A framework for interaction training in Global Software Development

PhD Thesis

PhD Student:

Miguel Jiménez Monasor

Supervisors:

Dra. Aurora Vizcaíno Barceló

Dr. Mario Piattini Velthuis

Dra. Sarah Beecham

University of Castilla-La Mancha

A framework for interaction training in Global Software Development

PhD Student:

Miguel Jiménez Monasor

Supervisors:

Dra. Aurora Vizcaíno Barceló

Dr. Mario Piattini Velthuis

Dra. Sarah Beecham

Ciudad Real, Spain, 2014

Dedicado a mi madre

Acknowledgements

I would like to express my sincere gratitude to all those (named or otherwise) who have helped me to complete this part of the road of life.

First of all, I would like to acknowledge all the time and effort of my advisors, Aurora Vizcaíno, Mario Piattini and Sarah Beecham. Without their advice, confidence and academic and personal support, this thesis would not have been possible.

I would also like to thank the reviewers, Dr. Tony Clear and Dr. Juho Maekioe, for their helpful comments and suggestions.

I would like to extend my thanks to the members of the Alarcos Research Group who helped me, and especially Pedro José Garrido for his support. I am also grateful to Lero, The Irish Software Engineering Research Centre, and all its members for providing me with the resources that were necessary to finish this thesis. My greatest thanks go to all the colleagues and friends that I met in Limerick, for the shared knowledge and the good moments that we spent together. I really appreciate what I have experienced and learnt with them during this experience. I would also like to extend my gratitude to the management and administration personnel from both research groups for their agility and assistance.

Finally, I would like to thank my colleagues and friends in Albacete who helped me to some extent during the initial research stage, and also my family for their support.

Contents

List of Figures

List of Tables

Abstract

Global Software Development (GSD) is an emerging paradigm in which development teams are geographically distributed throughout different countries whilst working on the same projects. Lower costs, greater access to a skilled workforce, the leveraging of time-zone effectiveness and a closer proximity to the market are some of the proven advantages that have contributed to this recent expansion. However, GSD also entails certain drawbacks caused mainly by distance and socio-cultural differences. Traditional face-to-face meetings are no longer common, thus making communication, coordination and collaboration more complex. Moreover, interactions frequently involve members who use non-native languages and whose customs and behavior may be misinterpreted.

Companies consequently require their workforce to have a balanced set of socio-technical skills more than ever before. Students and practitioners therefore need to acquire not only new technical skills, but also the ability to collaborate and interact effectively in a variety of geographic, cultural and linguistic settings. This kind of training is not currently common and companies often complain about their new recruits' lack of preparation in GSD related skills. However, companies do not always have the resources to provide these kinds of training programs, as the teaching of this kind of skills requires appropriate tools, methods and course materials. Moreover, training these skills is not easy at universities since it necessitates providing trainees with real experiences that it may be difficult to reproduce. Some universities organize joint courses through collaboration with distant universities so that students can benefit from a wide set of knowledge and experiences, but the coordination problems of these approaches make them hard to apply.

One way in which to avoid the problems outlined above, i.e., a lack of resources, coordination complexities, and poor levels of realism, is by using simulation, which has been used in various areas of Software Engineering training such as project management. However, although simulation has the potential to enable independent and controlled training, no significant proposals exist in the field of GSD training. The research question that guides this work is therefore: "How can we model a GSD educational framework that replicates real-world settings?"

This thesis presents a framework (VENTURE) for interactive GSD training which is based on simulation. The framework simulates real world interactions that occur in GSD which are appropriate for use in both industrial and academic settings. In order to simulate GSD settings, trainees take on a specific role and interact with Virtual Agents, characterized by a specific culture. A special kind of Virtual Agent, called a Virtual Guide, directs the trainees and provides feedback when they interact in an inappropriate manner.

These training scenarios take place by means of a *chat simulator* for synchronous communication training and through an *email simulator* for asynchronous communication training. The feedback provided by the Virtual Guide is automatically used to assess the trainees' performance in real time following a defined assessment method. The courses are provided via an *e-learning environment* that allows students to access the training sessions at any time. Moreover, a *scenario designer* has been implemented to support the design of new simulated interactions. New simulations will build on predefined GSD problems, including cultural and linguistic rules that represent typical interaction challenges in real settings.

In order to design GSD scenarios, realistic low-level GSD problems that arise during the day-to-day work are required. However, accessing these problems or scenarios is difficult, mainly because companies are not always willing to share their knowledge and experience. In order to tackle this challenge, the framework additionally considers a *Web based knowledgebase of GSD patterns and scenarios*. The GSD community can, through this open social network, share their experiences, problems and scenarios in an

organized manner. Simulations may therefore be designed by applying the data collected in this collaborative knowledgebase.

The VENTURE framework was created by following a mixed methodology that combines both qualitative and quantitative methods. An initial systematic literature review on GSD served to focus the topic on the field of GSD education, after which a second systematic literature review was conducted in the field of GSD education. Having studied the problems of the existing educational methods, a prototype of the framework was built and evaluated by experts, leading to a first version of the framework.

The principal components of VENTURE (the chat simulator and the scenario designer) were then evaluated by a group of experts. These experts in GSD and e-learning examined the design and executed a brief training simulation in order to analyze the tool's potential to provide GSD training, along with the tool's usability. The results of the survey, which were analyzed by means of a thematic analysis procedure, were mainly positive. Both evaluations served to confirm that VENTURE has the potential to fill the gap in GSD virtual team training. Moreover, indications on where improvements could be made served to develop a new version of the framework.

A Field Study of the new version of the framework was conducted in order to evaluate whether learning was achieved after participation on a real GSD course based on two training simulations. This evaluation involved potential users of the framework (students, trainers, researchers) interacting with VENTURE and completing two training scenarios. A rigorous evaluation of the user experience was conducted using qualitative and quantitative data collection. Several questionnaires (including pre and post questionnaires) were administered at different stages of the process and the interactions were also tracked for reasons of analysis in order to uncover any significant differences after the simulation process took place. The results revealed that participants, after going through the evaluation process, acquired new knowledge and capabilities that would allow them to perform better in GSD settings. The participants also agreed that they were satisfied with both the learning outcomes and the user experience.

This thesis provides two main contributions: the first of these is a training framework designed to take advantage of Virtual Agents in order to design accurate simulations of GSD interactions. Part of this training framework is also to automatically assess the student's progress. An invention disclosure has been produced, and commercialization activities are currently being addressed with regard to this contribution. The second contribution is the introduction of a new pattern model with which to acquire real world GSD problems that can be used with pedagogical purposes. Future work will consider the population of the GSD pattern based knowledgebase, along with improvements that could allow a more intelligent interaction of the Virtual Agents and facilitate the design of the simulations.

Chapter 1

Introduction

This chapter introduces the subject matter and states how this thesis addresses it. It covers the background and context in which the thesis has been conducted, explaining the motivation, the goals and contributions obtained.

1.1 Background and motivation

Global Software Development (GSD) is an emerging paradigm in which development teams are geographically distributed throughout different countries whilst working on the same projects [1]. The main reason for using this paradigm is to optimize and decrease costs by finding zones in which a skilled workforce is more readily available. GSD has other advantages which have also influenced the rapid evolution of this paradigm over the last few years, the principal factors being the leveraging of time-zone effectiveness and closer proximity to the market.

However, this type of development also entails certain drawbacks, resulting principally from distance and time differences. The fact that traditional face-to-face meetings are no longer common in GSD makes communication, coordination and collaboration more complex, as interactions require the use of technology [2]. Moreover, interactions frequently involve members who use non-native languages and whose customs and behavior may be misinterpreted [3], [4]. Linguistic and cultural differences are therefore additional variables that lead to a minimization of the effects of the well-known advantages of the application of GSD.

The distance factor, including temporal, geographical and socio-cultural distance, influences the way in which software is defined, built, tested and delivered to customers, thus affecting the corresponding stages of the software life cycle. This signifies that, more than ever before, companies require their workforce to have a balanced socio-technical skill set. Seat and Lord [5] recognize the importance of training interpersonal skills such as communication and teaming in their program for teaching interaction skills to engineering students.

In order to develop software in a globally distributed team, students and practitioners need to acquire not only new technical skills, but also the ability to collaborate and interact effectively in a variety of geographic, cultural and linguistic settings [6], [7], [8]. Training these skills in universities is not, however, easy since it necessitates providing trainees with real experiences that will allow them to develop both technical and non-technical skills [9]. Traditional GSD theoretical classes are found in literature as a response to the needs for adjustments in software engineering education [10]. In order to provide training, some universities organize joint courses in collaboration with distant universities, so that students can benefit from a wide set of knowledge and experiences [11]. As a consequence, this kind of training is not at present part of the conventional curriculum [12], [13]. One of the reasons for this lack of training is that it requires new theoretical content, training environments and practical experiences, which commonly entails a great deal of coordination with distant members and institutions and is not always practicable [14].

From an organizational perspective, companies often complain about the lack of preparation as regards their new recruits' GSD related skills. Moreover, companies are not always willing to invest in expensive training programs [15]. Many multinational companies provide their employees with some kind of theoretical training [16], usually regarding basic concepts and best practices of communication, trust, cultural differences and coordination. However, they would like to be prepared to support this kind of training, and to provide their instructors with appropriate tools, methods and course materials with which to teach these skills [17]. This would allow them to provide practitioners with real experiences and case studies that are adjusted to the reality of the companies' requirements that would connect theory with practice [18], thus creating a skilled workforce that could minimize the well-known problems that occur in GSD.

Simulation has been used in other Software Engineering fields in order to provide training in specific areas of interest, such as project management or task allocation [19]. However, no significant proposals exist in the field of GSD training. This thesis argues that the use of simulation is an appropriate solution by which to develop the skills and abilities required in GSD. The collaboration and coordination problems of the current methods could be avoided by applying simulation, and the ability to provide training in a controlled manner focusing on specific areas may make this option attractive for universities and companies.

It is for these reasons that this thesis attempts to provide training in the communication, collaboration and cultural and linguistic problems that appear in GSD through the use of simulation.

1.2 Statement of problem

Participants in GSD projects interact with people from different countries and cultures, and with different native languages, thus confronting coordination and communication problems. In order to prevent and minimize these problems, both students and practitioners need to acquire skills, knowledge and awareness that could allow them to confront the different kinds of difficulties that they may encounter in GSD. A framework based on simulation is therefore proposed as a means to provide training on GSD interactions. In order to tackle the challenges of GSD training, the framework has been designed by considering the following aspects:

- The promotion of awareness of the cultural and linguistic problems that occur in GSD. In GSD it is important to recognize specific behaviors that may lead to problems during interactions, and the ability to deal with them in an effective manner.
- The acquisition of knowledge concerning GSD concepts and problems that could lead to the development of skills as regards being resolute and effective when confronted with specific GSD circumstances and problems.
- The improvement of communication and coordination interactions performed by distributed team members by means of training.
- The provision of immediate feedback and detection of the areas on which the students should focus.
- The provision of assessment in an autonomous manner.
- The provision of a mechanism with which to easily elaborate training tasks adapted to specific needs and focused on real world problems that occur in industry.

In tackling these issues, the aim is to provide training that may reduce or alleviate some of the coordination and/or communication problems that appear in GSD settings. These issues are therefore considered in the following sub-section in which the work hypothesis and research goals are clearly defined.

1.3 Work hypothesis and objectives of the research

The work hypothesis of this thesis is the following:

It is feasible to provide effective training on GSD interactions by reproducing realistic settings and scenarios that occur in these environments.

Based on the work hypothesis, this thesis has three main goals:

1. **To define a simulation-based framework with which to provide training on GSD interactions**
2. **To develop the software required to support the training framework. The software must provide instant feedback, automatic assessment and be adaptable to different kinds of GSD scenarios.**
3. **To create a mechanism that allows real world scenarios to be gathered, which can then be applied with training purposes.**

Goal 1 can be divided into the following partial goals:

- A study of the state-of-the-art in GSD in order to extract the problems that participants and organizations have to confront in these environments.
- A study of the state-of-the-art as regards the concrete methods, techniques and tools applied to provide training in GSD.
- A study on the concrete interaction problems and on the knowledge and skills that students and practitioners need to acquire in order to efficiently interact in GSD settings.
- The design of an extensible framework that addresses the problems found in literature that will permit the training of GSD skills that are difficult to address with traditional training methods.

Goal 2 can be divided into the following partial goals:

- Development of a simulator that can provide training on GSD interaction by using Virtual Agents who play a role in the simulation. Students will interact with these Virtual Agents in order to train as regards specific GSD situations and problems.
- Development of a designer with which to easily define these simulations.
- Integration of an automated assessment method that may minimize the instructors' effort and provide students with additional feedback.
- Development of training simulations that could serve to prove the effectiveness of the framework as regards providing GSD training.
- The evaluation of the framework on a course that will check whether this framework is effective.

Goal 3 can be divided into the following partial goals:

- A study of the state-of-the-art of GSD patterns that occur during interactions.
- The definition and implementation of a collaborative tool that would serve to gather GSD patterns and scenarios and share them with the GSD community.

1.4 Context of the thesis

This thesis was developed in the Alarcos Research group at the University of Castilla-La Mancha (Spain) and Lero, The Irish Software Engineering Research Centre, University of Limerick (Ireland). The majority of the conceptual and theoretical work of this thesis was developed while the author was working at an IT company which applied GSD. However, in order to prepare and conduct the evaluation, the author had the opportunity to work at Lero. Table 1 summarizes the different stages, as explained below:

- **From January 2008 to September 2009:** In this first period, the author of this thesis started his research work in the area of GSD. The research was specifically focused on improving the software processes in companies that apply GSD.
- **From September 2009 to August 2012:** The previous research on GSD processes led the author to focus his research on GSD training and education. Based on the findings that training is a real need in GSD, this period was focused on studying the state-of-the-art regarding the matter and the definition of an

initial prototype of the framework (VENTURE prototype), which was evaluated by experts (Expert Evaluation 1).

- **From August 2012 to October 2013:** During this period the author received funding from Lero (grant 10/CE/I1855) which served to address the feedback obtained in the previous Expert Evaluation. This led to the development of an improved version of the framework (VENTURE V1) that was evaluated with a second expert evaluation (Expert Evaluation 2).
- **From October 2013:** During the last stage, the author received funding from the Alarcos Research Group and, an after analyzing the feedback obtained in Expert Evaluation 2, an improved version of the framework (VENTURE V2) was developed. This led to the final part of the thesis, consisting of preparing and conducting the evaluation (Field Study).

During the period in Lero, the core idea of the thesis received funding with which to conduct a commercial feasibility study. The objective was to investigate the potential of commercializing an improved version of the tool in industrial settings. This work involved a number of meetings with multinational companies that provided their feedback, confirming that the problems dealt in this thesis are a real cause of consternation for multinational companies. These companies frequently expend considerable amounts of money to provide training using different methods, as was also discovered in the literature review carried out.

Table 1. Research stages

Jan 2008 – Sept 2009	Sept 2009- Aug 2012	Aug 2012 – Oct 2013	October 2013 – present
Global Software Development Processes			
	GSD training and Education		
	VENTURE prototype / Expert Evaluation 1		
		VENTURE V1 / Expert Evaluation 2	
			VENTURE V2 / Field Study

This research was developed in the context of several research projects. A brief description of these projects and the contributions to each one are presented as follows.

- **ENGLOBAS:** (PII2I09-0147-8235). The goal is to improve global software development by developing a framework that will permit the structured and systematic management of the processes involved in GSD. To attain this goal, we have based our work on research carried out by the Alarcos Research Group at the UCLM in collaboration with the EIDOS Group in order to develop a solid theory and framework which can be immediately applied to the needs of European software factories. This research will help to determine the problems that affect GSD and will propose methods and guidelines that will help to avoid or to decrease the effect of said problems. The contribution to this project was to define a set of communication, coordination and control problems derived from a globally distributed environment. This set of problems was used as part of the state-of-the-art of this thesis.
- **PEGASO/MAGO** (Mejora Avanzada de Procesos de Software Globales). The goal of this project is to investigate the application of various software engineering process techniques in order to improve the quality of the software developed in global environments. The participants are the University of Castilla-La Mancha (Spain) and the University of Murcia (Spain). This project is financed by the Ministerio de Ciencia e

Innovación MICINN and the European Regional Development Fund (FEDER) (TIN2009-13718-C02-01). The contribution to this project was the definition of a set of techniques that might improve coordination and communication in GSE. These techniques were eventually used to design the architecture presented herein, and to implement the tools.

- **GLOBALIA:** (PEII11-0291-5274), funded by Consejería de Educación y Ciencia (Junta de Comunidades de Castilla-La Mancha) and co-funded by FEDER (Spain), whose goal is to improve GSD by creating a methodology that starts from a conceptual framework which studies and defines the characteristics of the DGS, and then analyze the factors that may adversely affect it (these factors may be cultural, cognitive, time difference, use inadequate communication tools, lack of knowledge management, etc.). The contribution to this project was both a systematic review of GSE tools and the study of a set of GSE technologies that can be used in such an environment.
- **ORIGIN:** (IDI-2010043 (1-5)), which is funded by CDTI and FEDER ORIGIN, primarily proposes increasing the productivity of software development activities in global scenarios and improving the quality of the software developed, based on organizational intelligence and innovation, thus increasing organizations' competitive levels. This will make a significant contribution towards enhancing Spain's role as a destination and origin of software factories, and will provide a model that is exportable to other countries and companies, with the consequent provision of consulting services in this area. A set of conceptual, methodological and software systems will be developed for this purpose which will enable providers to optimize the production of software in global scenarios, thus addressing poor communication and knowledge management, and ensuring the quality of the software developed. The main contribution to this project consists of the architecture designed to improve communication and coordination in GSE. Moreover, a first case study to demonstrate how the architecture is able to achieve its goals was also performed in this project.
- **GEODAS BC**: (TIN2012-37493-C03-01): The GEODAS Project (GEstiOnpara el DesarrolloglobAldel Software) will investigate the application of different techniques to improve those software management factories that work in global environments. These techniques will be applied in real environments in collaboration with leading companies in the international arena, such as Indra or Prologue. In this last project, a second and more complete case study was performed in order to check how using a tool that implements the architecture enables coordination and communication to be improved in a real distributed environment.
- **PAISEAN**: (Science Foundation Ireland, CSET2. GRANT 10/CE/I1855). This is an industry-focused project focused on the development of process models to support the software engineering required in industry. The main aim is to provide companies with hybrid process models resulting from research and which can be used in specific environments (http://www.lero.ie/project/paisean0). The project includes: evaluating global teaming and global teaming project management models, developing a global teaming decision support system, developing an inner source process model, developing an inner source / SME process model, investigating the development of a hybrid process model – GSD/Software for Health Information Systems/Inner Source, and studying the development of process models. The main researcher is Ita Richardson and the period of the duration of the project is from May 2011 to April 2016.

1.5 Structure of the thesis

This thesis has been divided into eight chapters. The first is this introductory chapter; the second summarizes the state-of-the-art of the main topics related to this thesis. The third presents the methodology followed to conduct the thesis. The fourth chapter details the main contribution of the thesis, which is the training framework. Two expert evaluations served to improve the framework and the supporting tools, and these are presented in the fifth chapter. These expert evaluations led to an improved version of the set of tools that support the framework, and which are presented in the sixth chapter. The Field Study used to obtain the results regarding the use of the framework is presented in Chapter seven. Finally the conclusions obtained and the analysis of how the hypothesis proposed in this thesis has been demonstrated, are presented in the last chapter.

The eleven appendices in this thesis present background material on: a literature review on assessment methods (Appendix I), a literature review on GSD patterns (Appendix II), questionnaire applied in the Expert Evaluation 1 (Appendix III), questionnaires applied in the Field Study (Appendix IV to Appendix VIII), some statistical calculations required for the analysis (Appendix IX) and the script of the two training scenarios used for the Field Study (Appendix X and XI).

Chapter 2

State of the art

This chapter describes the state-of-the-art of the core areas covered by this thesis. As depicted in Figure 1, the first topic studied is the concept of GSD, and it is here that we analyze the problems and solutions that are applied according to a systematic literature review performed regarding this topic. The findings reveal that training and education are key success factors in preparing engineers to tackle the challenges they will confront in this new development paradigm. Bearing in mind the importance of training and education, the second area studied is global software development education. In order to gain a broad understanding, a systematic literature review was carried out to discover how universities and companies train and teach GSD. The strengths, weaknesses and problems of the existing methods were then analyzed with the aim of finding unresolved issues that require further research. The third area studied is the skills required by a software engineer interacting in GSD settings. The focus here is on the problems as regards cultural and linguistic differences that software engineers must confront, as these are critical for clear understanding and collaboration.

Finally, existing learning environments (independent of their domain) designed to improve communication and coordination skills are reviewed. The aim here is to gain an understanding of the types of methods used to train and enhance skills that have worked in different domains. This study revealed that the use of simulation and Virtual Agents has a significant impact on improving skills, and these two areas (simulation and Virtual Agents) were therefore studied more extensively in order to see how they could be used in our proposal.

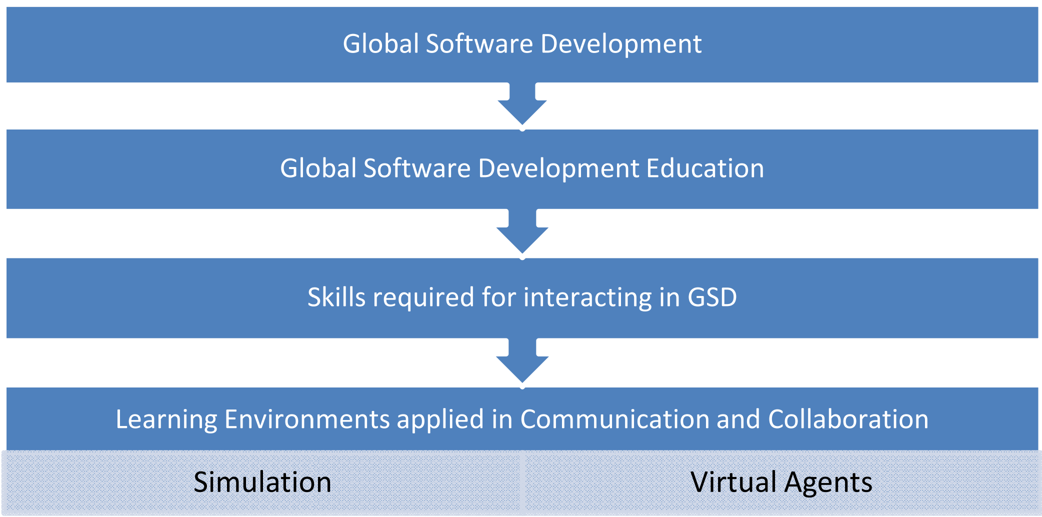


Figure 1. Topics dealt with in the State of the Art

2.1 Global Software Development

Recent years have witnessed a trend towards the globalization of many organizations and industries, including the software industry. Many companies have consequently changed from being co-located developments in which all team members are in the same building to distributed developments in which team members might be in different building or even different countries [20]. Distributed Software Development [21] therefore allows team members to be located in various remote sites during the software lifecycle, thus making up a network of virtual teams [22]. In some cases, these teams may be members of the same organization, while in other cases collaboration or outsourcing involving different organizations may exist.

Traditional face-to-face meetings are, therefore, no longer common, and interaction between members requires the use of technology to facilitate communication and coordination. Although this phenomenon began in the

90`s, only during the last ten years has its strategic importance been recognized [23], and related studies are very recent [24].

The distance between the different teams can vary from a few meters (when the teams work in adjacent buildings) to different continents [25]. “Offshoring”, meanwhile, refers to the transfer of an organizational function to another country, usually one in which human resources are cheaper. The term “nearshoring” refers to those situations in which jobs are transferred to geographically closer countries, thus avoiding cultural and time differences between members and saving travel and communication costs. Outsourcing is a means to contract an external organization, independently of its location, rather than developing in-house [26].

The situation in which the teams are distributed beyond the limits of a nation is called Global Software Development (GSD) [1]. This kind of scenario is interesting for several reasons, principally because it enables organizations to be abstracted from geographical distance, whilst having qualified human resources ([27], [1], [28]) and minimizing cost ([29], [30], [31]), thus increasing the market area ([1], [32], [33]) by producing software for remote clients and obtaining a longer workday by taking advantage of time differences [34]. One example of this is the “follow-the-sun” model, whereby teams coordinate to cover 24 hours of the working day ([35], [36], [20]).

There is a combination of terms used in research on GSD, including stand-alone terms and terms used in a combination with other terms. These have been studied by Šmite et al. [37] and are shown in Table 2.

Table 2. Terms for GSD

Terms	Meaning
Global software engineering	Development of a software artifact in more than one location
Insourcing	Leveraging company-internal human resources
Nearshoring	Leveraging resources from a neighboring country
Offshore insourcing	Leveraging company-internal resources situated in a different country
Offshore outsourcing	Leveraging external third-party resources situated in a different country
Offshoring	Leveraging resources from a different country
Onshore insourcing	Leveraging company-internal resources situated in the same country
Onshore outsourcing	Leveraging external third-party resources situated in the same country
Onshoring	Leveraging resources from the same country
Outsourcing	Leveraging external third-party resources
Sourcing	Leveraging resources

However, a number of problems [38] caused mainly by temporal distance ([39] [20]), geographical distance ([40], [41]) and cultural differences [42] that directly affect the processes of communication, coordination and control activities [43] must be confronted. In these environments, communication is less fluid than in co-localized development groups as there is a lack of face-to-face contact [44]. This leads to a high dependency on information and communication technologies [45]. Problems related to coordination, collaboration or group awareness therefore appear which negatively affect productivity and software quality. These factors all influence the way in which software is defined, built, tested and delivered to customers, thus affecting the corresponding stages of the software life cycle.

Table 3 shows a summary of the benefits (represented as ✓) and risks (represented as ×) of GSD presented in the framework proposed by Ågerfalk et al. [39]:

Table 3. GSD benefits and risks [39]

Processes	Dimensions		
	Temporal Distance	**Geographical Distance**	**Socio-Cultural Distance**
Communication	✓ Time zone effectiveness × Delayed communication × Delayed feedback	✓ Proximity to market/customer × Lack of informal communication × Dependency on ICT × Increased effort to initiate contact × Providing technical infrastructure × Cost of travel	× Asynchronous communication preferred by non-native speakers × Language differences and misunderstandings × Cultural differences and misunderstandings × Managing frames of reference
Coordination	✓ Time zone efficiency × Reduced hours of collaboration × Synchronized team meetings difficult × Availability of technical infrastructure × Coordination complexity ✓ Modularization of work × Lack of mechanisms for creating shared understanding × Management of project artifacts	✓ Access to large skilled labor pool ✓ Cross-site modularization of work × Reduced trust × Lack of awareness/team spirit × Lack of mechanisms for creating shared understanding × Coordination complexity	✓ Mix of skills and experiences × Language and cultural training × Lack of domain knowledge × Doubt about others' capabilities × Lack of mechanisms for creating shared understanding × Coordination complexity × Lack of awareness/team spirit
Control	× Management of project artifacts ✓ Time zone effectiveness	✓ Allocation of roles and team structure × Lack of concurrent engineering principles	× Adapting to local formalized norm structures × Different perceptions of authority/hierarchy

Organizations must consider these advantages and disadvantages of GSD in order to decide whether to go global by focusing on their specific circumstances. A Systematic Literature Review on GSD was therefore conducted for the purpose of this thesis, which allowed the main problems confronted by distributed teams to be determined. Having detected that education is an important success factor, a second Systematic Literature Review on GSD education was conducted. The results of both studies are presented below.

