphasize when training students for GSD, in addition to the basic skills of an SE" (from p. 686, D. Damian et al., 2006). However, certain fundamental skills of SE (see Table 2) should be studied in greater depth as they are affected by the critical factors of GSD and it is not therefore possible to apply them directly to GSD.

#### **4 TRAINING THE COMPETENCIES FOR GLOBAL REQUIREMENT ELICITATION**

The main challenge in the teaching of requirements elicitation is to succeed in giving students the chance to learn from concrete experiences that are closer to real work. To do so, it is necessary to develop teaching strategies for active and collaborative learning (Rosca, 2000), in which students learn by

doing instead of just listening to an expert talking about his/her experiences.

The techniques that are commonly used with such an aim are project-based learning (real or otherwise) (Vaughn & Carver, 2006), where students participate in teams, in order to solve a problem; role playing games (Barrett, 1997; Jaccheri & Sindre, 2007; Sindre, 2005), in which students play a role (software engineer, client, user, analyst, etc.) in a simulated requirement elicitation scenario.

Various strategies have been defined to confront the challenge of teaching and training software engineers to work on GSD projects. These are: curricular changes (Bellur, 2006; Cross-II, 2005; Minor & Armarego, 2005; Ramnath, 2006; Vasudevan, 2006), postgraduate specialization (Lago et al., 2007); a closer interaction between industry and the academic world (Lee et al., 2005); and joint software projects between universities from different countries (Ahamed, 2006; I Richardson, Milewski, Mullick, & Keil, 2006).

However, there are certain problems in putting these strategies into practice, such as the difficulty in finding companies who are willing to invest time and resources in joint education projects with universities, or the lack of experience of students which may be a very high risk factor for real projects.

The fields of both medicine and aviation use simulators to minimize the risks inherent in the training of professionals. We therefore propose the development of a simulation environment which will allow students to acquire some of the competencies mentioned in the previous section.

This simulator may be an initial step towards students' participation in real projects developed between universities and the GSD industry. In fact, the experience acquired with the simulator would diminish the risk of non-qualified people being involved in real projects. Furthermore, the simulator provides a virtual industrial partner for universities that do not have one.

We propose a simulator of the requirements elicitation process in the global context in which the student (taking on the role of an RE engineer) interacts with various stakeholders which will be virtual agents and/or real humans. The simulator will allow the professor to create new lessons, indicating the description of the scene, the virtual agents to be used, personality and culture.

The interaction will be natural through the use of the main tools of electronic communication used for requirements elicitation: Instant Messaging and Chat, E-Mail, Telephone and Video Conferencing.

Initially, the students must enter their data with the aim of tracking the learning process. Then the system must show the different lessons or units that it has developed, showing their results. In addition, the system must permit a review of the history of conversations with each of the stakeholders. Another capability of the system will be to show the lessons that it has not developed, allowing the student to select any of them.

When students perform a lesson selection, the system must submit the context of the problem in which the elicitation is developed and show the participant stakeholders and their roles.

Through interviews with the various stakeholders (who will be of different nationalities) the students should prepare a list of requirements, both functional and non-functional, which should be sent to the system for its validation at the end of the simulation with the purpose of measuring the quality of the work done by the student. The system should provide an interface for keeping a list of the student's requirements

The simulator will validate the student's work by means of a questionnaire in which it will present various requirements (both functional and non-functional) and the student will have to indicate whether or not they correspond with what the users need. The requirements document will be checked to detect faults such as: ambiguous requirements, non-existent requirements, unspecified requirements, etc. Besides this evaluation, the system will also record the questions that the student has formulated in an inadequate way with regard to cultural differences and the protocol of communication (manner of greeting and taking one's leave, degree of formality informality etc.).

In short, the simulator will teach the 17 competences listed below (see Table 4) by means of the following features (Column F in Table 4):

1. Interaction with virtual agents of different nationalities.
2. Interviews with the stakeholders.
3. The lessons are focused on the specific problems of RE in GSD to allow the student to confront common difficult situations.
4. The elaboration of the requirements document.
5. The validation of the requirements document.

## 5 CONCLUSIONS

Global software development is a current tendency in industry which is motivated by globalization and offshoring. This tendency is increasing, which implies that it is necessary to make the appropriate