2.1.1 Systematic Literature Review

A systematic review of literature [46] permits the identification, evaluation and interpretation of all the available relevant studies related to a particular research question, topic area or phenomenon, thus providing results of a high scientific value by classifying studies into primary studies and secondary or relevant studies, by means of synthesizing existing work according to a predefined strategy.

The systematic search procedure provided by Kitchenham [46] and the selection of primary studies method followed by Pino et al. [47] have been applied in this study.

Question Formularization

The research question that guided this systematic review was: *What are the initiatives carried out in relation to the improvement of DSD processes?*

The keywords that guided the search to answer the research question were: *distributed*, *software*, *development*, *global*, *enterprise*, *organization*, *company*, *team*, *offshore*, *offshoring*, *outsource*, *outsourcing*, *nearshore*, *nearshoring*, *model*, *strategy* and *technique*.

The ultimate goal of this systematic review was to identify the best procedures, models and strategies employed, and to determine the most important improvement factors for the main problems found. The population was composed of publications found in the selected sources which apply procedures or strategies related to DSD.

Sources Selection

The search strings (shown in Table 4) were established by combining the keyword list from the previous section using the logical connectors "AND" and "OR".

Table 4. Basic search strings

	Basic search strings
1	(“*distributed software development*” OR “*global software development*”) AND ((*enterprise* OR *organization* OR *company* OR *team*) AND (*offshore* OR *offshoring* OR *outsource* OR *outsourcing* OR *nearshore* OR *nearshoring*))
2	(“*distributed software development*” OR “*global software development*”) AND (*model* OR *strategy* OR *technique*)

The studies were obtained from the following search sources: Science@Direct (www.sciencedirect.com), Wiley Interscience (www.interscience.wiley.com), IEEE Digital Library (www.computer.org) and ACM Digital Library (portal.acm.org/dl.cfm). The quality of these sources guarantees the quality of the studies. The basic search chains had to be adapted to the search engines of each source.

Studies Selection

The inclusion criteria used to determine whether a study should be considered relevant (a potential candidate to become a primary study) was based on an analysis of the titles, abstracts and keywords in the studies retrieved by the search to determine whether they dealt with DSD as regards being orientated towards process improvement, quality, coordination, collaboration, communication and related issues that make any improvement as regards the subject in question. In some cases it was necessary to read the entire document to determine its relevance.

After analyzing the results of the first iteration of the systematic review, exclusion criteria were applied to obtain the primary studies, and those studies which, despite addressing the issue of DSD, did not contribute to any significant improvement method were excluded. Those studies which focused solely upon social issues, cultural or time differences or focused solely upon free software were also dismissed, although other papers that address these topics in a secondary manner were taken into consideration.

The search procedure produced 768 initial studies, of which 497 were not repeated. 170 of these were selected as being relevant and 78 were selected as primary studies. Table 5 shows the distribution of studies found according to the sources used.

Table 5. Distribution of studies found

	Studies				
Sources	Found	Not repeated	Relevant	**Primaries**	**%**
Science@Direct	175	143	53	**19**	23,8
Wiley InterScience	30	20	17	**13**	16,3
IEEE Digital Library	66	49	19	**14**	18,8
ACM Digital Library	497	355	80	**32**	41,3
Total	768	567	170	**78**	100,0

Information Extraction

The process of extracting information from the primary studies followed an inclusion criterion based on obtaining information concerning the key success factors, improvement strategies employed, processes improved and the most important ideas in each study, thus establishing a categorization between objective and subjective results. All articles were categorized by paying close attention to the methodological study followed according to the models presented in [48]; these categorizations are as follows:

- case studies,
- literature reviews,
- experiments,
- simulations and
- surveys.

The non-experimental model for studies (which makes a proposal without testing it or performing experiments) was also applied.

Information corresponding to a specific template (including the type of study, methodology employed, processes affected and a description of the approach) was extracted from each paper selected for analysis, particular attention being paid to the problems dealt with and the solutions contributed.

2.1.2 Trends in Distributed Software Development research

This section analyzes and discusses the content of the primary studies found in order to extract relevant information.

Figure 2 (left) shows that the majority of the primary studies analyzed are case studies and experimental papers. Non-experimental studies and surveys in which members involved in the development take part in outlining their difficulties have a significant representation.

However, as Figure 2 (right) shows, the majority of primary studies are focused upon the business field, but studies in the university environment also appear in which groups of students carried out developments in different locations. 38% of the studies did not indicate their field of work or their classification was not applicable owing to the nature of the study, while 6% were from organizations which did not specify their corporate or university environment.

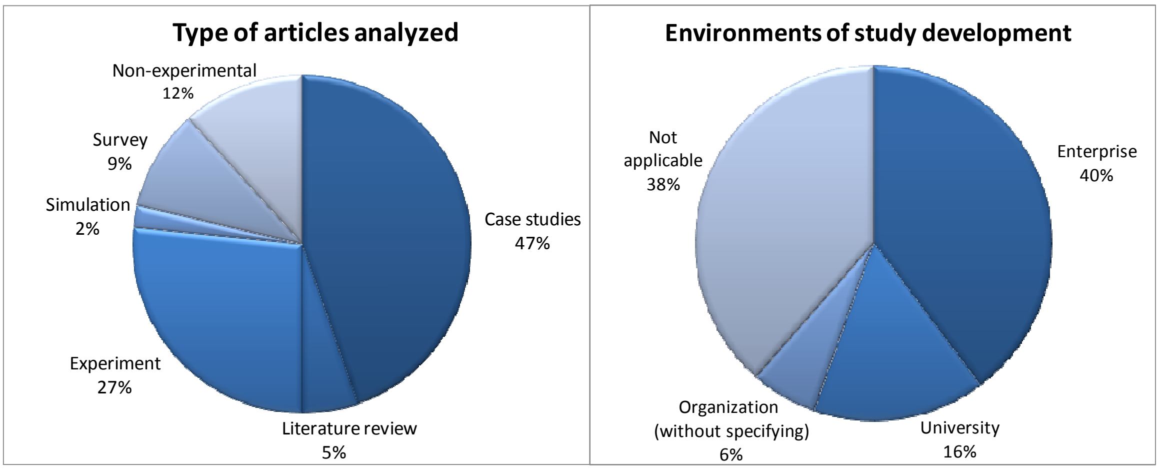


Figure 2. Type of articles analyzed (left), and environments of study development (right)

Standards Employed

Figure 3 presents the standards addressed by the articles analyzed. On the basis of the data available, it may be inferred that few studies indicate the use of specific standards. This is, in part, attributable to the fact that the vast majority of studies deal with issues such as communication difficulties in which the standard used is not of importance. The standards supported by most primary studies are CMM, CMMI and ISO 9001, it being common to jointly apply both. The majority of the studies which applied CMM and CMMI employed a maturity level of 2.

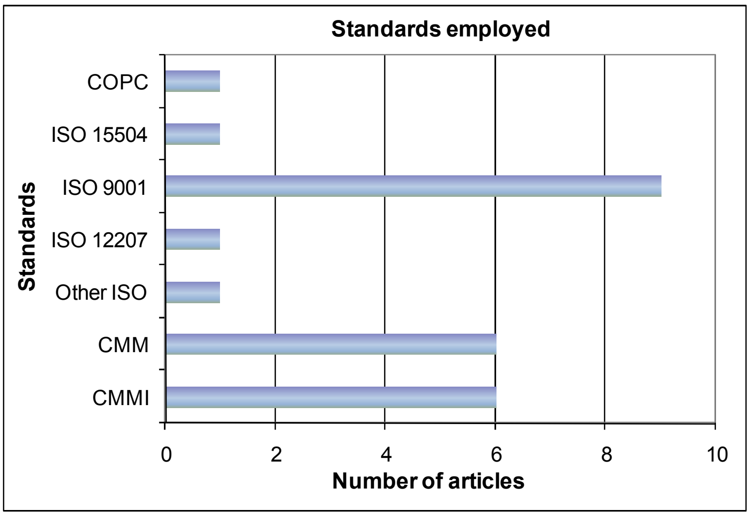


Figure 3. Standards employed in the studies

Improved or Analyzed Processes

Taking the primary studies analyzed as a reference, a classification in terms of processes in the software life cycle was carried out in order to consider those processes to which improvements were proposed or success factors or areas to be improved related to DSD were discussed. Primary studies were classified according to the improved or studied processes, in each case based on the ISO/IEC 12207 standard [49], with the aim of obtaining a vision of the process life cycle that requires special attention when working in a distributed environment, and discovering the improvement efforts made until that moment.

The ISO 12207 standard establishes the activities that may be carried out during the software life cycle, which are grouped into main processes, support processes and general processes. The results are presented graphically in Figure 4, which indicates the frequency with which the studies in question address each process.

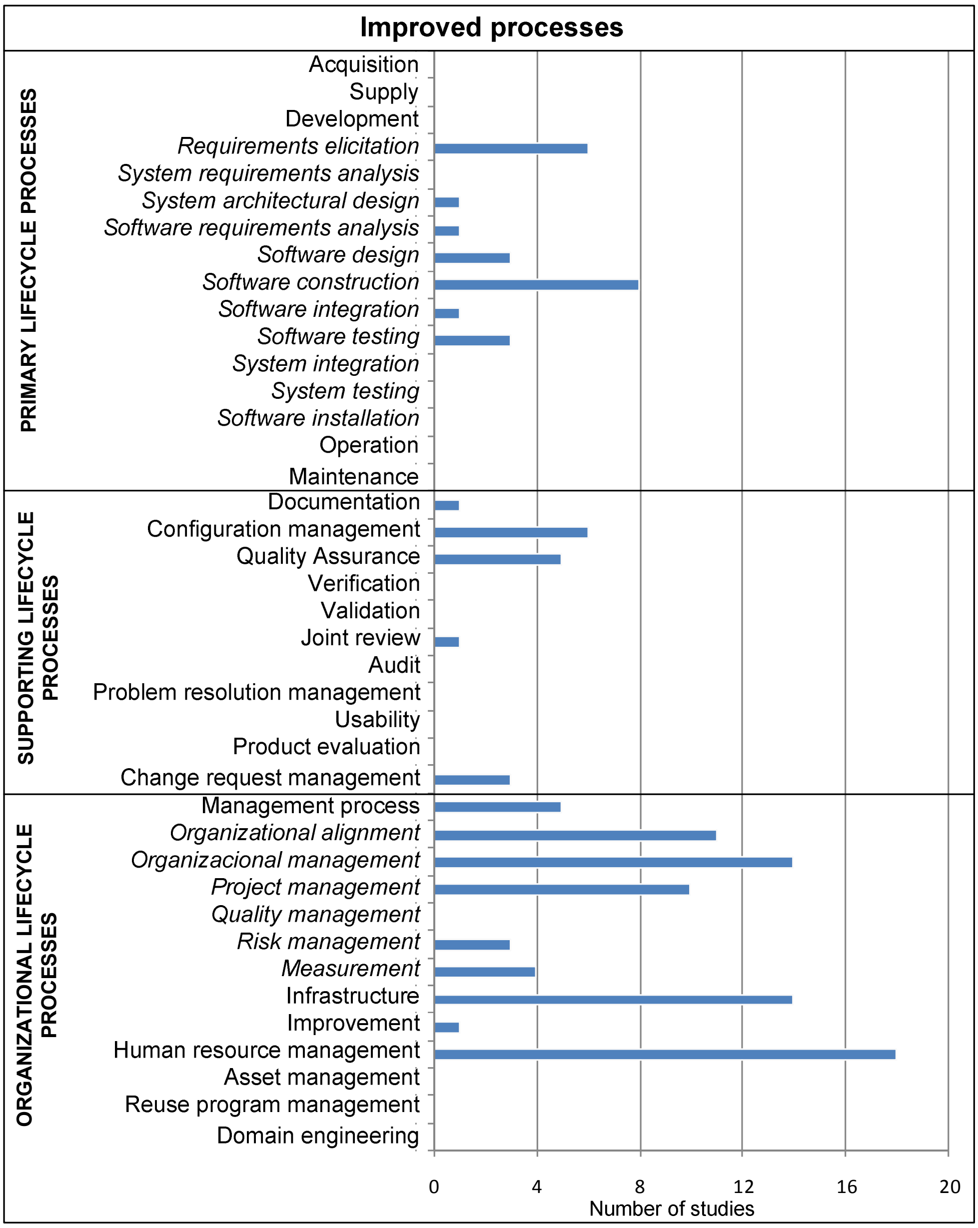


Figure 4. Processes improved or analyzed by the primary studies adjusted to ISO 12207

The results obtained indicate that more efforts are focused on human resources, organizational management, infrastructure, organizational alignment and project management. From these data it can be inferred that communication between team members is a critical factor. Most of the studies are focused on the organizational processes, and it is thus believed that there is a need for more studies focused on the level of projects and technical aspects.

Contents of the Studies

Table 6 provides a schematic representation of the lines on which the primary studies have focused. Most of the works study tools or models designed specifically for DSD attempt to improve certain aspects related to development and coordination. Another large percentage of the studies are related to communication processes and the integration of collaborative tools, combining tools such as e-mail or instant messaging, and studying their application by means of different strategies. Most of the studies address the subject of communication difficulties in at least a secondary manner, presenting this aspect as being one of the most important in relation to the problematic nature of DSD.

Table 6. Thematic areas dealt with in the primary studies

Thematic areas	Studies (%)
Process control, task scheduling and project coordination	43.5
Collaborative tools, techniques and frameworks	35.9
Configuration management	5.4
Multi-agent systems	4.3
Knowledge management	7.6
Defects detection	2.2
Test management	1.1

Table 7 shows the primary studies found.

Table 7. Primary studies selected in the systematic review

Reference	Year	Source	Methodology	Reference	Year	Source	Methodology
[50]	2002	IEEE	Use Case	[51]	2004	Wiley Interscience	Use Case
[52]	2007	ACM	Use Case	[53]	2008	Wiley Interscience	Survey
[54]	2006	ACM	Use Case	[55]	2008	Wiley Interscience	Non experimental
[56]	2006	IEEE	Experimental	[57]	2008	Wiley Interscience	Use Case
[58]	2006	IEEE	Experimental	[59]	2007	Science Direct	Use Case
[60]	1998	Science Direct	Literature review	[61]	2004	Science Direct	Use Case
[62]	2006	IEEE	Use Case	[63]	2007	Science Direct	Use Case
[64]	2008	ACM	Experimental	[65]	2008	ACM	Use Case
[66]	2007	Science Direct	Use Case	[67]	1991	Science Direct	Use Case
[68]	2007	Science Direct	Use Case	[69]	2007	Wiley Interscience	Use Case
[70]	1999	ACM	Non experimental	[71]	2008	ACM	Experimental
[72]	2008	ACM	Experimental	[73]	2003	Wiley Interscience	Use Case
[74]	2004	Science Direct	Use Case	[75]	2003	Wiley Interscience	Use Case, Survey
[76]	2004	ACM	Use Case	[77]	2006	ACM	Use Case
[78]	2004	ACM	Literature review	[79]	2008	Wiley Interscience	Use Case
[80]	1996	Science Direct	Experimental	[81]	2003	Wiley Interscience	Use Case
[82]	2008	ACM	Experimental	[83]	2006	IEEE	Use Case

Reference	Year	Source	Methodology	Reference	Year	Source	Methodology
[84]	2006	ACM	Use Case	[85]	2007	ACM	Use Case
[86]	2008	Science Direct	Experimental	[87]	2005	Science Direct	Literature review
[88]	2006	IEEE	Experimental	[89]	2008	ACM	Experimental
[90]	2000	ACM	Survey	[91]	2006	Science Direct	Experimental
[92]	2001	ACM	Survey	[93]	2008	ACM	Experimental
[94]	2008	ACM	Experimental	[95]	2003	ACM	Experimental
[40]	2005	ACM	Survey	[19]	2007	Wiley Interscience	Simulation
[96]	2008	ACM	Experimental	[97]	2004	ACM	Experimental
[98]	2006	IEEE	Use Case, Survey	[99]	2006	IEEE	Literature review
[100]	2008	ACM	Use Case	[101]	2006	Wiley Interscience	Use Case
[102]	2008	Science Direct	Use Case	[103]	2007	ACM	Experimental
[104]	2006	IEEE	Experimental	[105]	2006	IEEE	Use Case
[106]	2004	ACM	Use Case	[107]	2005	ACM	Experimental
[38]	2006	Science Direct	Use Case	[108]	1999	IEEE	Non experimental
[109]	2006	ACM	Survey	[110]	1995	Science Direct	Non experimental
[111]	2006	ACM	Survey	[112]	2006	Science Direct	Experimental
[113]	2006	IEEE	Use Case	[114]	2006	Science Direct	Use Case
[115]	2006	IEEE	Experimental	[116]	2007	ACM	Use Case
[117]	2006	Science Direct	Non experimental	[118]	2005	IEEE	Non experimental
[119]	2007	ACM	Experimental	[120]	2008	ACM	Experimental
[121]	2002	Science Direct	Experimental	[122]	2003	Wiley Interscience	Use Case
[123]	2006	ACM	Use Case	[124]	2006	Wiley Interscience	Non experimental

2.1.3 Challenges and improvements

This section provides a synthesis of the challenges and proposed improvements identified through the systematic review, and the main subjects are also discussed.

Communication

The software life cycle requires a great deal of communication between those members involved in the development who exchange a large amount of information through the use of different tools and formats without following communication standards, and who thus confront misunderstandings and high response times. These drawbacks, combined with the complex infrastructure and the tremendous size of personal networks which change over time, are summarized in a decrease in communication frequency and quality, which directly affects productivity. In order to decrease these effects, both methodologies and processes must be supported by collaborative tools, which are a means of avoiding ambiguity and face-to-face meetings without comprising the quality of the results, as is proposed by Ali-Babar et al. [91]. Mohan and Ramesh [59] discuss the need for user-friendly tools, and integrate collaborative tools and agents to improve knowledge integration. Thissen et al. [116] examine communication tools and describe collaboration processes, dealing with techniques such as conference calls and e-mail.

Cultural differences imply different terminologies which may lead to mistakes in messages and translation errors. Different levels of understanding of the problem domain also exist, as do different levels of knowledge, skills and training between teams. The use of translation processes, and codification guidelines is therefore useful [60].

Requirements should also be clearly defined and modeled in order to make them easily understood, and dependencies among modules should be identified in the architecture. Aranda et al. [113] propose a technique with which to reduce communication problems in the process of requirements elicitation by selecting a suite of groupware tools and techniques from the field of cognitive psychology.

The security of communications must also been taken into account. All the members involved must be able to work with several tools, and the human factor takes on more importance; the team members' communication skills are a critical factor.

Group awareness

Members of a virtual team tend to be less productive as a result of feelings of isolation and indifference. Literature deals with the poor socialization and socio-cultural differences which cause a lack of trust [57]. Developers need to have as much information as possible at their disposal, and to know the full status of the project and its past history, which will in turn allow them to create realistic assumptions about the project. Frequent changes in processes, lack of continuity in communications and lack of collaborative tool integration cause remote groups to be unaware of what is important because they do not know what other people are working on. As a consequence, they cannot find the right person and/or timely information which will enable them to work together efficiently, thus resulting in misalignment, re-planning, redesign and rework.

Storey et al. [107] propose a framework for the comparison and understanding of visualization tools that provides awareness of software development activities, giving a solid grounding to the existing theoretical foundation of the field. Froehlich and Dourish [76] similarly describes a visualization tool which supports DSD processes by creating visual representations of both software artifacts and software development activities, thus allowing developers to explore the relationships between them.

Holmes and Walker [96] present the YooHoo awareness system which helps developers to keep apprised of code changes, providing notifications in a flexible manner.

Apart from using these tools, the development process must also be adapted to provide the team members with a better awareness of the project status. It must therefore be automated to provide notifications of actions and decisions to the roles involved.

Software configuration management

Distributed environments have problems resulting from conflicts related to source code control. Coordination and synchronization become more complex as the team's degree of distribution grows, and traceability is a critical factor. Source control systems must support access through Internet, thus confronting its unreliable and insecure nature and the higher response times.

In order to reduce these drawbacks, Dossick and Kaiser [70] propose CHIME, an Internet and Intranet based application which allows users to be placed in a 3D virtual world representing the software system. Users interact with project artifacts by "walking around" the virtual world, in which they collaborate with other users through the use of a feasible architecture. Al-Ani et al. [72] presents a similar tool which visualizes the developers and artifacts in a project using a 3D metaphor and gives managers an overview of ongoing activities in the project. With the same purpose in mind, Biehl et al. [52] present FASTDash, a user-friendly tool that uses a spatial representation of the shared code base which highlights team members' current activities, thus allowing a developer to rapidly determine which team members have source files checked out, which files are being viewed, and what methods and classes are currently being changed. This provides immediate awareness of potential conflict situations, such as two programmers editing the same source file.

Knowledge management

The team members' experiences, methods, decisions, and skills must be accumulated during the development process through effective information-sharing mechanisms, so that each team member can use the experience of his/her predecessor and the experience of the team accumulated during development, thus saving costs and time by avoiding redundant work. Distributed environments must facilitate knowledge sharing by maintaining a product/process repository focused on well understood functionality by linking content from sources such as e-mail and online discussions, and sharing metadata information among several tools.

Ali-Babar [94] proposes solving the drawbacks caused by distribution by applying an electronic workspace paradigm with which to capture and share knowledge in order to support the software architecture processes.

Zhuge [121] presents an approach that works with a knowledge repository in which information related to each project is saved by using internet-based communication tools, thus enabling a new team member to become quickly experienced by learning the knowledge stored.

Mohan and Ramesh [59] present an approach based on a traceability framework that identifies the key knowledge elements which are to be integrated, and a prototype system that supports the acquisition, integration, and use of knowledge elements. This allows the knowledge fragments stored in diverse environments to be integrated and used by various stakeholders in order to facilitate a common understanding.

Change cannot be limited solely to tools, but must also take place in the organization and role distribution. Documentation must always be updated and structured to prevent assumptions and ambiguity, thereby facilitating the maintainability of the software developed.

Coordination

Coordination in multi-site developments becomes more difficult in terms of articulation work, as problems resulting from communication, lack of group awareness and the complexity of the organization appear which influence the way in which the work must be structured and managed [54]. Herbsleb et al. [92] suggest that multi-site communication and coordination requires more people to participate which causes delays. Large changes involve multiple sites and greater implementation times. Changes in multiple distributed sites involve a large number of people. More progress reports, project reviews, conference calls and regular meetings to take corrective action are therefore needed, thus minimizing task dependencies with other locations. Collaborative tools must support analysis, design and development to permit monitoring activities and managing dependencies, notifications and implementation of corrective measures. Ovaska et al. [73] study the coordination of interdependencies between activities, including the figure of a chief architect to coordinate the work and maintain the conceptual integrity of the system.

Setamanit et al. [19] describes a simulation model to study different ways in which to configure global software development processes. These models, which are based on empirical data, permit research into and the calculation of the impact of coordination efficiency and its effects on productivity.

Souza et al. [103] present the Ariadne tool which analyzes software projects for dependencies and helps to find coordination problems through a visual environment.

Collaboration

Software development is a collaborative activity in which business analysts, customers, system engineers, architects and developers interact. The concurrent edition of models and processes requires synchronous collaboration between architects and developers who cannot be physically present at a common location. Software modeling requires real-time concurrency control, thus enabling geographically dispersed developers to edit and discuss the same diagrams, and improving productivity by providing a means to easily capture and model difficult concepts through virtual workspaces and the collaborative edition of artifacts by means of tools which permit synchronized interactions. S. Liu et al. [115] present an interesting approach which can support real-time collaborative UML-based modeling.

Bruegge et al. [56] describe SYSIPHUS, a distributed environment which provides a uniform framework for system models, collaboration artifacts, and organizational models, with services that can be used to explore, search, filter and analyze the models.

A further approach is presented by Suzuki and Yamamoto [108] with the SoftDock framework which solves the issues related to software component modeling and their relationships by describing and sharing the information in component models and ensuring their integrity. Developers can therefore work by analyzing, designing, and developing software from component models, and transfer them by using an exchange format, thus permitting communication between team members. Sarkar et al. [93] describe CollabDev, a human assisted collaborative knowledge tool with which to analyze applications in multiple languages and provide those members involved in maintenance with various structural, architectural, and functional insights.

Biehl et al. [100] present IMPROMPTU, a framework for collaboration in multiple display environments, which allows users to share task information through displays via off-the-shelf applications.

In another direction, Xiao et al. [119] study Galaxy Wiki, an on-line collaborative tool based on the wiki concept which permits the existence of a collaborative authoring system for documentation and coordination purposes, thus allowing developers to compile, execute and debug programs on wiki pages.

For an organization, the most valuable characteristics of these kinds of tools are their simplicity, usability, accessibility, adaptability and broadband requirements. It can therefore be inferred that proposals based on the wiki concept and Intranet web-based environments are more generic and easier to apply.

Project and process management

High organizational complexity, scheduling, task assignment and cost estimation become more problematic in distributed environments as a result of volatile requirements, changing specifications, cultural diversity and the lack of informal communication [62]. Managers must control the overall development process, improving it during the enactment and minimizing any factors that may decrease productivity, taking into account the possible impact of diverse cultures, identifying interrelated tasks and minimizing dependencies among distributed groups.

The maturity of the process becomes a key success factor. Passivaara and Lassenius [75] propose incremental integration and frequent deliveries by following informing and monitoring practices.

Spanjers et al. [105] present SoftFab, an infrastructure which enables projects to automate the building and test process, and which manages all the tasks remotely though a control center.

Gousios et al. [82] proposes a model for the evaluation of developers' contributions by combining traditional metrics with data mined from software repositories to extract contribution indicators. In the same line, Nagappan et al. [65] present a metric scheme with which to quantify organizational complexity.

Madachy [55] deals with economic issues, presenting a set of cost models with which to estimate distributed teams' work, and which take into account the teams' different environmental characteristics, localized labor categories, calendars, compensation rates, and currencies for costing.

It is consequently necessary to automate the process through the use of an adaptable tool in order to manage tasks and metrics using customizable reports managed by a central server and ensuring the application of the development processes in compliance with a predefined standard.

Process support

Processes should reflect the direct responsibilities and dependencies between tasks, notifying the people involved of the changes that concern them, thus avoiding the information overload of team members. Process modeling and enactment should support the inter-site coordination and cooperation of the working teams, offering automated support to distributed project management. Problems resulting from process evolution, mobility and tool integration appear within this context. Process engines have to support changes during enactment. Furthermore, distributed environments usually involve a large network of heterogeneous, autonomous and distributed models and process engines, which requires the provision of a framework for process system interoperability.

In relation to these problems, Fernández et al. [51] present the SPEARMINT process modeling environment, which supports extensive capabilities for multi-view modeling and analysis, and XCHIPS for Web-based process support which permits enactment and simulation functionalities.

Setamanit et al. [19] describe a hybrid computer simulation model of software development processes with which to study alternative ways in which to configure GSD projects in order to confront communication problems, control and coordination problems, process management and time and cultural differences.

Quality and measurement

The quality of products is highly influenced by the quality of the processes that support them. In DSD projects the impact of issues may be magnified when a problem is discovered, and it is more difficult to recover from this than in collocated projects. Organizations should introduce new quality assurance models and measures to obtain information which can be adapted to the distributed scenarios, thus ensuring that the requirements reflect the customer's needs. One of the most frequently recommended practices is that of automated code inspections [56] and the application of coding standards. With this aim in mind, Siakas and Balstrup [124] propose the eSCM-SP capability model, which has many similarities to other capability-assessment models such as CMMI, Bootstrap or SPICE and the SQM-CODE model, and considers the factors that influence software quality management systems from a cultural and organizational perspective.

Herbsleb et al. [90] work with several interesting measures, such as the interdependence measure which allows the degree of dispersion of work among sites to be determined by looking up the locations of all the individuals involved. Lanubile et al. [122] similarly propose metrics associated with products and processes oriented towards software defects such as: discovery effort, reported defects, defects density, fixed defects or unfixed defects.

Furthermore, software architecture evaluation usually involves a large number of stakeholders who need face-to-face evaluation meetings, and adequate collaborative tools are therefore needed, such as that proposed by Ali-Babar et al. [91].

A lack of empirical studies that permit the enumeration of reliable measures has been observed, and more articles related to tests in distributed environments, which are directly related to software quality, are also necessary.

Risk management

Risk management is a critical project management activity. In addition to all the known traditional issues connected with collocated environments [62], DSD development includes issues related to coordination, problem resolution, evolving requirements, knowledge sharing and risk identification [125]. Software defects become more frequent owing to the added complexity which is, in most cases, related to communication problems and a lack of group awareness. Defects control must be adapted by making a greater effort in risk management activities. The use of adequate measures and the requirements definition are important key factors.

Lanubile et al. [122] attempt to minimize these problems by defining a process in which they specify roles, guidelines, forms and templates, and describe a Web-based tool that adopts a reengineered inspection process in order to minimize synchronous activities and coordination problems and thus support geographically dispersed teams.

Kuni and Bhushan [104] propose the WOOM methodology in order to provide measures and facilitate decision making, taking into account the risks during both various lifecycle phases and mitigation plans.

Rules and guidelines with which to organize the teams and their interactions have become necessary. Teams must be continuously controlled in order to detect problems and take corrective actions.

2.1.4 Systematic Literature Reviews on GSD

This section summarizes the Systematic Literature Reviews that have been published since 2008 until the present as regards the field of GSD. Table 8 presents the forty studies that were found, which deal with topics such as knowledge and project management, requirements engineering, configuration management, collaboration, communication, coordination and education. The most relevant for this thesis are introduced below in temporal order of publication.

In 2009, Hossain et al. [126] conducted an SLR on Scrum in GSD in which they found that the main concern in this area is communication problems. They selected a number of studies regarding strategies on the use of synchronous communication, collaboration difficulties, communication bandwidth problems or tools applied by Scrum teams, which shows that Scrum teams confront similar problems to those reported in this thesis.

In 2010, a new SLR on barriers and solutions on GSD was conducted by Noll, et al. [127]. Communication is again one of the main concerns reported by the primary studies, in which, in accordance with the results of this thesis, cultural and linguistic distance play an important role. As a consequence of these differences and the temporal and geographical distance, fear and trust problems also appear, as was also reported in the SLR described. In 2011, Smite and Wohlin [128], also conducted an SLR, and subsequently reported an absence of success stories and proven solutions in GSD, thus reaching the conclusion that this field is not yet mature, as found in the conclusions shown in the previous section. Most of the studies found explore the challenges of collaborative work, and few studies that focus on a particular topic, practice, or development phase were discovered. The most popular topics are requirements engineering, coordination and communication, and the application of agile processes. A number of studies were also found regarding the fields of design, testing, engineering management, engineering process, tools and methods and quality. The authors of the aforementioned study suggest that communication difficulties can be overcome by: investing in face-to-face meetings, rich communication media and promoting effective communication through synchronous interaction.

Table 8. Topics studied by the Systematic Literature Reviews

Area	Topic	Studies
Research	Reports on existing literature reviews	[129], [130], [131], [132]
	Empirical studies	[132], [133], [128], [134]
	Ontology	[135]
Knowledge management	Managing architectural knowledge	[136]
Development process	Scrum	[126], [137], [138]
	Agile methodologies	[139], [140], [141], [142]
	Risks management	[143], [128]
	Process and models	[144], [145]
Project management	Tools	[146], [146], [147], [148], [149]
	Success factors	[150], [147]
Requirements	Models, techniques and tools	[151], [152]
	Proposals for dealing with risks and challenges	[153]
Configuration management	Problems and solutions	[154], [155]
Offshore and outsourcing	Patterns, barriers and success factors	[24], [156], [157]
Collaboration, communication and coordination	Cultural, geographic and temporal distance in communication and coordination	[127], [158]
	Barriers and solutions	[159]
	Collaboration	[145]
	Team awareness	[160], [161], [162]
	Follow the sun	[163]
Teaching	Techniques and use cases to provide GSD education	[164]

In 2012, Verner et al. [131] also used a tertiary study in the field of SLRs on GSD, and detected an increasing amount of research on GSD, . Moreover, for the years 2009, 2010 and 2011, the number of studies related to GSD increased, which fits with the tendency forecasted in the SLR performed by the candidate. The principal GSD topics covered include: organizational environment, project execution, and project planning and control. The topics of requirements engineering, configuration management, culture, collaboration, communication, control and distance are still of special relevance in this matter.

Finally, similar results were obtained in 2013 by Kroll et al. [163] with an SLR on best practices and challenges in Follow-the-Sun. In this study, a number of studies on the loss of teamness, coordination breakdown, communication difficulties, socio cultural diversity and language differences were reported. Among the best practices proposed, cultural awareness training is mentioned. The importance of developing cultural awareness among team members should, according to the authors' conclusions, be implemented from the beginning in order to educate team members in each other's cultures. This shows that education in cultural issues is still a challenge that should be explored.

In summary, the proposals found by these SLRs have discovered that communication problems are still a central area in GSD, and collaboration and cultural distance are topics that need improvements and training. Upon analyzing these proposals, it was found that:

- The training of human resources in the new challenges that they must confront in distributed environments is a key success factor [92]. Human capital abilities are an important factor in minimizing problems. They must therefore be aware of the challenges that they may confront and possible solutions in order to be more effective. Human resources can also participate in improvement initiatives by providing their perspective and feedback through surveys or interviews aimed at detecting problems or suggesting improvements [91], [90].
- It is necessary to establish an efficient communication mechanism between the members of the organization, thus allowing the status and changes made within each project to be discovered [110], [52]. Using a common version control tool helps to control conflictive situations and attain a sense of awareness of the work conducted in other locations [77].
- There must be a manner in which to permit the planning and scheduling of distributed tasks, taking into account costs and dependencies between projects [76], [55]. The registration of activities is essential in promoting awareness and tracking pending issues, errors and people in charge [52], [107].
- Maturity models (such as CMMI) and agile methodologies [109] based on incremental integration and frequent deliveries should be applied. This will help to increase quality and detect problems early in the software life cycle. The systematic use of metrics tailored to the organization is also essential to better track the work and therefore make better decisions based on empirical data [92].
- The use of MDD approaches is useful in the automation of software development tasks, while reducing the amount of errors and better capturing domain knowledge are helpful as regards minimizing communication difficulties and conflicts [105], [118].

2.1.5 Conclusions

The results obtained from this systematic review have allowed us to obtain a global vision of a relatively new topic. It was discovered that every organization has concrete needs which basically depend on its distribution characteristics, its activity and the tools it employs. These factors therefore cause this subject to be extremely wide-ranging, and lead to the need to adapt both the technical and organizational procedures, according to each organization's specific needs.

It should be noted that maturity models such as CMM, CMMI or ISO, which would be of particular relevance to the present investigation, represent only 17% of all analyzed works. The fact that almost all the experimental studies that employed CMMI and CMM applied a maturity level of 2, suggests that the cost of implementing higher maturity levels in distributed environments might be too high. However, there is a need for more studies related to the application of maturity models and metrics to quantify issues related to the process areas. The application of agile methodologies based on incremental integration and frequent deliveries, and frequent reviews of problems to adjust the process have become important success factors. An increasing interest in modeling in software development was found, along with MDA approaches as a means to improve productivity, quality and understanding among members involved in the development process.

The proposals found in the studies analyzed were, in general, mainly concerned with communication [116], which is a central area in GSD. There is a growing trend in solutions related to the use of collaborative tools [59], [91], the integration of existing tools [59], source code control [77] or the use of collaborative agents [59].

A final conclusion, which is central to this thesis, was that the training of human resources is an important success factor [92]. Engineers need to know what problems they will confront when working on GSD and how they can tackle the different challenges. Bearing this fact in mind, the next step was to study how GSD is taught and what skills an engineer who is going to work in GSD needs to have.

2.2 Global Software Development Education

This section presents the findings of a Systematic Literature Review (SLR) in the field of GSD training and education in order to discover the main challenges, strategies and proposals available up to the present day.

Although GSD allows companies to take advantage of the higher availability of a qualified workforce in decentralized zones, their managers frequently complain that recent graduates lack the skills needed to tackle the new problems created by GSD. They argue that their experience is strictly limited to relatively short projects, because education programs do not deal with these subjects at an appropriate level in general [165], and for GSD in particular.

Training these skills is not, however, easy since it necessitates providing students and inexpert software engineers with real experiences that will allow them to develop both technical and non-technical skills [9]. In addition, companies are not always willing to invest their resources in training programs for two main reasons: firstly, because some organizational resources must be relocated, thus putting real on-going projects at risk, and secondly, because reproducing the complexity of real settings is difficult to achieve in educational environments [15], and training might consequently not be successful. If companies were prepared to support this kind of training, instructors would have appropriate tools, methods and course materials with which to teach these skills at their disposal [17], thus providing learners with real experiences and case studies adjusted to the reality of the companies' requirements that would connect theory with practice [18].

Literature shows that several proposals have attempted to tackle this subject: academic courses [166], learning environments [17] and the application of tools presented in real scenarios [18]. In another SLR by Silva et al. [167] the training initiatives shown in Table 9 were found in the field of Distributed Software Development:

Table 9. Training initiatives found by Silva et al. [167]

Type of training initiatives	References
Provision of and training in collaboration and coordination tools	[168], [169], [170], [171], [62], [172], [173], [174], [126]
Training on different cultures/cultural awareness	[170], [171], [175], [176], [173], [174].

It was therefore decided to carry out an in-depth study into the subject of training in GSD education by exploring the challenges and proposals that might be of great assistance to researchers, instructors and practitioners, and which will serve to define an educational environment in the future, based on the rigorous findings of the state-of-the-art.

2.2.1 Systematic Literature Review

In this work, an SLR has been carried out by following the guidelines provided by Kitchenham and Charters [46]. The elements involved in the process are detailed in the following subsections.

Research question

The research question that guided this SLR was:

What are the initiatives carried out in relation to Global Software Development training and education?

The ultimate goal of this SLR is to identify the best procedures, models and strategies employed in the training and education of software engineers. The population of primary studies will also be composed of publications that have been found in the selected sources. These publications deal with the skills required in GSD, and how to deal with the main problems identified as regards teaching these skills.

Sources selection

An initial scoping study was conducted to determine the search string and the resources to be searched. The search string was established as follows:

("distributed software development" OR "global software development" OR "global software engineering" OR "distributed software engineering") AND ("learning" OR "teaching" OR "education" OR "training" OR "simulation" OR "simulator")

Our search strategy consisted of the following decisions:

– **Search sources**: The search chain was adapted to the search engines used, namely:
- Science@Direct (www.sciencedirect.com)
- SpringerLink (www.springerlink.com)
- IEEE Digital Library (www.computer.org)
- ACM Digital Library (http://portal.acm.org/dl.cfm)
- Wiley Interscience (www.interscience.wiley.com)
- AIS eLibrary (http://aisel.aisnet.org)

All the proceedings of the International Conference on Global Software Engineering (ICGSE) were taken into account as they appear in the IEEE Digital Library.

– **Language**: Only papers published in English were considered.
– **Items searched for**: Conference papers, journal articles and workshop papers.

Primary studies selection

The inclusion criteria for determining whether a study should be considered relevant (a potential candidate to become a primary study) was based on analyzing the titles, abstracts and keywords from the studies retrieved by the search to determine whether they dealt with initiatives related to GSD training and education and whether they made any proposals concerning GSD coordination, collaboration and communication problems. In some cases it was necessary to read the entire document to determine its precise scope and relevance.

After analyzing the first iteration of the SLR, exclusion criteria were applied to obtain the primary studies, excluding those studies whose orientation towards GSD problems was not well defined and did not contribute with any significant proposals, despite addressing the training or education of GSD.

Secondary studies selection

The inclusion criteria for the selection of primary studies are listed below:

- Studies that describe GSD courses in university environments or companies.
- Studies that describe real experiences, problems or success factors.
- Studies that propose training tools or environments to carry out the training or education process.

Those studies that fulfilled these criteria were excluded:
- Studies that do not answer the research question.
- Studies that do not contribute with any relevant information or proposals in GSD training or education.
- Studies that do not describe their methods at an appropriate level of detail.

Quality assurance

After the initial selection of primary studies, the quality assessment of the studies was carried out in two stages. First the appropriateness of the studies was reviewed, bearing in mind the inclusion and exclusion criteria, and a more detailed review was then conducted in parallel with the information extraction process.

Table 10. Studies excluded for similarity

Excluded studies	Similar to
[177]	[6]
[178]	[179]
[180]	[13]
[181]	[13]
[182]	[183]
[184]	[19]
[185]	[186]

During this process, the relevance and quality of the studies were verified, also bearing in mind the clarity of their methods and proposals. A number of papers that were published in different sources but were based on the same study were also excluded, in addition to versions of previously published papers, in order to avoid their repetition during the data extraction process. With regard to this, the studies listed in Table 10 were excluded:

The search procedure (see Figure 5) produced 67 relevant studies. Of these, 38 were selected as primary studies. The complete list of primary studies is shown in Table 11.

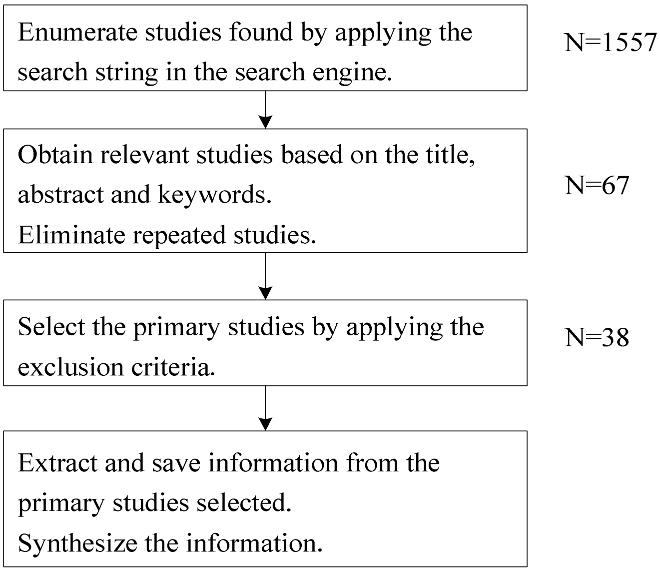


Figure 5. Selection process for primary studies

Table 11. List of primary studies

Reference	Year	Source	Methodology	Reference	Year	Source	Methodology
[187]	2000	IEEE	Case study	[10]	2007	IEEE	Case study
[188]	2001	IEEE	Experimental	[189]	2007	ACM	Case study
[190]	2002	IEEE	Non-experimental	[191]	2007	ACM	Case study
[17]	2002	IEEE	Case study	[192]	2005	IEEE	Case study
[193]	2005	IEEE	Non-experimental	[179]	2008	SpringerLink	Non-experimental
[194]	2008	IEEE	Experimental, Survey	[195]	2008	SpringerLink	Case study, Simulation
[196]	2005	ACM	Case study	[16]	2008	IEEE	Case study

Reference	Year	Source	Methodology	Reference	Year	Source	Methodology
[14]	2006	IEEE	Non-experimental	[13]	2008	IEEE	Case study
[197]	2006	IEEE	Experimental	[166]	2008	ACM	Case study
[12]	2006	IEEE	Case study	[11]	2008	ACM	Case study
[198]	2006	IEEE	Case study	[199]	2009	SpringerLink	Case study
[6]	2006	IEEE	Case study	[183]	2009	SpringerLink	Case study
[200]	2006	IEEE	Case study	[201]	2009	IEEE	Survey
[19]	2007	Wiley	Simulation	[202]	2009	ACM	Case study
[203]	2009	ACM	Case study	[204]	2007	SpringerLink	Case study
[205]	2007	IEEE	Non-experimental	[18]	2009	ACM	Case study
[206]	2007	IEEE	Case study	[207]	2009	ACM	Case study
[9]	2007	IEEE	Case study	[186]	2009	ACM	Case study
[208]	2009	IEEE	Experimental	[209]	2009	IEEE	Case study

Information extraction

An information extraction process for each primary study was applied by using a database with a pre-defined data extraction form containing the following information: title, authors, reference, year, researchers' country, source, number of pages, scope, proposal, organization type (university, company) and company size, processes covered, target population, date of review and methodology.

This process was initially carried out by a novice researcher. A more experienced researcher then reviewed the results in order to ensure that the selected studies matched the research question at a sufficient level of quality for them to be considered as primary studies.

2.2.2 Trends in research on GSD Education

This section analyzes and discusses the content and characteristics of the primary papers found.

The methodology used in the primary studies selected was categorized as : case studies, literature reviews, experiments, simulations and surveys. The non-experimental model (for studies that make a proposal without testing it or performing experiments) was also applied.

Figure 6 shows that the majority are case studies which basically describe experiences in university courses. We did not find any SLR specifically concerning GSD education.

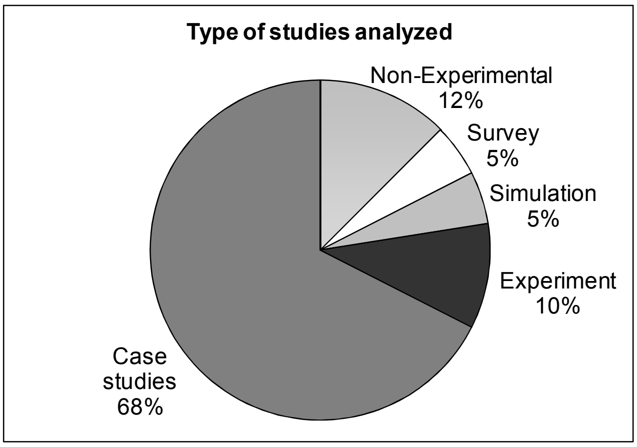


Figure 6. Type of studies analyzed

Figure 7 shows that most of the primary studies are contextualized in a university environment. They describe how groups of students at different locations have carried out joint developments. It is interesting to note that we have also found companies' approaches and studies developed in collaboration between universities and companies.

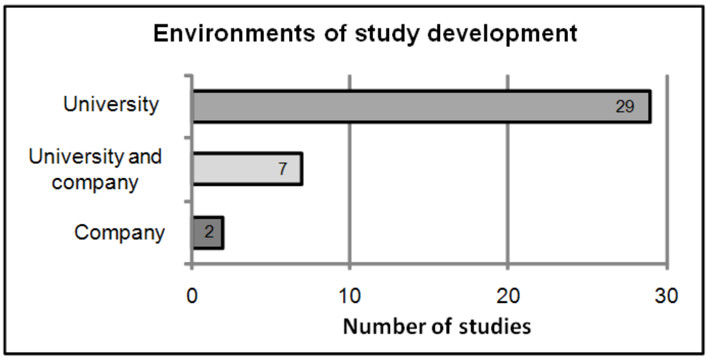


Figure 7. Environments of study development

Classification of Primary Studies according to ISO/IEC 12207

In order to discover the main areas in which studies have focused to date, the primary studies were classified according to the processes of the software life cycle studied, based on the ISO/IEC 12207 standard [49]. The results of the mapping are presented in Table 12. Some primary studies were not classified since their scope was more generic.

Upon considering the number of publications associated with each process category, it is possible to state that the greatest efforts are mainly focused on software construction, software design, requirements engineering and software testing. From the point of view of specific characteristics of GSD, it is possible to conclude that all these processes are particularly affected by communication problems and cultural and language differences. This is therefore consistent with our initial ideas that the training and teaching of these skills have become an important topic. Some of the studies included dealt with more than one process at a different level of detail.

Table 12. Processes considered in the primary studies

Process	Studies
Software construction	[202], [18], [187], [12], [11], [6], [197], [9], [200], [17], [191], [201], [13]
Software design	[202], [187], [12], [6], [197], [9], [200], [17], [191], [13], [10], [194]
Requirements analysis	[202], [187], [12], [6], [197], [9], [17], [13]
Software testing	[202], [203], [187], [6], [200], [17], [205]
Project management	[187], [166], [6], [197], [17], [191], [199]
Requirements elicitation	[202], [187], [12], [6], [179], [13]
Organizational management	[190], [197], [204], [195], [19]
Configuration management	[18], [11], [9], [10]
Quality Assurance	[166], [11], [6], [183]
Documentation	[202], [11], [191]
Problems resolution	[18]
Change requests	[18]
Joint reviews	[166]
Management process	[204]
Risk management	[191]
Software integration	[209]

2.2.3 Results

This section provides a synthesis of the challenges, methods and proposals identified through the SLR, and the most relevant studies are discussed. The section is structured according to the following types of proposals found:

- Learning environments
- e-Learning approaches
- Simulators
- Teaching GSD in the classroom
- Training GSD in the classroom
- Teaching GSD in the company

Learning environments

Various primary studies dealing with learning environments that provide functionalities with which to manage the training of GSD activities and which can be used in companies or universities were found.

iBistro [190] is an augmented space based on the 'learning by doing' approach, thus enabling distributed members to collaborate in the development of software that can be used to learn project management, software development and social skills.

This environment addresses miscommunications and information problems in informal meetings by allowing students to capture structures and retrieve knowledge from the meetings through the use of the audio, video, sketches, notes and the drawings generated. This is supported by a minute generator tool that stores the contextual information, allows the meetings to be represented and facilitates navigation through the database which contains information about the meeting.

iBistro also provides intelligent support mechanisms with which to perform certain tasks such as computer supported group formation, and the ability to effectively find stakeholders and experts in certain areas.

Blended learning approaches were also found, such as that presented by Schümmer et al. [196] for a lab course on Distributed Software Development based on the collaborative virtual learning environment CURE [210]. This lab course uses virtual places (called rooms) for collaboration. These virtual places may contain pages (content), communication channels (such as chat, threaded mailbox, etc.) and users, who will interact with other users 'located' in the same room.

This proposal is based on the Problem-Based Learning approach [211], in which students have to form groups and are encouraged to find a solution to a problem in a collaborative manner, during which they submit deliverables at prescribed milestones.

On the other hand, Bouillon et al. [192] present a platform based on Eclipse. This platform consists of the integration of CURE and CodeBeamer. CodeBeamer is a collaborative platform that offers integrated support in project management, requirements management and code management, and allows asynchronous communication by means of a wiki system. The proposed platform allows students to collaborate during all the phases of the development to produce a large software system.

The work of Berkling et al. [204] is also focused on CodeBeamer. In this case, a framework is proposed which is oriented towards offshoring practices by expanding the functionality of this environment. The authors have basically developed the following Requirements Engineering oriented tools:

- Ibere: an internet-based tool that guides distributed members in the selection of requirements and the estimation procedure.
- TraVis: for traceability management and supporting change management and the visualization and analysis of the dependences among artifacts. This tool helps CodeBeamer to improve awareness by managing a wide range of information through graphical representations and real-time analysis of the project.

A Web-based collaboration platform that eases communication and content management, providing a discussion board, a file sharing repository and a project calendar is presented by Ocker et al. [186]. Instructors can add training modules, and students can access their description along with their instructions, milestones, and deliverables. The students use this platform to work with their partners in order to achieve the module's scopes.

Jazz [18] is a collaborative development platform that integrates several functionalities to support the software life cycle, such as: source code repository, chat, web interface, reports generation, and work items. Students can generate work items containing all the relevant information related to the resolution of a problem along with the associated chat conversations.

Toyoda et al. [199] detail a tool that supports problem analysis and helps students to improve their problem-solving skills. Users can add labels to problem nodes of the project that will be helpful in resolving the problem.

Swigger et al. [200] describe a computer supported collaborative tool for teaching distributed teams through the use of collaborative tools (including chat, a scribble tool, an application sharing tool, graphics tools for designing UML documents, etc.). The authors have also developed course management software that helps instructors in tasks related to the administration of groups and the collection of information concerning the students' actions, thus providing a means to evaluate them.

This study also presents a Web portal developed with the aim of helping students to manage the groups and projects in which they are involved. Students use this means to share their personal information (such as name or email address) with their partners, and they can also access their partner's schedule in order to agree possible meeting times for the projects they have in common.

Favela and Peña-Mora [188] propose a framework which is useful for dealing with some of the difficulties in GSD. It basically consists of the following set of tools:

- Project scheduling and tracking tool: to help the instructors in their tasks.
- Configuration management tool: that manages change requests by filling out and submitting a form.
- Technical review tool: including inspections and walkthroughs.

This study also introduces the idea of the team contract, in order to standardize the project's daily process and avoid missed meetings and other problems caused by the former informality of the process. The team drafts a contract to establish rules concerning communication, response times and the authority that is allowed to manage these kind problems. This same idea is also used by Ocker et al. [186].

Swigger et al. [200] describes a collaborative environment in which to teach distributed teams through the use of a set of tools (including chat, a scribble tool, an application sharing tool, etc.). This study also presents a Web portal that helps students to manage the groups and projects in which they are involved. In this category, ClockingIT [212] is a project management and collaboration customizable platform that provides chat rooms, instant messaging, a built in wiki and discussion boards.

Corder and U [213] explore the interaction with avatar-based humans through the integration of Second Life in virtual collaborative projects, which allows collaboration skills and intercultural competence to be trained by performing real-time realistic activities. In a similar vein, Teamlink [214] is a Collaborative Virtual Environment based on configurable avatars in a virtual 3D world which supports icebreaking activities with the aim of establishing trust between virtual team members by using asynchronous communications.

Although these environments can help students to learn concepts and real problems that may occur during collaboration in GSD, developing the soft skills required cannot be achieved in an accurate manner by these means. The difficulty of interacting with distant participants is also present in these kinds of environments.

e-Learning approaches

The application of e-learning approaches has also been found, which, in contrast to learning environments, consists of web-based applications oriented towards delivering online courses.

OAS!S [207] is a virtual learning environment created by customizing the WebCT Vista / Blackboard platform. Its use is broadly extended in many universities. It allows the use of discussion boards, mail systems, chat and content management, which can be very helpful, principally as regards improving communication skills.

In this respect, the open source platform learning OLAT (Online Learning And Training) [207] similarly supports forums, chats, file sharing, and mailing system, offering support for various e-learning standards such as IMS or SCORM.

Finally, Toyoda et al. [199] present the development of a Web-based e-learning course dealing with project integration and quality management (including project chartering, monitoring, and controlling), along with time and cost management through the scheduling of tasks and activities using methods to control their cost.

Although e-learning platforms are a good means to establish communication with students and to provide them with materials, their use is not greatly advantageous from the point of view of the skills and abilities that can be trained.