Table 4: Competences to be developed with simulator

Competencies to develop with Simulator		F
Generic Competencies	1 Computer mediated Communications (D. Damian et al., 2006; Ita Richardson et al., 2007)	1, 2
	2 Communication protocols (Huang & Trauth, 2007; Ita Richardson et al., 2007)	1, 2
	3 Virtual team skills (D. Damian et al., 2006; Vasudevan, 2006)	1, 4
	4 Ability to work in an international context (Tuning, 2007)	1, 2, 3
	5 Appreciation of diversity and multiculturality (IEEE & ACM, 2004; Tuning, 2007)	1, 2, 3
	7 Living with ambiguity/uncertainty in Remote Teams (D. Damian et al., 2006)	1, 2, 3
	8 Ability to learn quickly about a domain or technology in order to begin project planning (I Richardson et al., 2006)	2, 4
	9 Capacity to adapt to new situations (Tuning, 2007)	3
	10 Understanding of cultures and customs of other countries (Ita Richardson et al., 2007; Tuning, 2007)	1, 2, 3, 4
	Specific Competencies	11 Comprehension of GSD Critical Factors
12 Detection of Elicitation Sources (e.g. stakeholders, domain experts, operational and organization environments, etc.) (IEEE & ACM, 2004)		2, 3, 4
13 Comprehension of Software requirements specification (IEEE & ACM, 2004; Young, 2006)		4, 5
14 Knowledge about analyzing quality (non-functional) requirements (e.g. safety, security, usability, performance, root cause analysis, etc.) (IEEE & ACM, 2004)		4, 5
15 Capacity to manage changing requirements (IEEE & ACM, 2004; Young, 2006)		4, 5
16 Elicitation of real requirements based on stakeholder's need using an Interview Technique and computer mediated communications		2, 3, 4, 5
17 Representation of functional and non-functional requirements for different type of systems		3, 4, 5

adjustments to the preparation of software engineers and other specialists to ensure their capacity to work in this new global environment. This work focuses upon requirements elicitation, on the one hand because it is one of the stages that is most affected by the challenges of GSD, and on the other because it is one of the most important stages in software development in general.

This work presents the competencies desirable by a requirements elicitation specialist and proposes a simulator which will permit the development of a subset of these competencies.

After our analysis and proposal of the competencies our future work consists of the development of this simulator, and its use in university teaching and in the training of professionals.

The list of competencies, moreover, can be used to define a course and an educational guide for the teaching of requirements elicitation in GSD environments. Furthermore, a guide towards the recruitment of requirements specialists in GSD environments will also be developed.

## ACKNOWLEDGEMENTS

This work is partially supported by the MELISA project (PAC08-0142-3315), Junta de Comunidades de Castilla-La Mancha, Consejería de Educación y Ciencia, in Spain; ESFINGE project (TIN2006-15175-C05-05) Ministerio de Educación y Ciencia (Dirección General de Investigación)/Fondos Europeos de Desarrollo Regional (FEDER) in Spain; the CompetiSoft project (506AC0287, CYTED program).

## REFERENCES

- Adya, M. P. (2006). *Imparting Global Software Development Experience via an IT Project Management Course: Critical Success Factors*. Paper presented at the 30th Annual International Computer Software and Applications Conference (COMPSAC'06).
- Ahamed, S. I. (2006). *Model for Global Software Engineering Project Life Cycle and How to Use it in Classroom for Preparing Our Students for the Globalization*. Paper presented at the 30th Annual International Computer Software and Applications Conference (COMPSAC'06).
- Aken, A., & Michalisin, M. D. (2007). *The impact of the Skills Gap on the Recruitment of MIS Graduates*. Paper presented at the SIGMIS-CPR'07, St. Luis, Missouri, USA.
- Aspray, W., Mayadas, F., & Vardi, M. (2006). *Globalization and Offshoring of Software* (No. ACM 0001-0782/06/0200): Association for Computing Machinery, Job Migration Task Force (ACM).
- babar, M. A., Verner, J. M., & Nguyen, P. T. (2007). *Establishing and maintaining trust in software outsourcing relationships: An empirical investigation*. *The Journal of Systems and Software*, 80(9), 1438-1449.
- Barrett, M. L. (1997). *Simulating requirements gathering*. *SIGCSE Bulletin*, 29(1), 310-314.
- Bellur, U. (2006). *An Academic Perspective on Globalization in the Software Industry*. Paper presented at the 30th Annual International Computer Software and Applications Conference (COMPSAC'06).
- Bhat, J. M., Gupta, M., & Murthy, S. N. (2006). *Overcoming Requirements Engineering Challenges: Lessons from Offshore Outsourcing*. *IEEE Software*, 23(5), 38-44.
- Callele, d., & Makaroff, D. (2006). *Teaching Requirements Engineering to an Unsuspecting Audience*. Paper presented at the 37th SIGCSE technical symposium on Computer science education, Houston, Texas, USA.
- Cheng, B. H. C., & Atlee, J. M. (2007). *Research Directions in Requirements Engineering*. Paper presented at the Future of Software Engineering, 2007 (FOSE '07).
- Cross-II, J. H. (2005). *21st-Century Global Software Development and Education Position Statement*. Paper presented at the 29th Annual International Computer Software and Applications Conference (COMPSAC'05).
- Damian, D. (2007). *Stakeholders in Global Requirements Engineering: Lessons Learned from Practice*. *IEEE Software*, 24(2), 21-27.
- Damian, D., Hadwin, A., & Al-Ani, B. (2006). *Instructional design and assessment strategies for teaching global software development: a framework*. Paper presented at the International Conference on Software Engineering (ICSE'06).
- Damian, D. E., & Zowghi, D. (2002). *The Impact of Stakeholders' Geographical Distribution on Managing Requirements in a Multi-Site Organization*. Paper presented at the IEEE Joint International Conference on Requirements Engineering (RE'02), Los Alamitos, CA, USA.
- Ghezzi, C., & Mandrioli, D. (2005). *The challenges of software engineering education*. Paper presented at the 27th International Conference on Software Engineering (ICSE'05).
- Gorgone, J. T., Davis, G. B., Valacich, J. S., Topi, H., Feinstein, D. L., & Herbert E. Longenecker, J. (2002). *IS 2002. Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems*.: Association for Computing Machinery (ACM), Association for Information Systems (AIS),