Simulators

This SLR has led us to realize that some primary studies propose simulators which permit the training of specific skills, with the advantage of reducing the costs and risks involved in performing that task if it were to be performed in real environments.

In this respect, Setamanit et al. [19] proposes a simulator to study different ways in which to configure GSD projects. The configuration includes parameters related to task allocation, studying phase-based, module-based and follow-the-sun allocation strategies. The simulator computes the project duration for each configuration, thus allowing the student to study the impact of different GSD factors, such as distance, culture, language, trust and time zone, on project duration. This simulator could therefore be used not only to help managers to find the best configuration settings in order to improve their project performance, but also as a training device in the new challenges created by GSD, by helping managers to better estimate the values of the parameters that make a project suitable for GSD, or which sites should be included in the project, how the work should be divided, or which tools are effective in these environments.

The kind of skills and abilities that could be trained by means of simulation might result in a competitive advantage in comparison to the aforementioned methods. However, it must be concluded that the literature in this field is still in its infancy, and that no significant proposals exist for the training of most of the GSD skills required.

Teaching GSD in the classroom

Traditional theoretical classes are also commonly found in literature as a response to the needs for adjustments in software engineering education [10]. In this respect, many studies coincide in highlighting the necessity for joint courses with different universities, so that students can benefit from a wide set of knowledge and experiences [11]. However, a commonly reported problem with this approach is related to the difficulty of attaining an appropriate level of coordination and collaboration with the different institutions [14].

Several studies that describe learning course approaches were found, such as that presented by Bmegge et al. [187], which consists of an experience in collaboration with other universities with three distributed software engineering project courses. The approach not only includes theoretical classes, but also seminars and laboratories which are presented through the use of a collaborative infrastructure.

A European Masters program on Global Software Engineering which involved several universities from different countries was presented by Lago et al. [10]. This program enables the technical and cultural dimensions of GSD to be taught and is focused on the areas of: software architecting, real-time embedded systems engineering and web systems and services engineering.

Liu [193] provides details of a software engineering program with which to incorporate collaborative global software development processes into universities and industrial software companies and also to enhance distant learning programs.

The Masters course presented by Lago et al. [194] takes into account the fact that students from different universities have different backgrounds, skills and experience, and that it is therefore important to analyze the students' prior education and characterize them in order to design the course.

As shown in the following section, some studies also combine teaching classes with practices and actual developments.

Training GSD in the classroom

"Learning by doing" is the most common approach suggested by the primary studies, since developing GSD skills requires putting theory into practice by using tools and methods to tackle typical problems found in real environments.

Universities that teach GSD tend to organize joint student developments, collaborating with universities from different countries. In these cases, the students communicate by using email, telephone and instant messaging [189], and this interaction allows them to learn from others students' skills and cultures. Their involvement in real development experiences is therefore both positive and beneficial for their curriculums [9], given that they tackle processes with a close similarity to those applied in industry [17].

Burnell et al. [17] present an experience in which several universities from the same country participated. Distant team members collaborated by documenting each task and sending it electronically to the other university groups. The process defined was sufficiently flexible to allow modifications to be made to the schools' task assignments.

Rusu et al. [202] present a case study of a course in which virtual teams from different universities collaborated to develop a project for a real customer that profited from this learning process. However, this study was limited to the interaction between only two universities, both of which were located in the same country, and they could not therefore deal with all the problems caused by internationalization as regards language, time and cultural problems.

The experience presented by Richardson et al. [197], in which three universities from different countries were involved, identifies scheduling as one of the most common difficulties for engineers when they are full-time developers. One of the important subjects mentioned in this study consists of training communication and informal skills and learning how to work effectively with a team and react quickly to changes in requirements, architecture and organization.

Petkovic et al. [6] combine a class which teaches software engineering methods and processes with the development of a real project in a setting designed to simulate a small company. Students were divided into small groups (in some cases distributed throughout different countries), and had to collaborate in order to develop a complete working application.

Swigger et al. [207] details two projects involving students from four countries that were developed with the aim of studying a team performance model that was used to measure certain factors that affect collaborative work.

In Gotel et al. [183], the authors present a course on quality assurance from three universities from different countries that have worked together on the development of a software system. This practice allowed students to interact with partners from different cultures, and also to play different roles in the project. Since they collaborated with partners from different backgrounds, skills and ways of thinking, their learning experience was richer that it might otherwise have been. This study, which is complemented in [13], makes some recommendations oriented towards project planning in GSD education based on the students' experiences. The interaction was carried out by using several types of communication tools [13]:

- Mailing lists, emails and chats.
- Wikis: Each sub-team in the project maintained a wiki containing all the documents and artifacts that they produced.
- Blogs: in which the students reflected their weekly progress in the project.

The authors also suggest that it is easier for course instructors to play the role of project managers. This permits them to take advantage of their position in order to lead students through the processes and, for instance, prevent them from experiencing certain coordination problems.

In [12], the corresponding authors report the strategies applied in a course on requirements engineering in three different universities, in which a web-based inspection tool named IBIS was used in an educational environment. These strategies were oriented towards learning the skills needed for international teamwork, along with the use of specific tools for remote communication.

Some outsourcing experiences for students reported by Honig and Prasad [189], courses oriented towards eXtreme Programming in distributed environments [166], and other experiences referring to offshore software development between undergraduate students at different locations replicating a client/vendor relationship in a virtual setting, as presented by Adya et al. [191], should also be mentioned.

A similar idea is reported under the name of Open Ended Group Projects (OEGPs) which is a supplement to conventional teaching approaches that can be used to develop the essential skills needed in the 'real world' [215]. One noteworthy experience is that of the OEGP Framework presented by Daniels et al. [216], which provides a flexible educational course which evolves with the assistance of an action research program that allows improvements to be made through a combination of learning theories and stakeholder input. The introduction of an external mentor to support the project leaders proved to be helpful in improving the quality of the projects' results [217].

Finally, RUNESTONE [218] is an International Student Collaboration initiative which consisted of a project-centered course in computer systems during which students were involved in the development of a pedagogical project involving two universities in different countries. Other works related to this initiative can be found in [219] and [220], but are not explained owing to space constraints.

Despite the appropriateness of this kind of training in realistic experiences, some skills or abilities cannot be accurately trained, as it is not always possible to reproduce certain situations (i.e., the occurrence of a specific cultural problem among two participants). Moreover, some drawbacks must be confronted: the amount of work required to organize and prepare this kind of practices, the difficulties or the interaction between students from different locations and clashes as regards the objectives of the different institutions.

Teaching GSD in the enterprise

Although the topic of teaching GSD within a company has not been commonly reported in the primary studies, some related experiences have been found. The most remarkable example is reported by Prikladnicki and Pilatti [16], who present a training initiative in a multinational organization that applies GSD. A training course related to concepts and best practices of communication, trust, cultural differences and coordination was developed. After the completion of each project, the students generated a document containing the lessons learned.

The higher level of availability of an experienced workforce appearing in these environments makes the application of the concept of learning networks [183] possible. Since instructors cannot be experts in all the GSD areas and cannot consequently cover every topic, learning networks provide a new way in which to perform the learning process based on the experience of a multidisciplinary set of trainers.

An example of a learning network is described by Lutz [206]. It consisted of a team of trainers from a company who were experts in specific software development activities and who, in this case, combined their training activities with their work as engineers, project managers, quality managers, etc., in that company. Students were therefore able to interact with them and take advantage of real work experiences by maintaining contact with professionals who were carrying out day-to-day work.

2.2.4 Conclusions

In this section, a Systematic Literature Review method has been applied in order to analyze the literature related to GSD training and education. The results obtained have allowed us to depict a vision of the challenging factors and strategies applied in the teaching and training of the new skills required. These results have led us to the following conclusions:

Conclusion 1. There is a growing interest in GSD training and education, since GSD has grown in recent years and training has become essential [10], [11].

Conclusion 2. The teaching and training of GSD must be supported by practical experiences through which students can learn by doing [189], [17].

Conclusion 3. Simulating the complexity of real environments is difficult for universities, and the different timetables of the students make it difficult to coordinate training projects [194].

Conclusion 4. It is not possible for instructors to cover all the stages and problems of GSD [206], so any initiative should be focused on a specific field.

Conclusion 5. Students involved in GSD training programs usually experience a lack of motivation, schedule problems and communication difficulties [196], and this is greater still when cultural and language differences are present [188].

Conclusion 6. Particular training scenarios and learning environments require specific tools for communication, collaboration and document management. An appropriate selection of tools is therefore a key aspect [12].

According to the results of the literature review, the educational proposals found have been grouped in categories and are represented in Figure 8. The quadrant shows the cost of the proposals in the horizontal axis, in terms of money, risks and resources requirements, while the vertical axis represents the fidelity of the approaches in terms of their accuracy and appropriateness for training GSD skills.

Literature shows that theoretical classes, both in universities and companies are those that require less resources, as they are only based on the study of learning materials and verbal explanations by a lecturer or trainer [194]. In the case of training company members, they cannot attend their productive job when they are attending these courses, which may make them more costly [183]. Multi-university courses involving students from distant universities are more realistic; however, they are still far from capable of reproducing real GSD settings in an accurate manner. Moreover, reproducing GSD environments in educational settings is more costly since it requires a great deal of coordination, infrastructures and the participation of several lecturers [202].

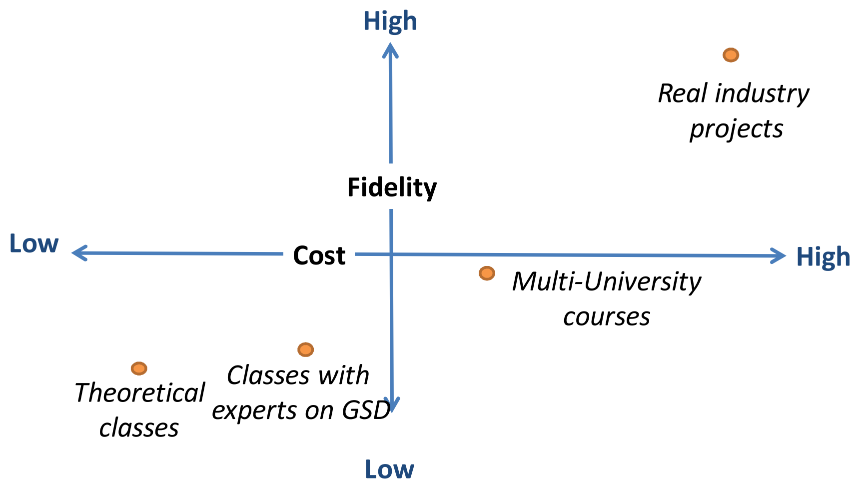


Figure 8. Current approaches in GSD education

The most accurate method consists of placing participants in real industrial projects. However, the high costs in terms of risks signify that this method is not often applied. Companies rarely invest their resources in training programs, since this may entail some risks for on-going projects owing to the need to reallocate certain resources. Primary studies generally focus on improving the participants' collaborative skills and are oriented towards a specific stage of the software development, such as the requirements stage or software construction [221]. In accordance with these results, the framework presented in this thesis intends to cover the fourth quadrant (low cost, high fidelity) by using simulation with the aim of diminishing the problems associated with traditional teaching methods.

After studying how GSD is taught and trained in universities and companies, the next step in this research consists of studying the knowledge and skills that should be acquired by students and engineers in order to cope with the new challenges created by globalization. One of the most important subjects consists of improving formal and informal communication skills, which is not part of traditional software engineering education at this time.

In other research fields, the resolution of this kind of difficulties has been addressed by means of games that simulate realistic situations [222]. A representative example of this is SimSE [223], which is an interactive game that illustrates particular Software Engineering (SE) processes to students. In this respect, after presenting the skills required in GSD, learning environments for communication and collaboration are also described in this chapter.

2.3 Knowledge and skills required in GSD

Current Software Engineering education is generally focused on the study of the contents that are included in the bodies of knowledge SWEBOK [224] and SE2004 [225] and PMBOK [226], which basically define the general and technical knowledge required in software engineering activities. These models present a global view of the skills that present-day software engineers should have when they finish their degrees, and are useful for detecting what additional competencies should be acquired in relation to GSD practices [227], [228], which are mainly related to the specific needs of communication, coordination and collaboration, along with the attitudes and knowledge that they may need in globalized scenarios and which were not required to such a great extent in the traditional co-located development [189].

In GSD projects, the use of a variety of communication media, such as e-mail, instant messaging, telephone or teleconferences is the rule [229]. There is growing empirical evidence that personality factors impact on team performance [230]. It has been found that participants with poor language skills tend to prefer using text-based communication media as opposed to ad-hoc audio conferencing [231]. Virtual teams interacting in GSD environments therefore tackle complex difficulties that are not the rule in traditional co-located environments [232], [233]:

- Interaction with multicultural and multilingual members.
- Interaction with multidisciplinary teams.
- Use of different dialects of the same language: different spelling, accent and use and meaning of words [234].
- Use of different standards and terminology [235], [233].
- Misunderstandings and high response times [232].
- Fewer opportunities for communication. Loss of non-verbal cues and limited informal conversations and, in consequence, inequality of participation.
- Difficulty in building up consensus, trust and team awareness ([7], [236]), with the consequence that it takes longer to reach decisions.
- Less cohesiveness of the team [237]. Difficulty in sharing ideas, artifacts or components needed during the process.
- Conflicts among locations and difficulty in reaching agreements.
- Negative impact of fear of interacting with virtual teams [236], [174] as a consequence of the conflicts that are inherent to this kind of development.
- Lack of trust in other team members: The addition of distance makes it more difficult to establish reciprocal faith in others’ intentions and behavior.

- Feelings of isolation and indifference in virtual teams [232].
- Use of communication and collaboration technologies [232] [235], which require time to be learnt [237].
- Failures in communication [238]: misuse of communication tools, lost messages or technical problems can create conflicts. A related problem arises when senders of an email expect a response, but the receiver does not consider it necessary to respond, which may lead to time loss and a lack of trust [235].

It is therefore necessary for software engineers to have additional technical skills as regards using the collaborative tools that are available in the fields of software requirements, design, construction, testing, maintenance, configuration management, software engineering management and software engineering processes. The general skills principally required by virtual team members are summarized in Table 13.

Providing training in these skills is a challenge, and more so if we consider that cultural and linguistic differences significantly influence the problems that can appear on GSD. A description of how cultural differences affect GSD is provided in the following section.

Table 13. Skills required in GSD

General skills	Detailed skills
Communication	- Appropriate level of the language employed in the organization, along with knowledge of the organization's culture [34]. - Knowledge of communication protocols and customs [9], [13], [239]. - Ability to effectively communicate with a multidisciplinary group using the same terminology through a second language [240], [17], [241] in addition to knowledge of common expressions [242]. - Ability to detect missing pertinent information and the interlocutors' misinterpretations [242], [243]. - Knowledge of negotiation skills and contract writing in a common language [12]. - Ability to be explicit and descriptive during communication [233], attempting to minimize ambiguity, since the lack of non-verbal cues makes comprehension more difficult. - Ability to speak slowly and clearly using simple language in order to ensure clarity and limit ambiguity [234]. - Informal communication, creativity and improvisation skills [197].
Tools and processes	- Ability to communicate effectively by using computer-mediated communications [186], [9], [12], [233]. - Knowledge of methods, data, and processes required in a distributed project [16]. - Use of knowledge, artifacts sharing, and control version tools and collaborative tools generally employed in GSD [190], [191].
Teamwork	- Ability to think from the perspective of the other side [188]. - Skills to gain the team's confidence and trust [242], [244], [245]. - Leadership skills and interpersonal dimensions [199], [246], [247], [248]. - Time management skills [206], [6]. - Ability to motivate and influence the team. Use of rewarding structures that can make the team members feel more motivated [235]. - Ability to give feedback and praise in an appropriate manner. - Soliciting and responding to feedback [235]. - Analytic competence [243].
Conflict resolution	- Dealing with difficult and conflictive situations [235]. - Early diagnose and rectification of problems within a virtual team [7]. - Interpersonal conflict resolution [186], [242], [227], particularly affected by cultural differences and communication problems. A positive attitude and motivational skills are necessary to confront these problems. - Ability to formulate criticism/praise carefully, since it may not be received in the intended manner [249], [250]. - Ability to raise counter-arguments in order to avoid exacerbated attribution of errors [7].
Miscellaneous	- Self-learning capability [227]. - Estimation of tasks and prioritization [227]. - Critical analysis of information that can be distributed in various means and be defined in an ambiguous manner [227]. - Information management, management of ambiguity and uncertainty, synthesis, analysis and critical reasoning skills [9].

2.3.1 Cultural differences in GSD

Culture can be understood as customs, beliefs and a learned way of life that is shared by a group of people with similar ethnic origins, professional status, religious convictions, gender, area in which they live or shared interests [238]. Cultural values are ingrained in people, making them behave a certain way automatically. It is sometimes even difficult for people from a certain culture to understand the behavior of a different culture. Since people assume that their assumption is correct, a wrong interpretation tends to remain unrecognized [251]. This can create feelings of discomfort or confusion when people from different cultures interact, as people tend to automatically believe that their way is the right way [238].

Acquiring **cultural awareness** involves realizing that people belong to a diverse range of cultures and that they will hold different values, behave in different ways and have different ways of viewing the world [238]. According to Hofstede [252], cultural awareness is a first step towards inter-cultural understanding, and the following steps are described:

1. Awareness: Being aware that there are differences in behavior involves accepting that there are no better or worse ways of behaving and that one's own behavioral routines are not superior.
2. Knowledge: Learn which differences in behavior exist, being aware of their importance and possible consequences in the case of behaving in a different way.
3. Skills: Training of communication skills for a specific culture.

For Endrass et al. [251], Virtual Agents are a powerful medium by which to cover the first two steps of learning inter-cultural communication: "By observing their behavior, a trainee can learn about a culture's specific differences without the need to embarrass real humans by watching them, or reading complicated explanations in textbooks".

According to Eden-Jones [238], an important method that can be used to gain cultural awareness is that of acquiring knowledge about other cultures and the development of a range of skills required to effectively interact with colleagues, clients and members of a community. The objective of this is to improve relationships based on cultural diversity, communicate effectively with culturally diverse people and resolve cross-cultural misunderstandings. Culture cannot easily be changed, and cultural training must therefore be focused on creating cultural awareness and improving these skills. Organizational culture (policies, procedures, and customs) could also impact on performance [253].

Cultural differences involve: the language spoken, dress, religion, attitudes to family, attitudes to work, roles of individuals in society, food, holidays and celebrations. Several studies have made an effort to classify these differences:

- **Trompenaars and Hampden-Turner** [254] compare culture to an onion made up of layers that can be peeled to be understood. They outline seven dimensions of culture: universalism versus particularism, individualism versus communitarianism, "specific" versus diffuse, affective versus neutral, achievement versus ascription, sequential versus synchronic, and internal versus external control.
- **Hall** [255] believes that culture is equated with communication, which is made up of three elements: words, material things and behavior. This author defines seven relevant concepts with which to study national and corporate culture: speed of messages, context, space, time, information flow, action chains and interfacing.
- **Hofstede** [256] defines a classification focused on the values and culture of computer professionals, considering five value dimensions in which countries differ: power distance, uncertainty avoidance individualism/collectivism, masculinity/femininity and long-term/short-term orientation.
- **House et al.** [257] provide a more recent classification focusing on the culture and leadership in 61 nations, and defining the following dimensions:

- **Uncertainty Avoidance**: Degree to which the individual feels 'comfortable' in new situations. Members tend to avoid uncertainty by relying on social norms, customs, and bureaucratic practices to deal with unpredictable events. A common training simulator would consist of involving a student with a culture that presents a low degree of uncertainty avoidance in an interaction with Virtual Agents that show a strong resistance to change and to accepting their ideas. The student should, therefore find a way in which to politely persuade them.

- **Institutional Collectivism and Ingroup Collectivism**: Degree to which a community encourages and rewards the collective distribution of resources and collective action. This includes factors such as the individuals' loyalty and cohesiveness. For example, a student uses a simulator to train in how to interact with Virtual Agents with low institutional collectivism that tends to be independent and not used to working in groups.

- **Assertiveness**: Degree to which individuals are dominant, confrontational, and aggressive in social relationships. A student from a highly assertive culture would need to learn to cooperate and have warm relationships by using a simulator that would allow him/her to interact with Virtual Agents of low assertiveness.

- **Future Orientation**: Measures time aspects for the satisfaction of needs, including how the students will engage in future-oriented behaviours such as planning, delaying gratification and investing in the future. For example, a student with high future orientation interacting with low future individuals should learn how to provide gratification as soon as possible.

- **Human Orientation**: Degree to which individuals encourage and reward individuals for being fair, altruistic, friendly, tolerant, generous and supportive of each other. A student with low human orientation would need to train with a simulator in which s/he would interact with Virtual Agents in order to learn to show interest and affection.

- **Performance Orientation**: Degree in which members are encouraged in their tasks and rewarded for their achievements. Low performance orientation students prefer to use a more indirect style of speech while high performance members tend to be more competitive and direct. The instructor could, for example, decide to train a student with a direct communication style to communicate with low performance members in a cordial manner in order to improve team trust and motivation.

- **Power Distance**: Degree to which less powerful individuals agree that power should be shared unequally. A training activity for this dimension would be focused on establishing good relationships between participants who understand authority in different ways. For example, a student from a culture with a low power distance might find it difficult to accept authority when interacting with colleagues from high power distance cultures, in which classes are well differentiated. Or vice versa; a student from a high power distance culture would need to learn to interact in a less hierarchal manner.

- **Gender Egalitarianism**: This is the degree of gender equality. Simulations could teach a woman how to interact with cultures in which there is a great separation between male and female roles, while men from these cultures would learn how to interact with women with regard to their position and culture.

Culture and communication management is therefore essential in GSD [258], and training in specific cultural issues is required [259], [260]. Some representative cultural problems that occur in GSD:

- Team behavior and the manager's authority is usually understood in different ways in different cultures [261], so people from one culture may appear to be politically incorrect to people from another culture during the interaction [262], [234], [263], [42].
- In some cultures, people are open and free to express their real opinion, while in other cultures it may be considered impolite, especially when it entails disagreeing with somebody [242].

- In some cultures, interrupting a partner's communication is interpreted as showing interest in the conversation, while in other cultures, this behavior is regarded as very rude [251].
- Participants do not speak during team discussions until invited to do so, since this is not considered polite in their culture [250].
- The use of direct or indirect style refers to the way in which people reveal their intentions. Being too direct may appear very rude in some cultures, while being too indirect implies an excessive deviation from the conversation for other cultures [249].
- Succinct/elaborate style refers to the amount of information provided during communication. Some cultures may experience problems as regards understanding silences or fully understanding a succinct message. A style that is too elaborate may seem too boring to other cultures [249].
- People from some cultures tend to say that they have understood something when they really have not [250]. This can cause misunderstandings that may be detected too late. Team members should be asked to summarize points in their own words or use "open ended questions" to confirm that they understand, otherwise the answer will always be "Yes" [250].
- The use of a formal/informal communication style may also differ in different cultures depending on the situation [249].
- The "Mum Effect" [264]. This refers to the tendency to cover up critical information or to distort negative news by presenting it as more positive information. However, receiving timely and accurate negative information may be critical for the project's success in GSD [265]. It is therefore necessary to know how to communicate bad news in an appropriate manner, and confronting the fear of being punished or criticized [266].
- The 'Deaf Effect' [266] refers to the reluctance to hear bad news in some cultures, where people tend to ignore problems and warning signs. This occurs because they are not aware of the seriousness of the problem or they think that it is not their responsibility.
- The humor and jokes of people of some cultures may prove to be offensive or incorrect for other cultures [242].

By considering the cultural dimensions proposed by House [257], it is possible to establish which kind of cultural problems may appear in GSD, and measure the degree to which they are affected by the different cultural dimensions set out by House. By establishing this relationship (cultural problems/cultural dimensions), it is possible to quantify the degree to which each cultural problem will appear during the training process by considering the cultures of the participants in the virtual meetings.

2.3.2 Linguistic differences in GSD

Linguistic differences can be classified as a kind of cultural difference. In GSD, the use of a non-native language entails particular problems such as the overuse of some verbs of a high semantic generality (do, have, make, put, etc.), the use of false friends (incorrect use of words that look similar in two languages but differ in meaning) and grammatical inaccuracies that can lead to misunderstandings. Taking into account these problems, typical recommendations when communicating with people from different cultures are [249], [250]:

- Formulate criticism/praise carefully, since it may not be received in the intended manner.
- Avoid slang, colloquialism, jargon, acronyms and metaphors.
- Avoid humor, since jokes are rarely understood in a different culture.
- Use simple language.
- Keep to the point and be quite descriptive.
- Speak in the writer's view and change between informal/formal writing according to the situation.

Some of the typical written linguistic problems when non-native speakers interact using English are [249].

- the incorrect use of "false friends" (when a word looks or sounds similar in two different languages, but differs in meaning)

- the overuse of certain verbs of high semantic generality (e.g. do, have, make, put, take)
- the incorrect formation of conditional clauses
- the incorrect use of verbal tenses
- the avoidance of passive forms
- the absence of the third person –s

This kind of problems, can sometimes lead to misunderstandings, and participants should on the one hand improve their linguistic skills and on the other hand know the kind of problems that other non-native speakers may have.

2.4 Learning Environments on communication and collaboration

Given that communication and collaboration are the main topics that need training in GSD, this section focuses on learning environments that provide this kind of training. One of the most common approaches found consists of applying 3D environments as a space in which to interact with distant learners. These approaches offer digital landscapes in which users can share information by interacting with each other, usually considering full body immersion [267]. Virtual collaboration trains competencies such as effective communication, intercultural awareness, and the ability to build trust and understanding [213].

The use of avatars or embodiment is common as a means to interact in these environments [268]. An example of 3D virtual environment applied with collaborative learning objectives is presented by Hasler et al. [269] in which the authors studied the outcomes of global virtual teamwork in an international student collaboration project focusing on usability and sociability issues in collaborative work with 3D virtual environments.

The use Second Life as a means to provide out-of-class intercultural experiences is also extended. Usually, after interacting in virtual worlds with other students, participants critically analyze their experiences, applying theoretical frameworks to intercultural collaboration [213], [270].

Interactive drama has also been applied in many related fields. Interactive narrative and drama engines are used to situate the learner in a story in which he or she is one of the main characters. A good narrative engine has to dynamically maintain a storyline and adapt to the learner's intervention [271].

Games and **simulation** have been applied in many fields of software engineering since these approaches are among the most motivating for students. The SESAM project [272] is a representative example whose intention is to investigate and compare different strategies for software development. Students use a textual interface in which they read and type text for training in project management activities. M. Samejima et al. [273] address situation-dependent scenarios, in this case in order to simulate project management activities, specifically covering the generation of scenarios for the progress management phase. SimSE [223] is a game in which the player takes on the role of project manager who manages a team of developers in order to successfully complete an assigned software engineering task. A computational model based on simulation with which to teach cultural differences is presented by Jan et al. [274] , who focus on non-verbal behavior clues, such as proxemics and gaze.

The learning of international collaboration has been addressed by many authors in different contexts [275], [276], [277], [278], [259], [219], usually involving students from different locations, and has proved to be helpful in cultural training when combined with adequate technology [279]. A number of collaborative applications have also been developed in order to provide this kind of training with support [280], [281], [282].

In the multicultural field, ALELO [283] is a platform that enables learners to engage in spoken conversations in foreign languages by interacting with virtual participants. It integrates intelligent tutoring in order to help learners acquire communication skills and cultural awareness. The dialogs conducted are based on a framework for dialog modelling, which is based on multi choice selection.

In this kind of environments, learning takes place most easily when the students actually need the knowledge of how to do something for a reason. Feedback reception has been proved to be an important factor as regards enhancing students' knowledge and skills [284].

2.4.1 Virtual Agents

One of the findings after studying learning environments is that the use of avatars or virtual participants is extended in the field of communication and collaboration training. Virtual Agents have been used with good results in learning environments when teaching intercultural competencies [251]. They would therefore appear to be useful as regards fulfilling the educational objectives pursued in this thesis. It is thus necessary for this work to study how Virtual Agents have been designed and implemented in training platforms. The advantages of using Virtual Agents in the definition of a training framework on GSD are:

- the repeatability of the training scenarios.
- emotional distance from virtual characters,
- the opportunity to over-exaggerate or generalize behavior.
- they can operate in an autonomous manner, thus minimizing the costs of involving human training-partners.
- they avoid scheduling and coordination difficulties with other partners.
- they monitor capabilities.
- scalability: it is easy to adapt the environment to a different number of learners.

A relevant finding found in literature shows that if the benefits of learning are to be maximized, timely and thoughtful guidance is necessary in order to understand and take advantage of the interaction with Virtual Agents [285], [286]. The work presented by Core et al. [286] focuses on training negotiation skills by using Virtual Agents that have a certain cultural background. They apply a technology called Explainable Artificial Intelligence (XAI) which provides an interface that allows users to "rewind" a simulated scenario to a time point in order to discuss and show rationale for certain actions. The tutor employs XAI as a teaching tool so as to investigate player errors in a dialogue. The dialogue is conducted using the learner by choice selection, thus allowing users to gain language skills and knowledge about cultural non-verbal behaviors, such as gestures

The use of embodied **conversational agents** [287], [288] is common in learning environments. Endrass et al. [251] present an approach with which to simulate differences in the management of communication for the American and Arabic cultures. These authors apply Virtual Agents as a demonstrator in order to make it possible to learn about different cultures. Iacobelli et al. [289] similarly present a virtual peer that shows different verbal and non-verbal behaviors. Their work suggests that Virtual Agents are an effective tool in educational applications.

Kavakli et al. [290] present an embodied conversational agent which provides the aboriginal people of Australia with personalized counseling services. In this case, the agent plays the role of a sociologist in order to advise on strategies that can be used to overcome addictions. An expert system stores an expert sociologist's knowledge and the inference engine applies certain rules in order to generate a verbal output that animates the conversational agent. The same authors have developed a narrative engine with which to control the dialog between the user and the system which is based on chained rules. This proposal also uses animation by giving the characters full body movement, which allows personality attributes and emotions to be simulated. In this respect, some studies aim to create affective conversational agents through the introduction of gestures and facial expressions in order to simulate the expressivity of real communication [291].

2.5 Limitations

Despite the rigorous results that can be obtained using Systematic Literature Reviews, there are also some limitations. These limitations apply to both of the Systematic Literature Reviews presented in this section.

The main limitation concerns the search keywords applied, as on some occasions authors may have used terms that were not considered. There is also a restriction as regards both the number of source engines that were used and the selection of the inclusion and exclusion criteria.

Finally the period of time in which the studies were conducted is also a limitation, as new studies may have appeared. This limitation was confronted by studying other Systematic Literature Reviews on GSD up to the present.

These limitations could impact on the understanding of the state-of-the-art, since some papers that might be within the scope may be missing.

2.6 Conclusions

After analysing the results of the literature review, it has been found that the trend of applying GSD has led to the need for industry and universities to provide training and education in the difficulties that globalization entails, principally those concerning communication and collaboration with multicultural teams.

However, it has also been discovered that this kind of training is not commonly applied, usually because of coordination difficulties and the amount of time and resources needed to provide accurate training. The following limitations of the current GSD teaching methods have been found:

- Limitations in the number of people that can participate and problems in finding an appropriate multicultural group [206].
- Difficulties in establishing relationships with other universities or institutions and coordinating the work [197].
- Students depend on their colleagues' availability and skills [10].
- Students do not have common experiences. Everyone plays a different role and it is difficult to evaluate them fairly [183].
- It is difficult to provide opportunities for self-reflection and feedback. The repetition of tasks under the same conditions is not always possible [11], [14].
- Lack of rigorous orientation towards cultural and communication difficulties [11], [14].

The effectiveness of the current approaches is not therefore promising, and these approaches are not commonly viable. As a consequence, students usually become only familiar with theoretical knowledge.

The study of learning environments when applied to the fields of communication and collaboration, has led to the idea that simulation has the potential to be applied in GSD training. On the one hand, it can be a flexible tool for use in educational environments, thus minimizing coordination difficulties. On the other hand, using Virtual Agents to conduct these simulations may provide a means to reproduce settings that it would be difficult to generate with the current educational methods. This kind of simulation would allow us to effectively focus on specific abilities, skills, or areas of concern in GSD, and would permit exaggerated specific behaviours or actions in order to promote learning. Other advantages such as scalability, easy management and monitoring, low requirements as regards preparation and infrastructure are deciding factors for pursuing this work. On the other hand, there are also some disadvantages of using simulation. As an example, some people do not like using learning platforms and they prefer to interact with real peers. In this respect, providing prompt feedback may be a success factor as regards avoiding a lack of engagement and promoting effective learning [284].

As a result of the analysis of these results, we discovered that it was necessary to define a training framework that would allow the abilities required in GSD to be trained. The focus of this work is the definition of such a framework based on simulation in order to avoid some of the difficulties found in literature.

Chapter 3

Research Methodology

The research methodology applied in this thesis involves the participation of students, practitioners and researchers. From the perspective of this study, it has been necessary to use empirical methods since they allow the researcher to incorporate multidisciplinary and interdisciplinary factors that frequently arise, such as human issues, communication difficulties, quality of processes and products [292].

Qualitative and quantitative methods are used in empirical research as a means of making direct observations that reflect experiences with methods, tools and techniques in a way that relates more to the real world than other research approaches [293]. Quantitative and qualitative methods are applied in this research in order to gather data about the use of the framework and its effects on the scope users (students and GSD practitioners).

The most suitable research method considered as regards achieving the research objectives in this context was mixed methods research, in which both quantitative and qualitative methods are combined. This chapter describes the method that has been applied. The first section explains mixed methods research, while the specific strategy that has been applied in the case of this research is explained in the second section.

3.1 Mixed Methods Research

Mixed methods combine qualitative and quantitative research approach elements (e.g., the use of qualitative and quantitative viewpoints, data collection, analysis, inference techniques) for the broad purposes of breadth and depth of understanding and corroboration [294].

Qualitative research is an inquiry method whose aim is to attain an in-depth understanding of human behavior and the reasons that govern such behavior [295]. It allows researchers to examine certain phenomena, develop insights, and report those insights [296]. It involves discovering and understanding causes within a context [292]. In qualitative research, the phenomena can be interpreted in a variety of ways, as it entails an interpretative evaluation [297].

A variety of empirical materials such as case studies, interviews and surveys can be applied. It is therefore important to choose an appropriate method to use in the context of a piece of research, as each practice could make the world visible in a different way [298].

One of the drawbacks of qualitative analysis methods is that they are generally more labor-intensive than quantitative methods. Moreover, their results are often considered 'softer' or 'fuzzier' than quantitative results, and therefore, difficult to report [299].

Quantitative research is an inquiry method that is concerned with quantifying a relationship or comparing two or more groups with the aim of attaining an in-depth understanding of human behavior and the reasons that govern such behavior [295], [292]. Quantitative data is usually collected using controlled experiments, and is typically represented by numbers, depending on the application of the measurements [292], thus allowing comparisons to be made and statistical analysis to be carried out [299]. Data collection in quantitative research can be conducted in many different ways (in which numeric data can be accessed), including case studies or surveys, and uses induction and deduction to search for aggregate patterns in empirical observations [296].

The “bottom up” or inductive exploratory method is used in qualitative research, and is used to develop theories, whereas quantitative research is more deductive or “top down”, and is used to test theories. Quantitative and qualitative methods are complementary [300]. While quantitative analysis can answer many types of questions such as when and how who did what and where, it tends to ignore the more qualitative question of why [298]. Drehmer and Dekleva [301] note that an initial examination of quantitative data can lead to a qualitative study, and according to Miller et al. [302], mixing both methods can provide empirical quantitative findings with a context.

Mixed methods research has proven to be effective in related educational areas. In [303] the quantitative data about the approaches of the students relating to a programming course, and the subjective interpretation of their experiences (qualitatively determined, and statistically and qualitatively interpreted) made the inferences from the study much stronger. The utilization of mixed methods allowed enrichment and triangulation with self-reported data. The principal goal when applying mixed methods is that of tackling a given research question from several relevant angles, making use of more than one type of investigative perspective. Mixed methods research is the third most frequently used research paradigm after qualitative and quantitative research, and may be a powerful, more balanced and informative choice.

One of the common challenges when using mixed methods involves reaching agreements as to the research process in which mixing can occur. This also lead to the difficulty of defining effective strategies for integration at different stages of the research [294].

One of the major advantages of using mixed methods research is that it allows researchers to view problems from multiple perspectives, thus enhancing and enriching the meaning of a singular perspective. A weakness of the qualitative methods versus quantitative is that the ontological and epistemological positions of the researcher are not made explicit [304]. However, the merging of quantitative and qualitative data helps to develop a more complete understanding of a problem [305]. This method also increases the diversity of the data and thus increases confidence in the results [299].

When a quantitative phase follows a qualitative phase, as occurs in this thesis, the researcher’s objective in the first phase is to generate important research questions, and to then provide warranted answers to those questions in the second phase by developing a survey instrument, an intervention or a program that is informed by qualitative findings [305].

3.1.1 Population sampling

Identifying the population to be used when conducting the different evaluation phases is necessary to draw relevant inferences from the results [306]. In order to obtain a generalized set of results, a scientific sampling method must be applied [307]. A balanced set of participants must be chosen by bearing in mind the scope of the study.

Of the various sampling methods that exist [308], stratified sampling has been used in order to conduct the evaluations in this thesis. This method considers dividing the population into sub-groups who all share a similar characteristic. In the case of this thesis, the subgroups considered are mainly: students, practitioners and researchers with different backgrounds and experience in GSD.

Stratified sampling permits the study of differences in the measurements between the different sub-groups. The study sample can therefore be obtained by taking samples from each sub-group.

3.1.2 Data collection methods

This section explains the data collection methods used in this thesis as part of the mixed method research applied. Two main forms of data collection are considered:

- Literature review: data is collected using two systematic literature reviews that set the context for the study by creating a synthesis of existing knowledge and solutions [309].
- Empirical data is collected from questionnaires and automatic evaluations and logs gathered during the simulations.

Questionnaires and surveys provide a flexible empirical method with which to collect quantitative or qualitative data [310]. One of the strengths of the questionnaire/survey is that all respondents receive the same set of questions, thus making it a transparent method [309]. The survey method is a good means to elicit a participant's own opinions and ideas without them being influenced by the others taking part [311].

This thesis applies qualitative and quantitative questionnaires at different stages. One weakness of using questionnaires is that it is possible that the respondents will interpret the questions incorrectly. In those cases in which the questionnaire is structured and quantitative, the respondents are restricted in the level of detail they can supply, resulting in a lower level of sensitivity and quality as compared with in-depth interviews [309]. However, a well designed questionnaire can avoid or minimize this effect. Dybå [312] considers the relative merits of different measurement scales and concludes that a 5 point Likert scale is the most reliable measure, which is the option used in the questionnaires. If too few scale points are used, much information is lost because the scale does not capture the discriminatory powers that the respondents are capable of making. Conversely, by using too many scale points, the respondents' answers may not be accurate as they may not be possible to grade them so finely.

The data collected by these methods need to be analyzed. This analysis will depend on the nature of the data, as explained in the following sections.

3.1.3 Qualitative data analysis

Systematic and reproducible methods for analyzing qualitative data can be applied in research in order to extract valuable information [313]. However, the analysis method may vary depending on the objectives and the data collection method [314], [315]. In this thesis, responses to open-ended questions were analyzed using thematic analysis in the expert evaluations that were conducted.

Thematic analysis consists of detecting patterns (or "themes") within data, thus allowing implicit and explicit ideas to be identified [316]. Themes are categories for the analysis extracted and are associated with a specific research question. The process of thematic analysis is performed in six phases [316]: familiarization with data, generating initial codes, searching for themes among codes, reviewing themes, defining and naming themes, and producing the final report. The primary process used to extract themes from within the raw data is coding, which recognizes important moments in the data and encodes it [317]. One of the major advantages of thematic analysis is that it allows the interpretation of themes supported by data [318], and it also allows themes to emerge from data [319]. However, reliability is a weak point, as interpretations may be subjective depending on the researchers' point of view [318].

3.1.4 Quantitative data analysis

This thesis considers data in the form of questionnaires, the responses to which are measured using Likert scales..Information from the evaluators' interactions as regards the time required or the number of rules fired is also considered. The first step when conducting frequency data analysis [320] is to group data into independent categories and define these independent and mutually exclusive categories, with no overlapping between categories. Data is presented as scores or values in a frequency table which can quickly reveal:

- number of missing values or unanswered questions
- outliers and extreme values
- central tendencies, variability and shape of the distribution.

Of the different statistical measures and diagrams applied in quantitative data analysis, this thesis makes use of the following:

Median: The median is the middle value of a set of numbers arranged in numerical order. If the number of values is even, then the median is the sum of the two middle values, divided by 2. The median Is more convenient that the mean as regards analyzing the information in this study because it is not affected by abnormally small or large values, and can be used with ordinal data.

Mode: The mode of a set of data is the value which occurs most frequently. In this study it is convenient because it permits us to determine the participants' common answers to certain questions.

p-value [321]**:** This is used for significance tests in which there is a null hypothesis (H_0) that will be tested against an alternative hypothesis using a dataset. The alternative hypothesis is expected to be true if the null hypothesis is false. Although the alternative hypothesis cannot be proved, it can demonstrate that the alternative is much more plausible than the null hypothesis, given the data. This demonstration is usually expressed in terms of a probability (p-value) which quantifies the strength of the evidence against the null hypothesis in favor of the alternative.

The p-value (also known as the alpha level or significance level) is therefore the probability of obtaining a test statistic that is at least as extreme as that which was actually observed, assuming that the null hypothesis is true.

The smaller the p-value, the greater the evidence that the data did not originate from the selected distribution. The hypothesized condition is rejected if the p-value is 0.05 or below. This provides 95% confidence that the hypothesized condition (null hypothesis) is not true. Many common statistical tests produce test statistics that can be interpreted using p-values.

Normal distribution [322]**:** When analyzing data, it is important to discover whether this data follows a normal distribution. In a normal distribution, the mean, median and mode coincide, and there is symmetry about the center, signifying that half the values are lower than the mean and the other half are greater. The normal (or Gaussian) distribution function can predict the probability of a number in a particular context falling between any two real numbers. Normal distributions have many convenient properties, and random variates with unknown distributions are therefore often assumed to be normal.

Several normality tests with which to determine whether a data set is normally distributed are available. Data are commonly tested against the null hypothesis for normal distribution. A normal distribution will therefore require the p-value obtained with these tests to be greater than 0.05.

Of the various tests that can be used to test normality, the Shapiro-Wilk Test [323] has been used in this study because it is appropriate for small sample sizes (< 50 samples). The null hypothesis for this test is that the data are normally distributed.

Histogram [324]: A histogram is a graphical representation of the distribution of data, in which relative frequencies are represented by relative areas. It represents tabulated frequencies, shown as adjacent bars with a height equal to the frequency density of the interval.

Histograms are used to plot the density of data, and often for density estimation. A common pattern is the bell–shaped curve, which occurs when the sample has a normal distribution. In a normal distribution, points are as likely to occur on one side of the average as on the other. Histograms may therefore be used to graphically visualize normal distributions. Figure 9 shows an example of a histogram, comparing the distribution with the bell-shaped curve expected from a normal distribution. However, since other distributions look similar to the normal distribution, statistical calculations must be used to prove a normal distribution.

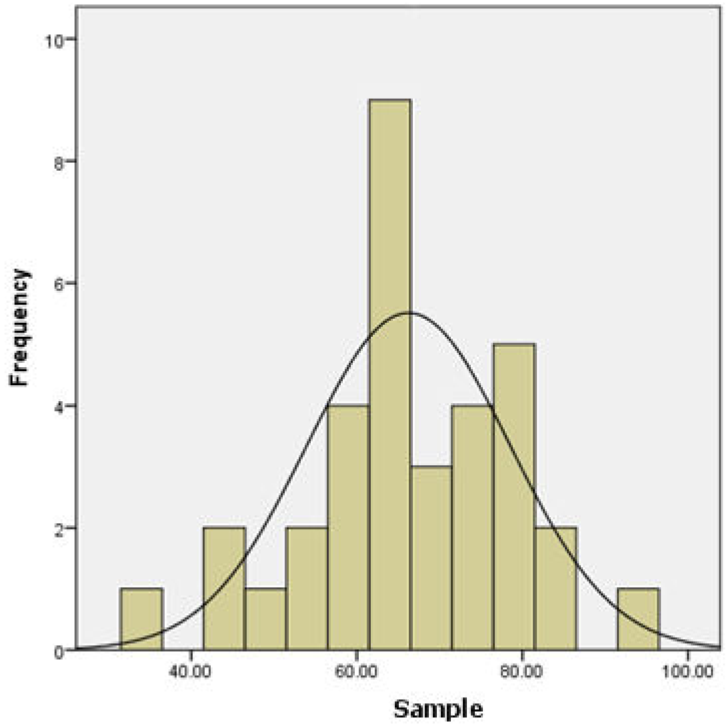


Figure 9. Histogram (sample)

Box plot [325]: this is a graphical representation that is particularly useful for comparing distributions between several sets of data. Box plots depict groups of numerical data through their quartiles, showing the differences between populations without making assumptions of the underlying statistical distribution. They are therefore non-parametric.

Box plots highlight the middle half of the data points and the whiskers indicate variability outside the upper and lower quartiles. Outliers are represented as individual points, and the interquartile range is also visualized. The width of the notches is proportional to the interquartile range. The spacing between the different parts of the box indicate the degree of dispersion and skewness in the data, thus making it possible to identify outliers. When the notches of two boxes do not overlap, this evidences a statistically significant difference between the medians. If the data is normally distributed, the locations of the different marks on the box plot are equally spaced.

Figure 10 depicts an example of a box plot with two outliers, showing: the bottom and top of the box (the first and third quartiles), and the second quartile (the median). The ends of the whiskers represent the minimum and maximum of all of the data.

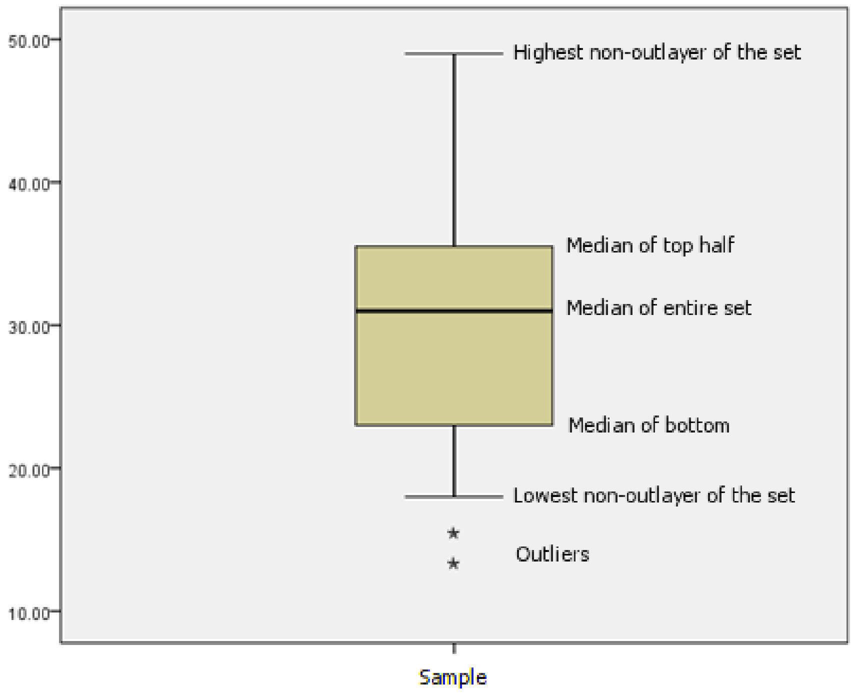


Figure 10. Box plot (sample)

Parametric and non-parametric methods

Quantitative data analysis can be conducted by using parametric and nonparametric methods. **Parametric methods** allow us to test hypotheses about parameters in a population typically described by a normal distribution. It requires assumptions to be made about the distribution of the data. It is also assumed that the variance is uniform either between groups or throughout the range being studied.

Non-parametric methods are, on the other hand, used to analyze data which do not meet the distributional requirements of parametric methods. Non-parametric methods do not depend on a complete specification of the probability distribution of the population from which the sample was drawn. In particular, skewed data are frequently analyzed using non-parametric methods. Non-parametric techniques are consequently less powerful, because they use less information in their calculation and are only valid under relatively general assumptions about the underlying population [326]. However, according to Kitchenham et al. [306] most non-parametric methods "are very efficient relative to their parametric counterparts and they are effective with small sample sizes".

If the distribution of the population does not follow a normal distribution according to the Shapiro-Wilk test, then the quantitative statistical analysis methods used in this thesis are non-parametric. The following non-parametric tests are used in this thesis:

Mann-Whitney U test [327]: The Mann-Whitney U is a non-parametric method that tests the null hypothesis that two populations are the same against an alternative hypothesis, especially that one particular population tends to have larger values than the other. It is commonly portrayed as the non-parametric substitute for the Student's t-test when samples are not normally distributed, and it is almost as efficient as the t-test as regards normal distributions. It is also useful when comparing how two groups of experts respond to a number of key items [328].

The Mann-Whitney U is a rank-order test that assesses the location and range of the lowest group's distribution within the overall sample range, and contrasts this with a theoretical ranked distribution approaching normal ('U' or 'z' distribution, depending on the sample size).

Wilcoxon test [329]: Frequency distributions are not sufficient for a comprehensive analysis of the data. Hypothesis testing methods are required to attain a fuller understanding. Hypothesis tests can provide an understanding of how statistically significant the differences observed are, and whether or not the variables are independent.

The participants involved in this study are not a large group selected through sampling method. Most of them are potential users and there are also some experts in GSD. The characteristics of the population and the design of the experiment therefore signify that a normal distribution cannot be assumed.

The Wilcoxon test is a non-parametrical statistical hypothesis test that is used to compare two related samples, matched samples, or repeated measurements in a single sample to assess whether their population mean ranks differ. It is appropriate for analyzing the data from a repeated-measures design with two conditions, and provides an alternative to the t-test when the population samples cannot be assumed to be normally distributed.

The logic behind the Wilcoxon test consists of ranking the data to produce two rank totals, one for each condition. If there is a systematic difference between the two conditions, then most of the high ranks will belong to one condition and most of the low ranks will belong to the other. The rank totals will be quite different, and one of the rank totals will be quite small. However, if the two conditions are similar, then high and low ranks will be distributed fairly evenly between the two conditions and the rank totals will be fairly similar and quite large. The Wilcoxon test statistic "W" shows how likely it is to obtain a particular value of W purely by chance. The Wilcoxon test is used with small sample sizes.

3.2 Validity

A number of threats to validity may appear depending on the method and conditions in which the research is conducted. An analysis of these threats helps to determine whether the conclusions drawn in a study are accurate. The external, internal and construct validity of a method should therefore be identified in order to maintain the integrity of the study.

- **External validity**. External validity relates to how well the conclusions of research conducted over the population sampled can be related to a broader population. External validity therefore refers to generalizations that can be made about propositions, inferences, or conclusions developed from scientific research [330].

- **Internal validity**. Internal validity is the extent to which changes in one variable cause changes in another variable. Causality is a factor in internal validity that suggests that a particular outcome creates a greater probability that a change will produce a particular outcome. The possibility that observed changes are the result of some unidentified reason must be taken into consideration when analyzing a research method [330].

- **Construct validity**. Construct validity is the extent to which a study assesses what it claims to measure [330]. It considers that variable correlation should not occur with changes in unrelated variables and, on the other hand, that changes in one variable must cause a change in other variables for which there should be a correlation. Construct validity provides a means to assess how well actual propositions, inferences, or conclusions reflect theories about how they affect each other [330].

3.3 Research strategy conducted

The research conducted in this thesis was guided by an iterative and incremental approach through which the framework was improved and iteratively extended.

The research method conducted, which is depicted in Figure 11, involved the following phases: The GSD domain was initially studied by using a systematic literature review (phase 1) which was guided by the research question: *What are the initiatives carried out in relation to the improvement of DSD processes?* This served to discover the problems in GSD on which the research should be focused. This study led to the decision to concentrate efforts on providing GSD training. Phase 2 of this research consequently consisted of conducting a second systematic literature review on GSD education focusing on the research question: *What are the initiatives carried out in relation to Global Software Development training?*

Certain problems were encountered in the traditional training methods, mainly related to coordination complexity and difficulties as regards reproducing real world problems. This led to a new research question: *Is it feasible to apply new effective methods for GSD training and avoid the problems of the traditional methods?* The findings of the literature reviews served to design a prototype of the educational framework based on simulation. Previous to the implementation, an initial Expert Evaluation served to obtain feedback on its usefulness (phase 3). After completing this evaluation, the framework was improved and a set of tools providing the framework with full support were implemented.

Phase 4 of the research method was to conduct a new Expert Evaluation in which evaluators, after using the main components of the first version of the framework (VENTURE V1), provided feedback on how to improve the usability and effectiveness in GSD training. This evaluation was conducted through the development of a short simulation.

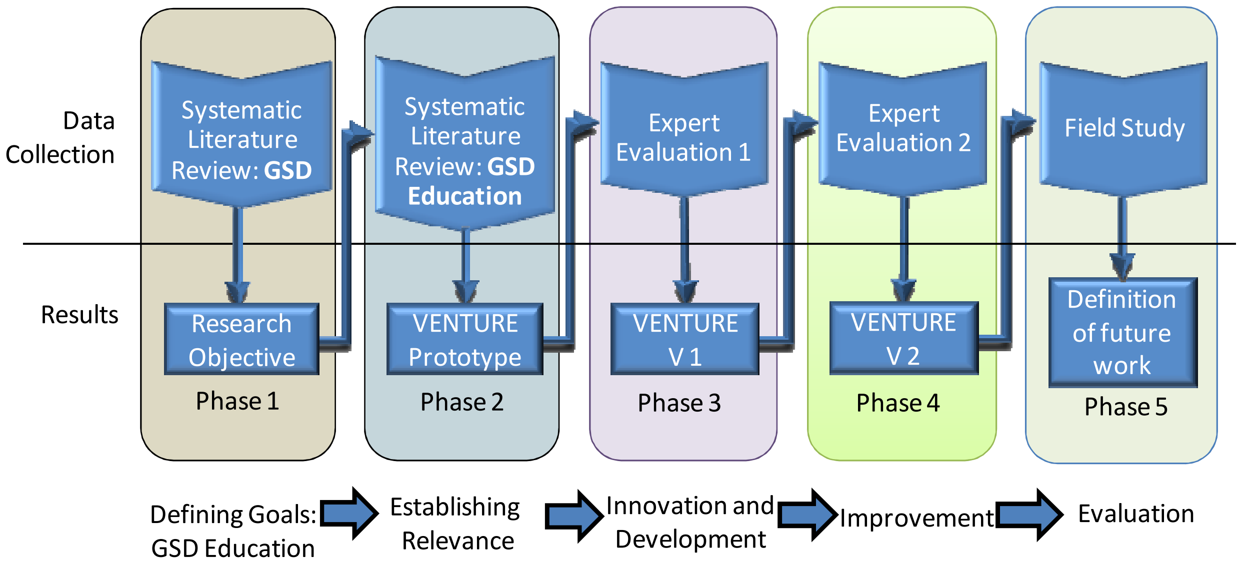


Figure 11. Research methodology

After considering the feedback received, an improved version of the framework and the tools were generated (VENTURE V2). The last step (phase 5) of the research method consisted of conducting a Field Study with the main objective of testing whether the framework has the potential to promote learning in GSD. Two detailed simulations were developed in order to conduct this evaluation. This final step will act as a guide for future work, in which feedback from the field study will be analyzed and assessed in order to improve the framework and the tool.