- Association of Information Technology, Professionals (AITP).
- Herbsleb, J. D. (2007, Mayo). *Global Software Engineering: The Future of Socio-technical Coordination*. Paper presented at the Future of Software Engineering(FOSE'07) at ICSE'07, Minneapolis.
- Herbsleb, J. D., & Moitra, D. (2001). Guest Editors' Introduction: Global Software Development. *IEEE Software*, 18(2), 16-20.
- Huang, H., & Trauth, E. (2007). *Cultural influences and globally distributed information systems development: experiences from Chinese IT professionals*. Paper presented at the SIGMIS-CPR '07.
- IEEE, & ACM. (2004). *Software Engineering 2004, Curriculum Guidelines for Undergraduate Degree Programs In Software engineering*: IEEE computer Society Press and ACM Press.
- Jaccheri, L., & Sindre, G. (2007). *Software Engineering Students meet Interdisciplinary Project work and Art*. Paper presented at the 11 th International Conference Information visualization (IV'07).
- Lago, P., Muccini, H., Beus-Dukic, L., Crnkovic, I., Punnekkat, S., & Vliet, H. V. (2007). *Towards a European Master Programme on Global Software Engineering*. Paper presented at the 20th Conference on Software Engineering Education & Training (CSEET'07).
- Lee, D., Jung, K., Yi, K., Cho, Y., Han, Y., & Kang, D. (2005). *Development of partnership between industry and university via customized field-oriented curriculum*. Paper presented at the 2005 IEEE International Conference on Microelectronic Systems Education (MSE'05).
- Minor, O., & Armarego, J. (2005). Requirements Engineering: A Close Look At Industry Needs And Model Curricula. *Australian Journal of Information Systems (AJIS)*, 13(1).
- Nguyen, P., Babar, M., & Verner, J. (2006). *Critical factors in establishing and maintaining trust in software outsourcing relationships*. Paper presented at the International Conference on Software Engineering (ICSE'06), Shanghai, China.
- Patterson, D. A. (2006). Offshoring: finally facts vs. folklore. *Commun. ACM*, 49(2), 41-42.
- Raffo, D., & Setamanit, S. (2005). *A Simulation Model for Global Software Development Project*. Paper presented at the The International Workshop on Software Process Simulation and Modeling, St. Louis, MO.
- Ramnath, R. (2006). *Global Software Development for the Enterprise*. Paper presented at the 30th Annual International Computer Software and Applications Conference (COMPSAC'06).
- Richardson, I., Milewski, A., Mullick, N., & Keil, P. (2006). *Distributed development: an education perspective on the global studio project*. Paper presented at the International Conference on Software Engineering (ICSE'06).
- Richardson, I., Moore, S., Paulish, D., Casey, V., & Zage, D. (2007). *Globalizing Software Development in the Local Classroom*. Paper presented at the 20th Conference on Software Engineering Education & Training (CSEET'07).
- Rosca, D. (2000). *An Active/ Collaborative Approach in Teaching Requirements Engineering*. Paper presented at the 30th Annual Frontiers in Education - Vol 1 (FIE'2000).
- Setamanit, S., Wakeland, W., & Raffo, D. (2006). *Planning and improving global software development process using simulation*. Paper presented at the International Workshop on Global Software Development for the Practitioner (GSD'06), Shanghai, China.
- Sindre, G. (2005). *Teaching Oral Communication Techniques in RE by Student-Student Role Play: Initial Experiences*. Paper presented at the 18th Conference on Software Engineering Education & Training (cseet'05).
- SWEBOK. (2004). *Guide to the Software Engineering Body of Knowledge*.
- Tuning. (2007). *Tuning General Brochure. English version*. Retrieved 17/12/2007, 2007, from [http://www.tuning.unideusto.org/tuningeu/index.php?option=com\\_docman&Itemid=59&task=view\\_category&catid=19&order=dmdate\\_published&ascdesc=DESC](http://www.tuning.unideusto.org/tuningeu/index.php?option=com_docman&Itemid=59&task=view_category&catid=19&order=dmdate_published&ascdesc=DESC)
- Vasudevan, V. (2006). *Global Software Entrepreneurship*. Paper presented at the 30th Annual International Computer Software and Applications Conference (COMPSAC'06).
- Vaughn, R. B. J., & Carver, J. (2006). *Position Paper: The Importance of Experience with Industry in Software Engineering Education*. Paper presented at the 19th Conference on Software Engineering Education and training Workshops (CSEETW' 06).
- Young, R. (2006). Twelve Requirements Basics for Project Success. *CROSTALK The Journal of Defense Software Engineering*, 2006(December).