The results from phases 1 and 2 have been presented in the chapter concerning the state-of-the-art. The following chapter presents the Framework. Chapter 5 reports on the analysis and the results of the two evaluations conducted in phases 3 and 4. The resulting Version 2 of the VENTURE tool is described in Chapter 6, whereas the Evaluation of Version 2 (Field Study in phase 5) is presented in Chapter 7.

3.4 Conclusions

This chapter presents the research methodology applied in this thesis. Given the characteristics of this research, a phased-based evaluation method was considered, since it is an effective means to guide the development of a research based framework.

Two Systematic Literature Reviews served to focus the topic of the research and to gather the requirements and ideas for the initial design of the framework. An early Expert Evaluation acted as a proof of concept of the framework, while a second Expert Evaluation with experts in the knowledge domain who used the underlying tool, provided early feedback that anticipated challenges in the domain. The final step of the methodology consisted of evaluating that it is possible to learn GSD by using the framework on a complete GSD training course. The following chapters describe the final version of the framework and the tool, after which the Expert Evaluations conducted explain what improvements were made to achieve that version of the framework and tool. Finally, the validation chapter tests whether real learning is achieved when users interact with VENTURE.

Chapter 4

VENTURE Framework

The state-of-the-art presented in the previous chapter reveals that there is a gap between training skills and the knowledge required to working in GSD. The most relevant findings are the difficulty of coordinating various institutions, the limitation as to the number of participants, accessibility and the difficulty of reproducing specific training scenarios. After studying the problems associated with traditional educational methods, the focus of this thesis is oriented towards the definition of a framework with which to provide GSD training by avoiding or minimizing these difficulties. Given the nature of the problems, the objective of the framework is to create GSD training simulations. Virtual Agents take part by playing different roles and simulating different cultures in order to interact with the trainees. This simulation-based framework provides:

- Independency: trainees only need to interact in real time with Virtual Agents.
- Accessibility: it avoids coordination difficulties with other institutions and training costs.
- Scalable learning: it can provide a variable number of trainees with a training service without increasing costs.
- Assessment: it combines automatic and manual assessment that minimizes trainees' and instructors' workloads.
- Accuracy: it reproduces specific GSD scenarios that it would be difficult to reproduce by other means.
- Real world scenarios: it considers the elicitation of real world GSD problems and scenarios that can be simulated in order to tackle specific problems in which organizations may be interested.

VENTURE (Virtual ENvironmenT for commUnication and collaboRativE training) therefore makes it possible to simulate GSD interactions between students and Virtual Agents. Virtual Agents will interact in an autonomous manner, in an attempt to perform certain GSD activities that will lead the students to follow the lessons and guidelines provided in the theoretical lessons. VENTURE therefore applies a Scenario-Based Learning approach in order to define the structure of virtual meetings in which GSD interactive scenarios are reproduced.

These scenarios are made available through an e-learning environment, which also provides theoretical material that presents those concepts which will be practiced and reinforced by means of simulation. By accessing this environment, instructors can track the students' activities and review their automatic assessments. The automatic assessments, along with manual assessments that can also be applied, can be used by the instructors in order to assign training materials adapted to students' specific needs.

A scenario designer has been integrated into the framework in order to facilitate the creating of scenarios. This designer permits the definition of the Virtual Agents involved and the GSD problems and challenges that the students will tackle when interacting with them. However, one of the problems of designing accurate training simulations is that of gaining access to scenarios and problems that occur in real life in order to reproduce them. The framework therefore considers the collaboration of a GSD community that is enabled to provide and review GSD scenarios and patterns that reflect solutions with which to tackle specific GSD problems.

The main components of the framework, which have been introduced in this section, are depicted in Figure 12. The following sections provide a more in-depth view of those components. Prior to this explanation, the virtual agents that may be involved in the simulations and the different roles that can take part in the learning process are detailed.

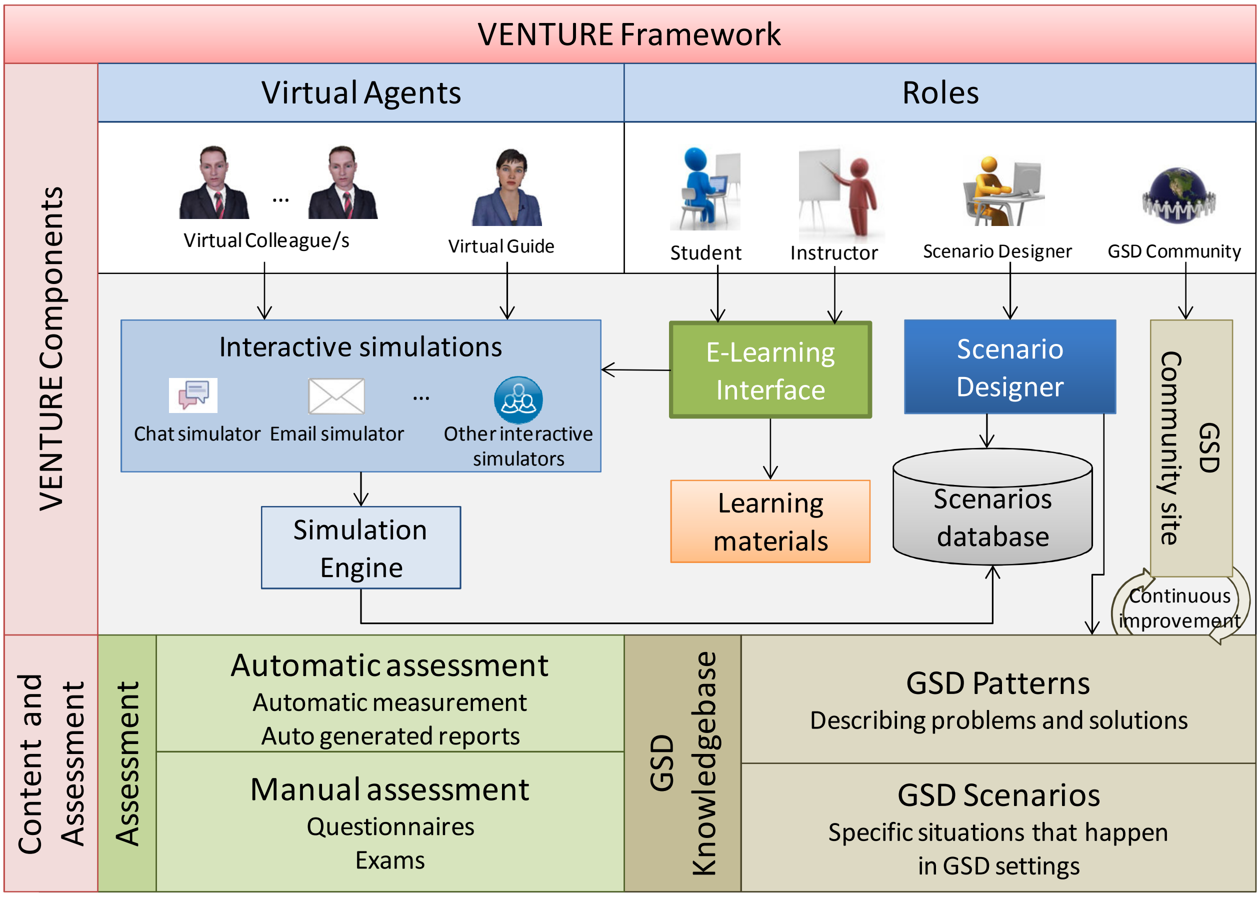


Figure 12. VENTURE Framework

4.1 Roles

The introductory section reflects on the fact that the framework has components with which to provide training, design this training and collect valuable information that could serve to design effective simulations. Each of these components is used or accessed by different actors. It is therefore appropriate to define the main roles that will participate in the different stages of the learning process:

- **Students**: Will be able to access their assigned learning and training materials by means of the e-Learning environment. They will execute GSD simulations that will allow them to interact with Virtual Agents with the objective of reproducing typical GSD problems. The students are automatically and manually assessed during these interactions, and can both access their marks and interact with the instructor and other students.
- **Instructors**: Can design courses and make learning and training material available in the e-learning environment. They can also assign tasks to the students, including the assignment of simulations. They are in charge of solving the students' problems and questions, and can interact with them. They coordinate courses, prepare the schedule, track the students' progress and review their assessments.
- **Scenarios Designers**: Are in charge of creating the simulations or update existing ones. This is done by providing them with access to the scenario designer, which allows them to easily edit and reuse simulations. They define the sequence of problems that will be simulated, the GSD context in which these problems appear, the Virtual Agents involved and the role of the student. Finally, they are also responsible for making the scenarios available for their execution.

- **GSD Community**: The knowledge required to implement accurate GSD simulations is defined and reviewed by a GSD Community. This community collaborates in the iterative improvement of a knowledgebase of GSD scenarios and patterns. This knowledge can be used by the scenario designers in order to create simulations and by the instructors in order to create learning material.

The following sections provide more details on the components of the framework in which these roles participate.

4.2 Virtual Agents

The simulations that can be created with the framework will allow students to interact with Virtual Agents in the context of a GSD scenario. Simulations are basically context-sensitive interactions in which stakeholders from different nationalities will confront a number of problems. In order to reproduce these GSD settings, students will interact with Virtual Agents playing roles in the interaction. Interaction problems such as those caused by cultural and linguistic differences, the use of different communication protocols, etc., may therefore appear. The framework considers creating simulations with two types of Virtual Agents:

The **Virtual Guide** will direct the trainee during the interaction, intervening in different ways depending on the context of the simulation:

- Providing guidance on the tasks that the students must perform. The Virtual Guide can provide information on the scenario, propose solutions or provide detailed explanations.
- Providing the students with feedback when they intervene in an inappropriate manner. Students can receive feedback if they make mistakes.
- Explaining problems that occur in a context (e.g. cultural and linguistic problems). During the execution of the scenarios, the Virtual Colleague can make interaction mistakes on purpose or show customs that would require the student being given an explanation.
- Preventing the student from initiating off-topic interactions.
- Helping the student to cope with the stage in those cases in which he/she does not know how to interact.

The **Virtual Colleague** plays a role in a GSD scenario, interacting with the student by asking and answering questions, and providing opinions, ideas or suggestions. The Virtual Colleague simulates that it is a colleague from a certain location, and can therefore be identified by a certain culture and personality. It can consequently show cultural behavior and commit interaction mistakes typical of that culture.

Both types of Virtual Agents can show gestures, customs and accents that can serve to provide students with an awareness of the cultural and linguistic differences that they may encounter in GSD when interacting with different means of communication.

The involvement of Virtual Agents solves some of the problems found in traditional methods, as they allow the reproduction of scenarios involving specific cultures and problems that it might be difficult to reproduce, and avoid the need to interact and synchronize with real peers.

4.3 Scenario designer

Simulations are designed by means of the scenario designer, which allows the definition of the parameters and settings of any kind of interaction in a GSD context. This context is defined by attending to the kind of distribution of the project, the team members involved and their characteristics, the kind of project and its status. The designer also allows an explanation to be defined for the trainee, detailing his/her role in the simulation, the objective or problem to be solved and an introduction to the Virtual Agents involved.

Virtual Agents with specific personal features and background are also defined and included in the simulations in order to interact with the students playing a role in the scenario.

The interaction is defined as a set of sequential stages that define the flow of the interaction. Each stage considers a well defined part of the interaction in which a specific problem is simulated within a given context. When the trainee interacts properly, the transition to the next stage of the interaction is automatic. In order to define these stages, the following information is required:

- Virtual Colleague's interaction: including the interaction knowledge that the Virtual Colleague needs to respond to trainees' interactions in that specific context.
- Virtual Guide's interaction: including feedback and guidance that the trainee may require during that specific stage of the interaction.

In those cases in which the Virtual Guide interacts to correct an inappropriate interaction, it is also possible to define a value that will be deducted from the final score that the student will receive. The problems that can be detected are organized according to the kind of problem within a GSD context.

This definition, which is based on a sequential transition from one stage to another involving Virtual Agents, permits the reproduction of GSD settings with a high granularity. This definition of the simulations is stored in the **scenarios database**, which serves as a repository for the reuse of previous scenarios. As the definition of the simulations already includes information about the objective, participants and abilities trained, this database only needs to store the definition. The scenarios contained in this database are used by the *simulation engine* in order to enact their execution through the e-learning interface.

The scenario designer therefore permits the reproduction of scenarios involving specific problems that it may be difficult to reproduce in traditional educational approaches. The problems and knowledge to be trained can be designed by considering different teaching needs in a systematic manner.

The design of trails that could branch the interaction depending upon response from the participant has been studied in other educational fields [331]. However, minimizing the design effort was an important objective of this proposal and this option would increase the design complexity, making the sequential option more adequate.

4.4 Interactive simulators

The principal innovation as regards this framework is that it applies interactive simulations of GSD scenarios with training purposes. The intention of these interactive simulations is to promote the acquisition of interaction abilities such as communication and teamwork skills in globally distributed environments.

The framework considers the simulation of synchronous and asynchronous means of communication such as chat or e-mail. The **chat simulator** places the trainee in a chat conversation in which the Virtual Colleague/s and the Virtual Guide textually interact with the student through a chat interface. Additionally, the avatars communicate through synthetic voices through a text-to-speech engine. The **e-mail simulator** involves the trainee in an asynchronous interaction using an e-mail interface. Specific problems associated with these means of communication, such as delays in the answers, can be simulated. Moreover, GSD factors that influence their use, such as the cultures involved, the processes followed, the characteristics of the project and the teams involved, can also be reproduced.

During the simulations, trainees can interact with Virtual Agents by following the flow depicted in Figure 13. The student will initially interact with the Virtual Colleague. If he/she interacts in an inappropriate manner as regards the GSD context, the Virtual Guide will provide him/her with feedback, after which, the student will need to interact again. If the interaction is correct, the Virtual Colleague will interact. If the Virtual Colleague's interaction needs any explanation, the Virtual Guide will provide feedback about any of the cultural, linguistic or GSD issues that may have appeared, after which, the student will interact again.

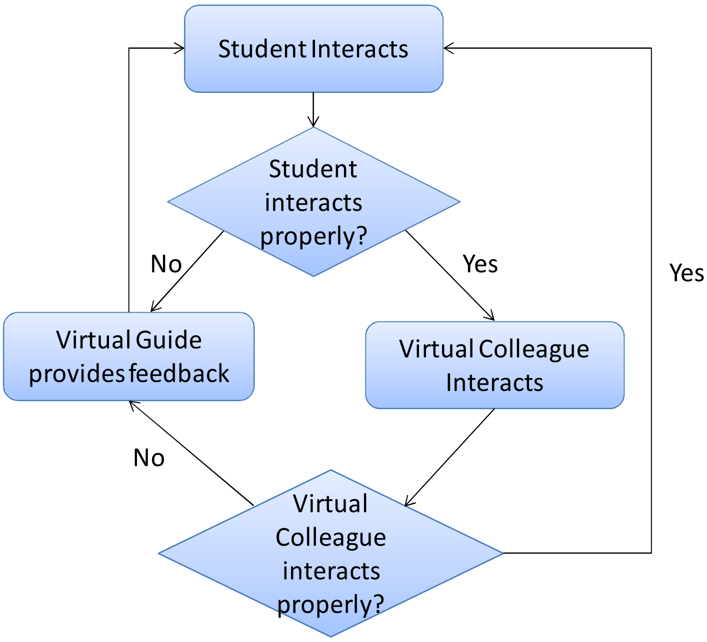


Figure 13. Simulated conversation flow

The interactive simulators can additionally be used to introduce concepts regarding GSD or simulate problems that are not necessarily related to the means of communication, but with any issue that may arise in GSD and can be discussed using these means. Further interactive simulators could be implemented in order to consider new means of communication or collaboration. For example:

- Telephone conversations and videoconference: spoken conversations can also be considered in addition to the textual options by applying voice recognition methods.
- Collaboration through forums, blogs and wikis: simulating collaborative textual interactions with virtual participants.
- Remote desktop interaction: In GSD it is usual to share remote desktops in order to solve problems or discuss issues. This kind of interaction could also be simulated by gaining access to a Virtual Agent's desktop.
- Management of work items: Many tools applied in GSD support the management of work items. A work item is a means of keeping track of the tasks, issues, bugs, requirements, or other items that a software team needs to address during the development cycle. The creation and assignment of work items, the intermediate steps requiring interaction with assignees and their resolution could be simulated with Virtual Agents.

The main advantages of the simulators as opposed to traditional educational approaches are independence, availability and repeatability. Trainees can use the simulators at any time without interacting with other peers and they can reproduce the same scenarios again and again until they have improved.

4.5 Simulation engine

Interactive simulators are executed by the students in order to train specific topics. Although there may be a number of different kind of simulators for different interaction means, such as chat or email simulators, all of them are based on the same concept of applying a sequential set of phases in which Virtual Agents take part. A common simulation engine can therefore provide all the different kind of simulators that can be implemented with a service.

The simulation engine thus gains access to the scenarios stored in the **scenarios database** and loads and enacts them. It orchestrates the actions that will take place during the simulation by loading each stage in a sequential manner during the execution. For each phase, it will receive actions from the students and will send a response with an action that will be performed by either a Virtual Colleague or the Virtual Guide, depending on the entry provided by the student and the definition of the specific stage.

The use of a common simulation engine makes it easier to integrate new kind of simulators into the framework in order to provide training on new interactive means without implementing a new simulation engine or a new scenario designer.

During the execution of the scenarios, the engine keeps a record of the student's interaction and assesses the students when they interact in an inappropriate manner according to the definition of the simulation. This automatic assessment is also an improvement on traditional GSD educational approaches.

4.6 E-learning interface

Both students and instructors use the framework by means of the e-learning interface. It contains **learning materials** (GSD theoretical lessons and practices in GSD) that will be made available to students by the instructors. The instructors can also make theoretical lessons on GSD available. The e-learning interface allows the sharing of resources and students' task schedules, and it facilitates communication between students and instructors.

Theoretical lessons are supported by simulations that will put theoretical knowledge into practice. Instructors schedule simulations that will be executed by the students by means of the e-learning interface. Each lesson has deadlines for the completion of the tasks assigned.

At the end of a course the student may also be asked to complete self-assessment activities. They can also review the scores that they have obtained in the automatic and manual assessments, as explained in the following section.

4.7 Assessment

The assessment of a student's knowledge and abilities is an important part of the framework. VENTURE Framework considers automatic mechanisms with which to evaluate a student's performance, and **manual assessment** methods such as questionnaires or exams in order to measure knowledge improvement.

From the student's point of view, the framework provides "informating down" facilities [332], which contain information about their performance, deadlines, qualifications, historical actions and instructors' comments at their disposal. The students can also discover which team members are online and receive automatic notifications about news or tasks, or fill out questionnaires and take exams. From the point of view of the instructors, the system provides "informating up" mechanisms [332], which allow them to discover the status of the tasks and actions of the students and communicate with them in real time.

The assessment process defined is based on good practices drawn from previous studies (summarized in Appendix I) and their adaptation to VENTURE. Minimizing the instructors' effort and tailoring the assessment to specific training needs are also primary objectives.

There are several benefits as regards combining an assessment with the training. For example, the students themselves will benefit from the assessment by having a feeling of achievement, and research shows that students are more motivated to learn when there is some 'proof' of learning [333]. What is more, in the case of companies, managers will have a better view of the workforce as a whole, and will be able identify where individual strengths and weaknesses lie as regards communicating with different cultures or in different languages. This knowledge will help identify where further training is needed, and how best to use the staff resources available.

A good assessment must be valid, reliable and transparent, and must clearly outline the goalposts [334]. Assessment procedures can vary depending on the type of training, tools employed, training methodology, individuals involved or field of application [334]. Although lessons learned in creating GSD learning environments have been reported [12], methods for GSD learning assessment is an area that still needs further research.

Assessment objective

A key challenge of VENTURE is how to objectively assess the level of learning achieved. The aim of the assessment defined for this framework is twofold: to provide students with advice on the skills and knowledge they have and those they must improve, and to help the instructors to create simulations based on each student's experience. More specific objectives are:

- Early identification of the skills in which the student needs training.
- Assessment of the student's ability to put into practice or retrieve the knowledge learned in training sessions.
- Monitoring of the student' progress in terms of knowledge and skills gained.
- Use of unambiguous, systematic, reliable and objective assessment criteria.
- Promotion of knowledge construction.
- Minimizing the time and resources needed by students and instructors.

Assessment process

In order to achieve the aforementioned objectives, an assessment strategy has been designed for the simulated learning environment. This assessment incorporates concepts taken from assessment strategies found in the related literature, which fall into four kinds of assessment, discussed in this section.

1. **Diagnostic assessment** in order to determine the students' initial level of knowledge and skills. The educator will determine the assignments for the students based on this initial level. The scores obtained in this assessment do not directly contribute to the students' final grades, but provide a base-line measurement to act as proof of their having advanced at the end of the course or unit.

2. **Formative assessment** by means of simulation. The students can experiment with different means of communication through feedback, asking questions and taking risks that they would not necessarily take in real settings. VENTURE provides students with feedback and direction, thus allowing them to improve their communication skills, which are assessed in real time.

3. **Summative assessment**, which measures achievement as regards the course's objectives. This framework considers the evaluation of artifacts that can be generated during the learning process, along with complementary materials such as exams or questionnaires.

4. **Self assessment** is also used to allow students to judge their own abilities. Surveys and questionnaires may be applied.

In VENTURE, students are assigned reflective tasks, and learn from their successes and failures through the provision of immediate feedback. The framework gathers these reflections by automatically monitoring their advances and measuring their knowledge.

Instructors can receive feedback from the students and examine their self-assessments. The analysis of these reflections is used to examine whether the students have attained their goals. During the simulations, reflection is promoted by: explaining the consequences of certain actions, proposing alternatives or giving the student the opportunity to find alternatives, thus allowing them to see the problem from various perspectives and placing them in different and varied contexts.

The assessment process developed (shown in Figure 14), is described according to the following activities in the training process:

A) Initial students' categorization (*diagnostic assessment*): Students are categorized in order to establish their assignments and to determine to what extent they have improved their knowledge at the end of the training process.

B) Course preparation and customization to students' characteristics: Instructors prepare the theoretical and training materials based on the previous categorization. Courses are divided into different GSD lessons according to the instructor's criteria. Training modules are designed by focusing on GSD problems that are suited to each student.

Preparation and adjustment of simulations focusing on specific skills or knowledge is also conducted at this stage. Instructors can decide which simulations (from a selection of available simulations) will be assigned to the student. In the case of specific needs, instructors can also adapt existing scenarios by customizing them to specific training objectives.

Training periods and timelines for the data collection and assessment are established at this stage, and students must be informed about the assessment procedure and schedule.

C) Study the theoretical material: In VENTURE, topics are first covered by means of theoretical lessons and are then trained by means of practical simulations. This cycle, in which theory is learnt and practical simulations take place, is repeated for each lesson until the scope of the course has been completed.

D) Execute training simulations: After completing the different lessons and their corresponding simulations, the students submit the deliverables. For each simulation, the instructor will monitor the accomplishment of each learning objective by reviewing the logs of the student interactions. At the end of the simulation, prior to producing the report, the user can review any mistakes committed, one by one.

Learners will be assessed at the beginning and at the end of the training session in order to measure their improvement. These assessments serve to adjust learning through an early identification of the students' problems and needs. After each lesson, the students receive the instructor's comments and future lessons can be adapted accordingly.

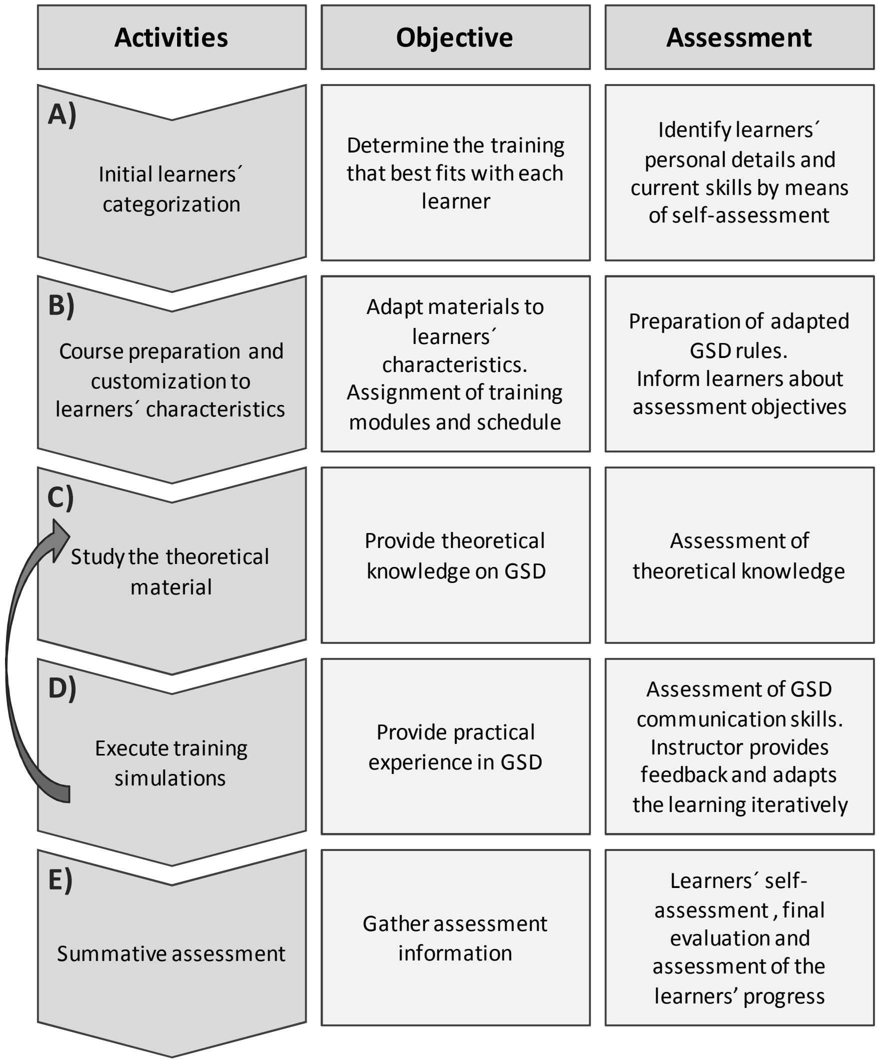


Figure 14. Assessment process

The results of these evaluations, along with the information provided by the automatic reports generated by the simulators, will serve to assess the student (*formative assessment*). Specific assessment materials of the lessons studied can also be created by the instructor; however, this could result in a great deal of effort. In order to tackle this issue, the framework aims to provide a wide set of teaching materials and simulations dealing with common GSD problems, thus promoting the reuse of these components.

E) Summative assessment: the final step consists of a *summative assessment* which encompasses the different marks of the final reports of the simulations and assessment artifacts. Each of these assessment methods can be applied using different weights in the final score, according to the instructor's criteria.

Automated assessment

VENTURE Framework provides automated assessment. Students are informed of their results via the use of this framework, thus providing a continuous evaluation of the students' knowledge. Instructors also have access to this information in order to track the students' progress.

The students' performance is automatically stored during the execution of the simulations. Moreover, the results of the assessment artefacts provide additional information on their knowledge improvement. The indicators that can be considered are: time taken, evaluation of the practical and theoretical exercises, delay in the upload of the deliverables, and factors regarding the students' activities during the simulations, i.e., the percentage of conversational knowledge that was not triggered, average response time, corrections made by the Virtual Guide, etc.

4.8 GSD knowledgebase

One of the main problems as regards creating helpful simulations and learning material is related to the difficulty involved in gaining access to valuable contents. This can be achieved by creating a representative knowledgebase of common problems that occur during real interactions in GSD. The GSD knowledgebase considered in the framework stores GSD-related knowledge relating to scenarios and common problems based on real experiences and problems studied in literature. In order to gather these experiences, a model based on **GSD patterns** and **GSD scenarios** has been developed. Scenarios represent real events that occur in a particular GSD context. The problems and solutions that are extracted from these events lead to the definition of patterns.

Patterns describe problems that usually occur in an environment. They present the core of the solution to that problem in such a way that this solution can be used in the future to tackle problems that may follow the same pattern in a similar context [335]. In this section, a method with which to define GSD scenarios is proposed, along with the patterns that appear in them with regard to problems and solutions related to communication, coordination or software development processes in the GSD field, is presented. Scenarios represent real events that take place in a particular GSD context. The problems and solutions that are extracted from these events lead to the definition of patterns. Patterns are the result of processing and generalizing the information that has come into being in a way that can be reused in similar contexts. Scenarios could therefore be associated with existing patterns and could be applied in the design of simulations in VENTURE.

In order to facilitate knowledge sharing, this pattern model is integrated into a GSD Community Web with the intention of promoting collaboration between industry and academia. This section presents a model that aims to identify pedagogical patterns from relevant GSD scenarios in order to promote their reuse. This model can be supported with a repository that can be used in software process improvement, research and teaching, and for training purposes in GSD. GSD patterns will be gathered by reviewing the relevant literature and with the participation of the GSD community (researchers, practitioners and instructors). This knowledge will therefore improve in an iterative manner and be contrasted with expert opinions and experiences. The model aims to fulfill the following objectives:

1. To allow the GSD community to incorporate, in an iterative manner, new scenarios and patterns that could be validated and reused.
2. To facilitate the sharing and reuse of patterns and scenarios.
3. To provide a platform for the promotion of collaboration within the GSD community.

Defining GSD patterns

GSD involves a wide variety of problems in different fields, and different kinds of GSD patterns can consequently be derived. The collaboration of a community in the gathering of this variety of knowledge makes it necessary to define a common and simple means of representing and storing accurate patterns. Examples of GSD patterns can be found in Appendix II.

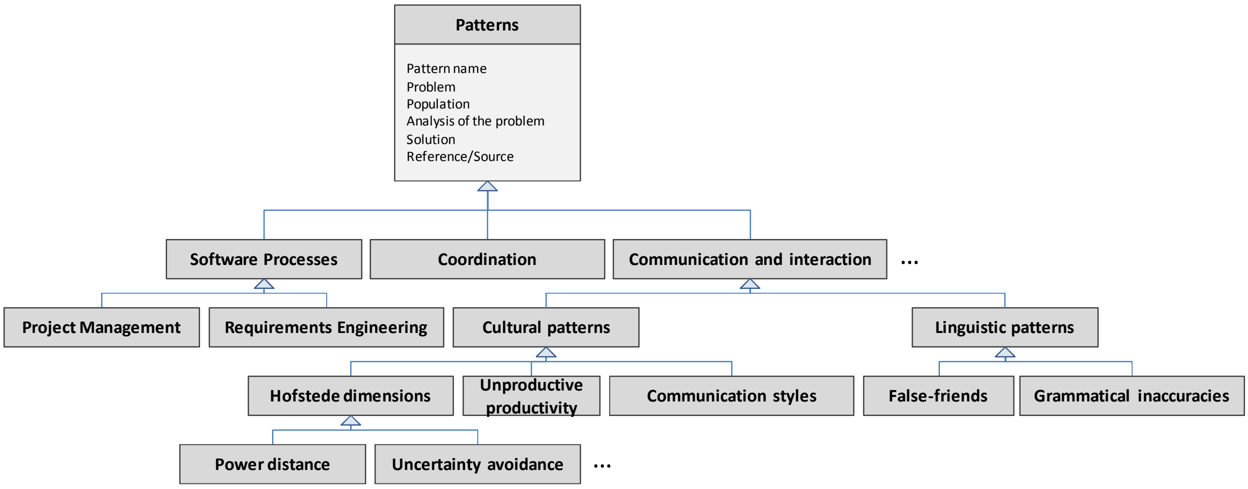


Figure 15. Pattern definition model

These patterns are defined by following a hierarchical model based on abstraction, in which the lower levels of the hierarchy make use of their own attributes, along with the attributes defined in the upper levels. Figure 15 provides a summary of the patterns found in our literature review, and includes some of the patterns defined during the first stage when following this hierarchical model. The first level defines the common attributes that all the patterns will share:

- **Pattern name**: descriptive title of the pattern.
- **Problem**: detailing both the problem and the conditions in which it may occur.
- **Population**: population that could be affected.
- **Analysis of the problem**: detailing information about the context or consequences of the problem.
- **Solution**: including practices that can avoid or mitigate the problem.
- **Reference/Source**: patterns can be gathered from literature or other sources.

These attributes are similar to those employed by Coplien and Harrison [336] who apply: name, context, statement of the problem and solution. However, as the context is important in GSD, it has been divided into the following attributes: population and analysis of the problem, with the aim of providing details about the population and consequences in that context.

These attributes should enable any kind of GSD pattern to be defined. For example, cultural patterns representing common problems in the communication field include:

- Hofstede cultural dimensions [256]: power distance, individualism, uncertainty avoidance, masculinity and long term orientation.
- Communication styles: direct/indirect, formal/informal.
- Unproductive productivity [8].

Table 14 shows how the "Common vision of tasks" pattern is defined according to the attributes used to define a pattern. Both cultural dimensions and linguistic patterns can be defined by means of the attributes, as in the example. Moreover, patterns can be considered as anti-patterns depending on the context (scenarios).

Table 14. Example patterns in GSD

Attribute	**Description**
Pattern name	***Common vision of tasks***
Problem	Vendor-team members work on tasks that are not really necessary. Improved focus on tasks that matter will increase productivity. Tasks are considered productive when they produce the results that the client requires despite the vendor's perception that they are unproductive.
Population	Client and vendor teams from different organizational cultures.
Analysis of the problem	Unproductive-productivity occurs when two different cultural models of productivity appear: the client team has a numerical perception, whereas the vendor team perceives productivity from the perspective of completion and quality of the work.
Solution	Vendors must communicate their perception of the tasks assigned to include how they view productivity. Clients have to be flexible in their productivity indicators in order to consider vendors' estimations and suggestions.
Reference	[8]

GSD scenarios

Scenarios define specific situations that may occur during the software process lifecycle in GSD. The different processes followed, composition of the teams, human factors, project characteristics or collaboration difficulties may influence a particular scenario in a different way. The definition of accurate GSD scenarios therefore requires the specific context in which they occur to be made clear. The scenarios will be stored along with their associated context and may be associated with an instance of a pattern. Recording the scenario–pattern relationship in this way permits transparency, traceability and repeatability. Šmite et al. [132] developed a classification scheme for GSD-related empirical research analysis. Some of their classification attributes are applied in the definition of the context in which GSD scenarios take place.

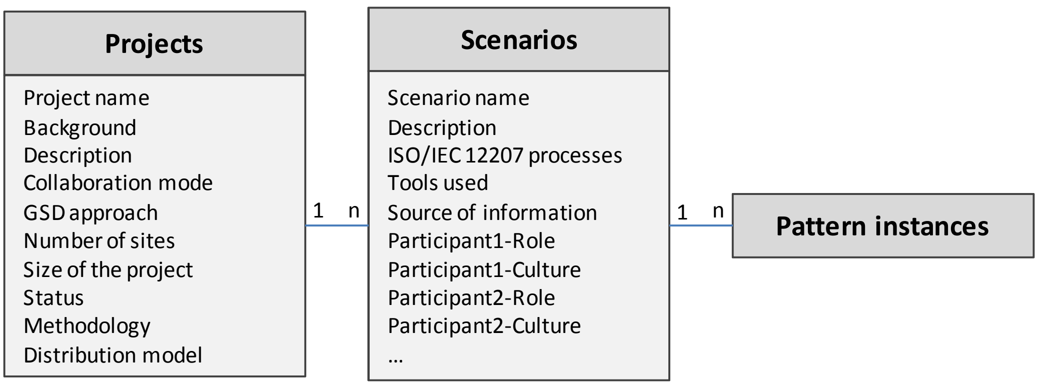


Figure 16. GSD scenario definition

The data model used to store the contextualized scenarios is set out in Figure 16, and comprises information related to the project background in which the scenarios occur, along with the details of the scenario and the specific instances of patterns that appear in it. The project entity contains the following attributes:

- Project name.
- Background: laboratory, industry.
- Description: details of the project.
- Collaboration mode: inter-organizational, intra-organizational.
- GSD approach: module-based, phase-based, follow the sun.
- Number of distributed sites.
- Size of the project: number of participants.
- Status of project: on time, delayed.
- Methodology applied: Ad-hoc, Waterfall, Prototype, Iterative, Extreme Programming (XP), Scrum, Agile.
- Distribution model: outsourcing, insourcing, offshoring, nearshoring, offshore outsourcing.

Each software development project can have several associated scenarios. For example, in the context of a project it is possible to consider a chat conversation among developers in an attempt to solve a problem, or an email conversation between a project manager and developers for the assigning of tasks. The scenario entity covers the following information:

- Scenario name.
- Description: details of the scenario.
- Software processes related to the context of the problem.
- Source of information: origin and method of empirical data collection: audio recordings (meeting, telephone), case study, existing communities or forums in the field of GSD, experience, experiment, interviews, literature in the field of GSD, logs of conversations (e-mail, chat), pedagogical materials, problems reported by companies, observations, reports and surveys.
- Tools used in the scenario.
- Roles, responsibilities and cultures of the stakeholders involved.

When a new scenario is created, and depending on the particular definition of the GSD scenario, several GSD patterns can also be instantiated and associated with the scenario. The attributes that define an instance of a pattern depend on the kind of pattern.

GSD community

In 2004, Hargreaves et al. [337] reported the need to build a strong GSD research community, and envisioned a website for the exchange of documentation, articles, project information, events, research techniques, models and theories. However, no further work on this has been reported to date. To tackle this issue, a GSD Community Web is considered in this framework with the aim of supporting the pattern-based model. This website will allow members of the community to contribute their knowledge, and provide patterns and scenarios.

The collaboration of the **GSD Community** is needed in order to populate this knowledgebase by gaining access to real experiences and knowledge. A **GSD community site** is therefore, part of the framework. Participants can access the knowledge that is shared and collaborate with their experiences, patterns and scenarios. This knowledge is iteratively improved and reviewed by the community.

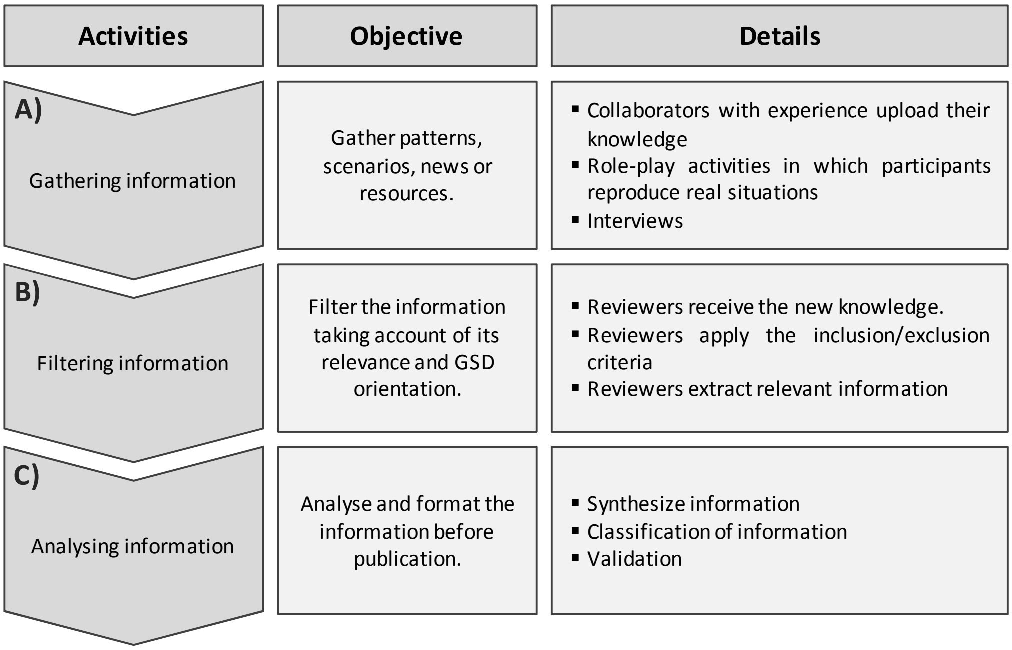


Figure 17. Knowledge acquisition process

Members of the community can collaborate by including their scenarios and associated patterns, based on real problems and experiences in GSD. The process used to gather this information by means of the website is set out in Figure 17, and is made up of the following steps (A, B, C):

A) Gathering information

In order to populate the knowledge base, three methods are considered:

- Interested participants (researchers or practitioners) actively provide any knowledge or experience that could be translated into patterns or scenarios.
- The use of a role-play activity in which groups from several organizations are asked to act how they would behave in a typical work situation. As they do so, the information is captured on Class Responsibility Collaboration (CRC) cards that can be analyzed to identify patterns and scenarios. This method is similar to that followed by Coplien and Harrison [336] to obtain Organizational Patterns.
- Interviews with experts in which they will be asked to provide their experiences in a structured manner.

B) Filtering information

- The submission of new material is automatically notified to a group of reviewers.
- Reviewers examine the new submission, checking that it is consistent with the objectives and that there are no similar cases already stored in the knowledge base (by means of the search engine). Then reviewers apply the following inclusion and exclusion criteria:

  Inclusion criteria: Problems that may appear in a GSD context and can be generalized and applied in similar settings.

  Exclusion criteria: Non-representative problems, issues that depend on personality factors or conflicts that have not come about as a consequence of applying GSD.

- Extract relevant information. Reviewers can modify the content by extracting relevant information.

C) Analyzing information

- Synthesizing information. Data is formatted, and useless or repetitive information is removed.
- Classification of information. This phase involves checking the correctness of the initial data classification.
- Validation. This phase validates the information and makes it available to the community.

4.9 Conclusions

This chapter presents a framework for GSD education based on interactive synchronous and asynchronous simulations. VENTURE framework provides interaction training in GSD by simulating the use of communication and coordination channels in which real world scenarios take place. The framework avoids the need for coordination with other institutions, thus reducing the instructors' workload and the scheduling problems of traditional approaches. It also avoids the difficulty of finding team members of different cultures with appropriate skills and knowledge to carry out the GSD activities. VENTURE framework has been designed with the intention of accomplishing the following objectives:

- To provide students with practical experiences as regards the theoretical GSD knowledge learned in class.
- To provide students with experience in teamwork with people from a foreign culture.
- To provide flexible training which can be run at lower costs, thus removing the difficulties involved in coordination with distant members and reducing instructors' workloads.
- To promote the incremental improvement of the framework with the incorporation of new material, simulations and GSD patterns and scenarios. During the training process, the instructors obtain feedback about the use of the framework and these results can be used to improve the scenarios and the knowledgebase.

One of the main advantages of this framework is its modularity and flexibility, since it can be easily extended by including new simulations and knowledge that could serve to train students in the use of new collaborative and communicative tools.

The design of these simulations is a relative simple task which is accomplished by means of the scenario designer. The architecture also promotes the reutilization of the scenarios and rules. However, the main difficulty of designing scenarios from a GSD perspective (considering Software Engineering problems) is that it is not always easy to gain access to this kind of problems. In order to tackle this challenge, a model based on patterns and scenarios was defined. Patterns can be used to apply specific protocols or solutions to the resolution of specific scenarios. Scenarios also help users to understand the patterns and their context of applicability. The aim of this model is to break a problem down into simpler ones, to package experiences for reuse, and make them available to study real problems recognized by the GSD community.

The identification and classification of different levels of GSD patterns and concrete scenarios have other potential benefits for the GSD community. First, they provide a means of sharing and discovering challenges or problems that members of the community could view as being interesting for their own study. Second, the potential reuse of the patterns and scenarios provides the community with the opportunity to act as a test bed for the evaluation and testing of the patterns and scenarios, in addition to their iterative improvement.

Finally, Virtual Agents provide students with opportunities for self-reflection and self-correction by explaining the consequences and rationales of their actions. Instructors also are able to provide feedback to the students, as they can monitor their activities and communicate with them. In order to guide and automate instructors and students' tasks, a systematic assessment process integrated in VENTURE has been defined, which requires minimal intervention from the instructor. The initial self-assessment makes it possible to assign simulations that are customized to a student's individual characteristics. The training activities can therefore be adapted to a student's specific needs, and to instructors' objectives. Moreover, this adjustment can also be implemented in an iterative manner during the process enactment after executing simulations during each cycle of the process by considering the students' progress, which is monitored by VENTURE in real time.

Chapter 5

Expert Evaluations

Chapter five provided a description of Version 2 of the VENTURE tool which, as shown in Figure 18, was derived from two expert evaluations. The two testing and improvement phases (3 and 4) are explained in this chapter. The sample of experts (including potential users of the tool), provides feedback on the strengths and weaknesses of the VENTURE prototype and VENTURE V1, which were duly used to update the tool in order to create Version 2.

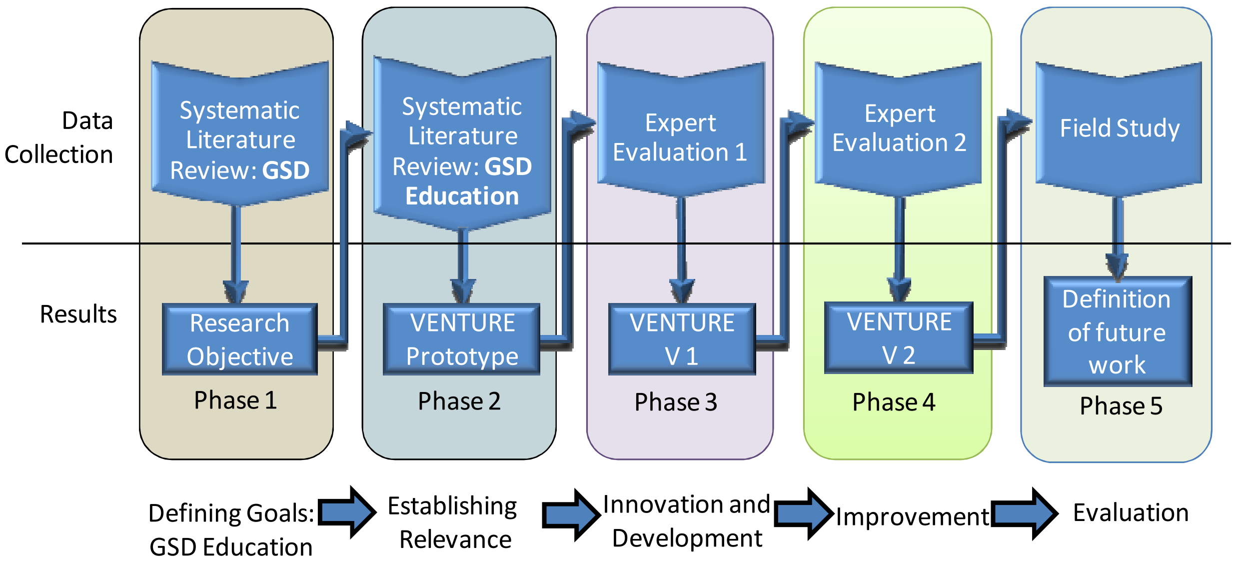


Figure 18. Research methodology

5.1 Expert Evaluation 1

The main objective of conducting the initial Expert Evaluation was to obtain prompt feedback about the suitability of VENTURE for providing GSD training. This evaluation addresses four key research questions:

RQ1: Does the proposed tool help participants to develop the skills needed in GSD?

RQ2: How should this tool be applied in educational settings?

RQ3: Would the tool be usable and effective for training purposes?

RQ4: What kinds of scenarios are suited to such a tool?

5.1.1 Expert selection

An opportunistic sample of four practitioners from three multinational companies participated in the survey, in which selection was based on their expertise in GSD and their availability. As shown in Table 15, two participants were project managers, and two were developers. Their experience in GSD projects ranged from three to eight years.

Table 15. Characteristics of the practitioners interviewed

#	*Nationality*	*Age*	*Experience in GSD (years)*	*Current role*
1	Spain	36	5	Project Manager
2	Spain	38	8	Project Manager
3	Spain	34	6	Developer
4	U.S.A.	35	3	Developer

In addition, a group of eight researchers also completed the survey. Table 16 shows the characteristics of these researchers. As a whole, this population reflects the needs of end-users: practitioners can give their opinion from the perspective of the group designing and delivering the courses (thus allowing us to evaluate the scenario designer, and the applicability of the framework in industrial settings).

Table 16. Background of researchers

#	*Nationality*	*Age*	*Previous Knowledge in GSD*
1	Cuba	26	No
2	Italy	27	Medium
3	Mexico	42	Medium
4	Peru	35	Medium
5	Argentina	26	No
6	Spain	27	Advanced
7	Spain	38	No
8	Uruguay	27	Basic

5.1.2 Procedure

The research questions were answered by applying a survey methodology that included a mixture of open and closed questions (see Appendix III). Prior to answering the survey questions, the participants were given an explanation of VENTURE's architecture and operation. In order to illustrate the chat simulator's operation, the participants were shown a series of snapshots. In these snapshots, a simulated Spanish student, playing the role of a software analyst, chatted with a virtual customer in the elicitation of a set of software requirements. Several stages of the simulation were displayed, showing how the simulator presents cultural and linguistic differences with the help of the Virtual Guide. The interviewees were thus able to form an impression of how the simulator operates.

After the briefing, the participants completed a survey consisting of structured questions with the objective of answering our research questions. The questionnaire comprised yes/no closed questions (to gather data on experience and personal details), and open-ended questions (to elicit opinions about the use of the framework in their companies/universities; i.e., the framework's usefulness and usability). It was intended that an analysis of these responses would act as a guide to refine the framework.

5.1.3 Results

The responses of the survey were analyzed by bearing the research questions in mind:

Does the proposed tool help participants to develop skills needed in GSD?

This question was addressed by focusing on the following points:

Usefulness of the tool as regards training the skills required in GSD

In the words of one respondent: “It is very useful, because in a real project, problems will never happen in a systematic, controlled way; with this environment, you can model many real situations and improve how you deal with them”.

In general, researchers liked the idea that the tool can provide independent, customized training that focuses on specific problems, and also the idea of having a repository of predefined rules and scenarios. In this regard, two respondents suggested that a key factor for the framework to be successful would be for a large repository of different training scenarios to be made available, in addition to a wide set of rules to ease the design of new scenarios based on specific problems.

Weaknesses of the tool and ideas for improvement

The respondents pointed out that real life is very complex; the training scenarios can represent only a small example of the problems that may arise. Moreover, in order to provide training adapted to each individual, with a specific student’s needs in mind, it is necessary to provide a sufficient number of training scenarios, which is not easy. The time required to design training scenarios is viewed as being a very important aspect to take into account in the success of the tool. This suggests that the usability and flexibility of the scenario designer component are critical.

The respondents also believe that in the initial training stage, an explanation should be given on how the tool works to make the users aware of the mechanism: “Users could lose interest when speaking to a machine. If this happens, they will make less of an effort. It is important for users to be aware of this, so that they get as much benefit out of the framework as possible”.

The respondents remarked that the simulations might appear to be artificial if the training scenario was not well designed, thus leading to a subsequent loss of interest on the part of the students. It was therefore suggested that interesting and fluid scenarios should be created to mitigate this problem. The following suggestions were also provided:

- Include hyperlinks in the text. Virtual Agents could provide links to documents.
- Provide situations in which the interaction between real participants could be guided or supervised by the Virtual Guide.
- Include function to iteratively improve a scenario after each training cycle.
- Include a function to pause the simulation.

How should this tool be applied in educational settings?

The following points address this question:

Applicability of the tool in university classes

In general, the researchers considered VENTURE to be suitable for training certain concepts. They agreed that a more in-depth corresponding theory should be provided in traditional university undergraduate software engineering classes to complement the virtual training. However, this add-on would depend on the course learning objectives.

Applicability of the tool in companies

This question was answered by practitioners only, most of whom thought that, once instituted in a company, the framework would be useful not only for training in GSD, but also for other kinds of interactions between people, such as customer support. Two practitioners stated that the main problems companies encounter when organizing courses are: the difficulty in finding available experts in GSD; the time needed to develop these courses; and the organizational difficulties involved in carrying out the courses in the company. Some practitioners agreed that, with a more complete database of problems and linguistic and cultural rules, this framework could potentially be used by inexpert developers. In the words of one practitioner: “In the future, when it has a sufficient amount of information, it will no doubt be a great tool”.

Problems in the application in a company setting

The main concern of the practitioners when applying this kind of training is the time and resources needed. Some comments also related to the operation of the framework were: “Learning in this way could be kind of artificial, but it gives the user the chance to have experiences that it would otherwise be difficult to have”. “Obviously, a simulation is always different from reality, but in this case you can simulate lots of problems that might well appear in real life”.

Time required for this kind of training

Although there were a variety of responses to this question, the practitioners seem to agree that two sessions a week would be reasonable. While the duration of the simulation should be close to real conversations, they concluded that 20 minutes (the average time suggested in the responses) might be the right length of time for a chat simulation. That could vary depending on the particular scope of the training scenario.

One researcher suggested that the scenarios should focus on specific objectives, rather than focusing on the time taken to complete a scenario. So for example, the scenario would not finish until it had reached a certain phase at least, or when it had generated a certain number of mistakes. This led to the decision that time management was an important skill in GSD that we wanted users to learn.

Would the tool be usable and effective for training purposes?

Participants answered this question by highlighting the following three points:

Look and feel of the Virtual Environment

In general, practitioners and researchers feel that the use of the chat simulator is similar to any other chat application; this is appropriate from the point of view of the user’s experience. One practitioner indicated that when interacting with Virtual Agents, users are not going to react in the exactly same way as they do with real people, but at the same time he agreed that Virtual Agents are perfectly valid for teaching purposes.

Time saving benefit and limitation

From the point of view of the instructor, one practitioner noted: “It minimizes the instructor’s workload…, it can reproduce difficult situations…, I think that the tool will be useful as support but the main concepts of the subject must always be taught by a teacher”. Moreover, as regards the scenario designer, one respondent showed concern about the design of the scenarios: “The main problem is how to adapt the tool so that it provides suitable suggestions and feedback to the user”.

Engagement and motivation

Researchers were asked about their interest in using the tool and how motivated they would be to do so. The intention of this question was to obtain feedback that could help to identify aspects of the simulator that needed to be improved if it was to be more readily and completely accepted.

Most of the researchers valued the framework’s capabilities as regards providing independent training. As the tool responds to a real training need in a practical way, they found that engaging with the tool was instructive and motivational. In their words: “When a developer confronts a global project for the first time, he may suffer from stress and fear of failure…”, “being able to practice beforehand and learn how to interact can reduce these problems in the initial stages of the project”. One respondent also remarked that iterative improvement of the training scenarios would be necessary to create scenarios with sufficient quality to be attractive to real users.

What kinds of scenarios are suited to such a tool?

The following point provides answers to this question:

Training scenarios and skills

The participants were asked which scenarios and skill training they would like to be added to the current framework. Having only been given an example of the training scenario that consisted of a requirements elicitation meeting, the practitioners made the following suggestions for future development:

- Meetings to ask clients for specific information.
- Client support activities. Dealing directly with clients about issues that may arise.
- Interaction with a remote developer to solve a problem with the software.
- Asking an expert about a particular technology in order to solve a problem.
- Asking a client for access to their systems and for details of their requests.
- Provision of training in the use of the specific tools employed in the company.
- Dealing with an angry customer who is concerned about the software.
- Dealing with a colleague who has done a bad job.
- Real cases previously documented by the company.

The researchers, meanwhile, proposed the following interesting ideas:

- Formal meetings with a manager.
- Informal meetings to exchange information that could be interpreted by the user.
- Resolution of an urgent situation that must be dealt with in a short period of time, where there is no room for mistakes.
- Delivery of software to a client. Providing its users with training and assistance.
- Asking for clarification of requirements to solve a certain problem.
- Discussing a reported error with a client. In some cases these errors are not really errors as such; these errors are sometimes complicated to reproduce.

5.1.4 Conclusions of Expert Evaluation 1

Conducting the Expert Evaluation helped to determine the strengths of the prototype framework and identify how it could be improved at an early stage.

The results showed that the VENTURE framework has the potential to be engaging and useful for training in GSD processes, as long as the training scenarios are designed in a dynamic way, and provided the scenarios present truly representative situations. In response to the positive encouragement received, we plan to develop a revised framework that incorporates some new requirements as suggested by the experts.

The framework was considered by both practitioners and researchers to be applied both in universities and in companies to provide GSD simulations. Respondents agreed that the kind of training provided by this proposal can help to focus the GSD training on specific objectives that can indeed be systematically reproduced in an accurate manner.

Using both researchers and practitioners proved to be useful in gaining ideas from a variety of perspectives. For example researchers, in general, liked the idea that the framework can provide independent, customized training that focuses on specific problems, that they could engage in when needed (without having to attend a formal class). As the framework reflects and simulates a real training need, they found the experience instructive and motivational. On the other hand, practitioners took a more pragmatic stance focusing on the time and resources which could be saved through this form of training. Both the flexibility of the tool as regards reproducing different kinds of scenarios and the independence of the training were seen as being of great value.

Some of those surveyed even found that the framework would be useful for designing other kinds of interactions that are unrelated to GSD training.

Both groups of participants felt that the use of the Chat Simulator is similar to any other chat application. However they did note that interacting with a Virtual Agent does have its limitations as users are not going to react in exactly the same way as they would do with real people, although the approach is perfectly valid for teaching purposes.

The main weak points reported were the problem of providing a sufficient number of training scenarios and the anticipated time required to create new training scenarios. Another problem reported by a participant was related to the motivation of the students: they might lose interest when interacting with Virtual Agents. “If this happens, they will make less effort. It is important for users to be aware of this, so that they get as much benefit out of the framework as possible”. Moreover, the Virtual Guide might be too intrusive on some occasions and stop the flow of conversation. This means that the course designer (instructor) must seek to obtain a careful balance between realism and training scopes, and should give the student just enough guidance to complete the scenario within an appropriate amount of time. Feedback provided by the Virtual Guide must also be carefully planned in order to avoid disturbing the student too much. Some effort should also be made to introduce students to the context of the training scenario and the operation of the environment prior to their first interaction with it.

Finally, the analysis of the results obtained indicates that the framework meets the training objectives. Indications are that VENTURE has the potential to give the student increased confidence to carry out effective communication in GSD, and is able to reproduce realistic scenarios and provide feedback that focuses on specific skills. An analysis of the responses to the survey provided useful insights that address our research questions. Viewing responses from a group of potential users in terms of current strengths and weaknesses of the concept will inform the next phase of development.

5.1.5 Limitations

This work has some limitations with regards to construct, internal and external validity of the evaluation [338]:

- **Construct Validity**: There may be some bias in the responses since all the researcher participants were from the same course and university. What is more, only one type of research instrument was used: a survey with a limited number of questions.
- **Internal Validity**: There may be some bias, since the participants handed their responses directly to the researchers undertaking and reporting this study.
- **External Validity**: We cannot generalize these results owing to the small sample which is not necessarily representative of the population of practitioners and researchers likely to use VENTURE. However, the sample does represent a cross section of countries, experiences and user groups.

5.2 Expert Evaluation 2

The main objective of this evaluation was to obtain feedback about the VENTURE prototype chat simulator and scenario designer and assess whether the training tool has the potential to provide GSD training, and can therefore be effective in giving the student increased confidence to carry out effective communication with people of different cultures and languages. Table 17 lists the research questions and associated objectives that guided this evaluation.

The specific sub-goals associated with these objectives are to:

- Elicit advice and feedback in order to improve the training tool.
- Identify future training scenario designs for the training tool.
- Identify improvements that it might be necessary to make in order to adapt the training tool to such scenarios.
- Establish a basis that can be used to determine the best scenarios for each student.
- Evaluate the difficulties of applying VENTURE in universities and companies.

Table 17. Objectives of the research questions

Research Questions	Objective
RQ1: Has the tool the potential to help participants to develop the skills needed in GSD?	To assess whether the tool is a good method for learning and training GSD skills overall, and particularly skills related to communication and collaboration when dealing with multicultural participants.
RQ2: How should this tool be applied in educational settings?	To address the feasibility of the application of the tool in both companies and universities.
RQ3: Has the tool the potential to be effective for training purposes?	To discover whether the interaction with VAs can actually be motivating, adapted to reality, and effective in its learning objectives.
RQ4: What kinds of scenarios is this tool suited to?	To define the various fields of application of the tool in order to provide training in areas related to GSD.

In this study, an evaluation of the usability and usefulness of VENTURE using a method similar to Heuristic Evaluation [339] was conducted. The Heuristic Evaluation method involves a small but varied group of experts representing different and relevant fields of expertise. Each expert used the tool and provided feedback based on their experience. Using Heuristic Evaluation implies the following advantages [340]: the application of recognized and accepted principles; intuitiveness; usability early in the development process; effective identification of major and minor problems; rapidity; and usability.

The need for this evaluation arose because of the complexity of GSD, which makes it difficult to evaluate the adequacy of the training tool as regards providing accurate training in GSD-related skills. The opinions of and the feedback from experts is consequently needed before using the tool in real educational settings. In our case, we do not only focus on usability factors, but also on the orientation and adequacy of the tool's use for training purposes in GSD-related topics.

5.2.1 Expert selection

Participants were selected from complementary disciplines: research, teaching and GSD practice. Although there is no consensus as to the optimal number of experts required to evaluate a system [340], Hwang and Salvendy [341], indicate the ideal number for this kind of study to be between eight and twelve. In this case, twelve participants from ICGSE and six from Lero, were selected.

As is shown in Table 18, the evaluation was conducted by eighteen participants. They were from thirteen different countries: Argentina, Brazil, Finland, Germany, India, Ireland, Italy, New Zealand, Nigeria, Netherlands, Nepal, Pakistan and USA. The participants were selected by considering their availability and their experience in GSD, and they had, on average, more than seven years of experience in GSD.

Table 18. Evaluation sample demographics

	No. Participants	Average of experience (years)
Lero	6	4.2
ICGSE	12	10.16
Total	**18**	**Average 7.9**

As is depicted in Figure 19, eight of the participants (44%) classed themselves as practitioners and researchers, as they were involved in both fields. Moreover, three participants (17%) were purely practitioners and the other seven (39%) were purely researchers.

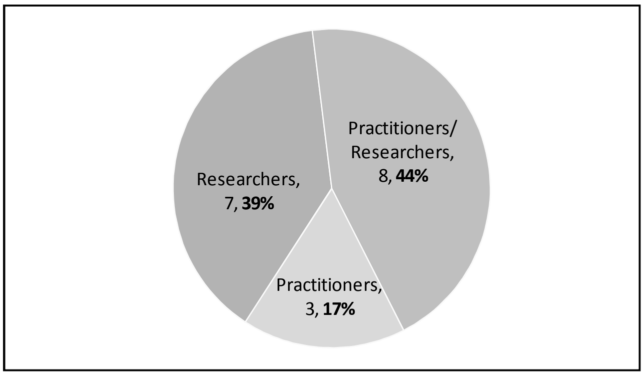


Figure 19. Profession of the participants

The respondents who are currently practitioners indicated that they play the roles of project manager, general manager, researcher (in two cases), marketing manager and quality and process manager. The headquarters of these companies were located in the following four countries: India, Finland, Brazil and Germany. Two of the researchers are university professors who impart subjects directly related to GSD.

Table 19. Characteristics of the companies of the practitioners interviewed

Company size	IT dept size	Countries involved in GSD
100000	10000	5
150000	130000	10
4500	3500	9
4500	3500	6
18000	1000	20
100	95	1

Table 19 describes the size of the respondents' companies, the size of their IT departments, and the number of countries usually involved in GSD activities.

5.2.2 Procedure

The Heuristic Evaluation procedure used in [339] has been applied, in which a small number of experts representing varying disciplines individually evaluate the interface [342].

The chat simulator and the scenario designer were used by the participants in order to answer the research questions. A questionnaire containing open-ended and closed questions was filled in by the participants who were mostly experts in GSD. Each participant spent approximately 20 minutes using the tool and then 10 minutes completing a short questionnaire.

Prior to answering the questionnaire, the participants were given a verbal explanation of the objectives of the evaluation and the operation of the chat simulator and the scenario designer. They then executed a short training scenario using the chat simulator on their own.

The first step before starting the simulation consisted of displaying some background information to orientate the participant. In this step, we explained that during the simulation the participant would play the role of software developer and that he or she would be required to interact with a virtual Spanish developer.

During the simulation, the Virtual Colleague (simulating a native Spanish speaker) made some typical Spanish mistakes when speaking English. The Virtual Guide explained these problems to the students and also corrected some of the student's incorrect interactions. For example, they were advised when they were too direct in addressing a problem.

On completion of the simulation, VENTURE's scenario designer option was also shown to the participants. They were shown how the scenarios were defined by dragging and dropping actions, and how the rules were added. Participants examined the definition of the scenario that they had executed, discovering its settings and the rules that they had fired. Finally, the participants completed the questionnaire and provided their feedback on both the training and design elements of VENTURE.

5.2.3 Results

These results present not only the survey results, but also how the participants interacted with the training scenario after analyzing the log of their interaction.

During the execution of the scenario, only five participants used the option to skip the current phase of the conversation. On some occasions this was owing to the fact that they attempted off topic conversations with the VA and did not then know how to continue the flow of the simulation.

Most of the participants (twelve) fired at least one cultural rule (as depicted in Table 20), which allowed them to check how the VC corrected their inappropriate interaction. No linguistic rules were fired owing to the participants' high level of English.

Table 20 also shows the participants' responses to the following questions in the questionnaire which they had to evaluate from 1 (strongly disagree) to 5 (strongly agree):

- Q1. The Chat Simulator is easy to use
- Q2. The Chat Simulator can be used to train people to recognize linguistic differences in GSD
- Q3. The Chat Simulator can be used to train people to recognize cultural differences in GSD
- Q4. The Chat Simulator is effective in correcting student's mistakes (e.g. use of ambiguous language or inappropriate behavior)
- Q5. The information in the final report provides students with useful guidance – i.e., they can, given this information, reflect on how they can improve their linguistic and cultural communication.
- Q6. The Designer can highlight typical cultural and linguistic mistakes in GSD
- Q7. I enjoyed using the Chat Simulator
- Q8. Using a Chat Simulator is a good way to train individuals (as a concept)

Table 20. Results of the open questions

Participants	Cult. rules fired	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
1-ICGSE	2	5	4	3	5	3	5	5	5
2-ICGSE	1	5	3	3	2	4	3	4	5
3-ICGSE	2	4	3	4	4	3	3	5	4
4-ICGSE	2	4	2	3	3	3	4	4	5
5-ICGSE	2	5	3	3	4	2	4	4	3
6-ICGSE	2	4	3	3	3	4	3	3	3
7-ICGSE	0	5	3	4	NA	5	4	4	3
8-ICGSE	2	5	4	4	3	4	3	5	4
9-ICGSE	1	5	5	5	4	3	5	5	5
10-ICGSE	0	5	3	4	4	4	4	4	4
11-ICGSE	0	5	4	3	3	2	5	5	4
12-ICGSE	0	3	4	3	3	4	4	4	4
13-Lero	3	4	3	3	5	5	4	5	4
14-Lero	1	5	3	3	4	4	4	5	5
15-Lero	1	4	4	2	3	4	4	4	2
16-Lero	0	4	4	4	3	3	3	3	4
17-Lero	4	5	4	4	4	4	5	5	4
18-Lero	3	4	4	4	4	3	5	4	3
Median	**1**	5	3.5	3	4	4	4	4.5	4

The results show that the respondents agree that the simulator is easy to use (Q1). Most of them enjoyed the experience (Q7) and believe that it is a good way to train GSD skills (Q8).

Q2, Q3 and Q4 were not given such high values, mainly because more participants interacted properly and did not fire any cultural or linguistic rules during the simulation, thus signifying that they could not be sure about the effectiveness of the tool in training these problems. Those who fired at least one rule valued these points positively. Although Q6 was also related to the previous questions, in this case the question refers to the Designer, and in this case the participants could see how the rules are added and modified.

The information gathered in the final report was reported as a point for improvement (Q5) and some participants provided feedback that will be shown in the following subsections. The remaining questions and answers that were gathered are summarized in this section, grouped by the research questions to which they apply:

RQ1: Has the tool the potential to help participants to develop the skills needed in GSD?

This question was addressed by the following points:

A1. Do you think this training method can be effective in accurately training the specific cultural and linguistic differences in GSD?

Most of the ICGSE respondents who fired cultural rules answered affirmatively: “It seems to have the potential to be flexible enough to cover many aspects”. One participant remarked that the Chat Simulator is interesting as complementary material although it could not be the core of the training method.

One respondent who did not fire any cultural rules stated: “It is a good start but it would need more rules and enhancements”, although this respondent understood that the scenario is very limited in time, and a real scenario would have more semantics and deal with realistic problems.

The opinions of the Lero participants were quite similar: “I like the way Maria (VC) corrects what is going wrong… when I was too direct she pulled me up straight away. She was also quick to point out Raul’s (VA) incorrect use of words”. One of them suggested its application in other courses related to Software Engineering.

RQ2: How should this architecture be used in educational settings?

This research question is answered by the following questions which were answered according to Appendix III:

A2. Do you think it would be feasible to train students/members of your university/company by applying this environment?

Most of the respondents answered this question affirmatively, and two of them remarked that before its application in educational settings it should be improved in order to make it more robust. In the words of one of them: “I could see it being used in classes by students and it could help to train them in cross cultural inequity”.

One ICGSE respondent found it more interesting to apply the simulator in companies rather than in universities. The main reason for this is that he found it specific to the training of certain problems that can appear in the specific settings of each company. One respondent at the conference thought that it would be more useful if this training could be conducted with real students in real time. However, the scope of this training tool is to avoid the inconvenience of such approaches.

RQ3. Has the tool the potential to be effective for training purposes?

Effectiveness for training GSD was addressed in the following questions:

A3. What problems did you experience while using the Chat Simulator?

Most ICGSE participants indicated that they had not had any problems during the simulation. One participant, admitted that he did not always know how to continue, but there were no great problems as he had the possibility of skipping the phase.

However, some of them suggested that improving the robustness of the simulator would avoid misunderstandings, since in those cases in which the student interacts in an unexpected manner (using words that were not taken into account in the scenario definition), the VA answers by using default conversational knowledge. This knowledge is not directly related to the context of the conversation and some respondents indicated that they had received strange answers in these cases.

Two participants attempted to challenge the tool and found some inconsistencies when they tried to be rude by using offensive words that were not taken into consideration in the scenario definition. They therefore received good answers from the VA without receiving appropriate feedback, such as “Avoid using expletives. This type of language can be offensive”. One of them also tried to be very direct in communication, but during a phase of the conversation in which the scenario did not expect such a rude answer, because it was out of context. However, we do not consider this as a problem, since this user was challenging the system with the expectation of receiving feedback concerning an inappropriate interaction at any moment rather than it interacting in a logical manner. One of the users also expected to receive answers to questions that he posed which were not considered in the scenario definition.

With regard to the use of the simulator, one participant pointed out that a lot of typing is involved in these simulations, and suggested that the VC could sometimes speak directly to the VA in order to communicate specific problems, thus reducing the student’s effort.

Another participant pointed out that the special words or expressions used to say the same thing can vary from one person to another and it is difficult to handle all of them. Another respondent also explained that “one has to follow a certain sequence of text inputs”, whereas he would have liked to decide the topic to be dealt with and have a freer conversation.

A4. Can you suggest any points for improvement or new features that you would like to be implemented in the simulator?

Two respondents interacted in a manner that was not taken into account in the design of the scenario, and the Virtual Agent did not therefore understand them. These respondents proposed automating the recognition of synonyms by using a thesaurus. In relation to this point, the recognition of language abbreviation was also suggested. For example, two different participants used the abbreviation "ws" to refer to "web service", but the scenario definition did not expect this abbreviation. Also in relation to this problem, a Lero respondent suggested the use of a machine learning technique to improve the training scenarios. However, this is not currently within our scope.

One of the proposals consisted of providing the students with feedback regarding their inappropriate interactions in real time before sending the message. The VC would thus act as a spellchecker or a grammar checker in a text editor, and the correction would be more dynamic.

In those cases in which the student does not know how to interact, it has been pointed out that a better option would consist of providing an explanation about how to continue rather than simply skipping the phase with the "Next Phase" button.

An ICGSE respondent suggested that when a rule is fired, more context information could be shown. For example, "when the VA says that a particular Friday is a holiday in Spain, he could give a list of public holidays in Spain that could let me learn something about Spain".

With regard to implementing classes in universities, one of the respondents proposed allowing the student's mistakes to be run through again at the end of the simulation, and replaying them as a means to reinforce learning. The generation of a global report was also suggested with the same aim. This report could group the common pitfalls committed by all the students in the class, and this would be useful for the teacher in order to explain the most common problems that the students in that class have.

With regard to the final report generated, which summarizes the results of the simulation, one respondent suggested providing more background information in order to reinforce learning. For example explaining typical characteristics of the culture of the VA or giving more information about why certain rules were fired. Moreover, the possibility of simulating team work with other students was also suggested, thus allowing them to participate in a common project, discussing problems guided by the VC.

Finally, the development of a broader set of training scenarios was also suggested as a means to provide instructors with sufficient material to make the application of the simulator more interesting.

A6. Can you suggest any points for improvement or new features that you would like to be implemented in the Designer?

One of the respondents who challenged the Chat Simulator by using inappropriate words, suggested including a blacklist in the Designer that could be used to detect inappropriate interactions during the simulations.

Improvements to the layout and the rule management were also suggested. In the words of one ICGSE respondent "a more user-friendly presentation of the rules could be useful". In this initial version of the Scenario Designer, the cultural and linguistic rules are added without any connection to the cultural and linguistic databases. This is a further improvement that will be made in the near future. In this respect, some participants suggested exactly this same idea.

One suggestion was related to controlling different levels of difficulty in the design of the scenarios. However, we do not plan to make any changes as regards this point, since it is already possible to design several similar scenarios that could be separately modified by considering different levels of difficulty without increasing the complexity of the Designer.

The possibility of providing multiple paths in the timeline of the simulation was also suggested. Students could therefore choose the course of the interaction by means of their answers, thus making the conversation more dynamic, from the point of view of one of the respondents. However, we do not plan to implement this option since the complexity of designing training scenarios would increase. Other minor ideas, such as increasing the number of avatars in order to cover a wider range of cultures in the simulations, were also suggested.

RQ4: What kinds of scenarios is this tool suited to?

This research question is answered with the following question:

A5. Upon considering the training scenario in which you have participated, which other training scenarios do you think it would be interesting to design for the training of specific GSD problems or skills?

The participants suggested the following kinds of training scenarios:

- The coordination of meetings to show differences in time perception.
- Setting deadlines to train differences in perception of pressure or hierarchy.
- Reaching the day of a deadline.
- Starting a project (introducing people).
- Dealing with a crisis. Dealing with serious problems in a project and conflict resolution. E.g. how to say that the due date will not be met.
- Task scheduling.
- Discussions between people with different competence levels. E.g. experts and new comers.
- Discussions about how to refactor code.
- Notifying a failed acceptance test.
- Querying an implemented feature that does not map onto a requirement as expected.
- Obtaining information about specific technologies from experts.

5.2.4 Discussion

The results of the evaluation are summarized below in order to show aspects of VENTURE that the experts liked, those that they did not like, and ideas for improvement from both a technical and content perspective. Technical areas refer to development and architecture, whereas the content is related to the underlying depth and usefulness of the scenarios.

Good points

From the technical perspective, the researchers liked the idea that the tool can provide independent and customized training focused on specific problems in GSD. As the tool responds in a practical manner to a real training need, they find it instructive and motivational. In their words: “When a developer confronts a global project for the first time, he can suffer from stress and fear of failure…”, “being able to practice beforehand and learning how to interact can reduce this problems during the initial stages of the project”.

In terms of student experience, both the practitioners and researchers felt that the use of the Chat Simulator is similar to any other chat application. The use of VAs is seen as a good option to simulate GSD settings and provide students with feedback. One practitioner indicated that when interacting with the VA, students are not going to react in the exactly same way that they do with real people, but agreed that VAs are perfectly valid for teaching purposes.

From the content side, the tool was well valued as it minimizes the instructor’s workload and the time required to organize courses and seek experts, when compared with traditional methods. The flexibility of the proposal to be tailored to a specific culture’s needs was seen as one of the tool’s strong points, and this made the respondents believe that it would be feasible to apply the tool in educational environments. In support of this idea, they suggested a broad set of training scenarios.

After using the tool, some practitioners agreed that with a complete database of problems and linguistic and cultural rules, this tool would have the potential to be used by inexpert developers. In the words of one practitioner: “In the future it can be a great tool with the sufficient amount of information”.

Weak points

The principal weak points reported were mainly concerned with VENTURE’s content. To enable training in a wide variety of skills and to tailor that training to the individual student’s needs would require extending the current number of training scenarios. Another problem was that the time required to create new training scenarios should be as short as possible, which would require a larger set of rules in the databases.

Some participants found inconsistencies in the VA’s answers when they interacted in a rude manner or when they attempted off topic conversation, and suggested further work in this direction to improve the robustness of the tool before its application in educational settings. One of the participants also expected to receive answers to questions that he posed that were not considered in the scenario definition.

The participants suggested that future training scenarios should be tailored and configured to specific cultural needs, since the training scenario used in the demonstration was too generic.

Finally, some participants also pointed out that the special words or expressions used to say the same thing can vary from one person to another and it is difficult to handle all of them, and it would therefore be appropriate to improve the robustness in this respect.

Ideas for improvement

After using the tool, participants suggested the following points, which will be a part of the improvements to be implemented in the near future:

Technical improvements

- Improve the usability of the Scenario Designer by automating some of the tasks related to the management of the rule database.
- Implement the automatic recognition of synonyms and abbreviations during the interactions.
- Support for the automatic detection of inappropriate or offensive interactions.

Content improvements

- Improvement of the final report in order to provide more detailed information about the specific problems that the student had during the simulation.
- Implement the database of cultural and linguistic problems, in addition to the e-mail simulator.
- Create a set of realistic training scenarios by considering realistic GSD problems.

5.2.5 Conclusions of Expert Evaluation 2

The results of the expert evaluation satisfy the aims of the study, by providing clear and useful answers to all the research questions. The method explored whether the tool has the potential to help participants to develop the communication skills needed in GSD, and to better understand how should this tool might be applied in educational settings. This method was effective as regards improving technical aspects and obtaining insights into how to develop valuable content for the courses.

The results indicate that the prototype made a positive impression on the participants. The diversity of the population participating in the evaluation, including researchers, practitioners and teachers, provided a variety of opinions and focused on different aspects of the proposal, such as the user interface, the usability, effectiveness for training cultural and linguistic problems in GSD and its applicability in university classes or companies. The analysis of the results suggests that the main objectives of the tool may be fulfilled as detailed in the answers to our research questions.

The general outcome of the study is that VENTURE has been shown to provide useful and meaningful scenarios with which to train practitioners in GSD, but that it needs a broader knowledge base of real problems in GSD in order for it to have a wider application. As a result of this study, a knowledgebase was developed and made available to the community interested in this area so that interested parties may collaborate with their knowledge in the form of GSD patterns and scenarios that appear.

5.2.6 Limitations

This work has some limitations as regards the construct and internal validity of the Heuristic Evaluation [338]. With regard to **construct validity**, the two evaluations conducted in this research were focused solely on the Chat Simulator and the Scenario Designer. Future evaluations are planned to test asynchronous interactions.

The training scenario used in the evaluation did not consider specific problems of the student's culture, since it was oriented towards participants at an international conference in which multiple cultures were involved. Another limitation of this training scenario is related to the time constraints, as we needed to evaluate the tool without disturbing the participants too much.

With regards to **internal validity**, since the participants handed their responses directly to the authors of the study, we are aware there may be some bias in how participants answered the questions.

5.3 Conclusions

This chapter has presented two Expert Evaluations conducted at different stages of the development which served to obtain Version (2) of the tool. This version has been used in the field study to test whether the new implementation is an improvement on the initial design. The objective was to address possible problems of the design at an initial stage and make the tool more effective as regards its GSD learning purposes. The initial expert evaluation acted as a proof of concept of the framework, while the follow-on Heuristic Evaluation with experts in the knowledge domain, provided early feedback that anticipated challenges in the domain. During the evaluation process and the analysis of this information, new ideas from the training and learning perspectives arose. Specifically, the evaluations generated the following ideas for improvements:

- With the objective of improving the just-in-time features [343] of VENTURE, the final reports generated after the execution of a training scenario may be improved by providing additional material. Videos, resources and advice related to the skills that the students need to improve can be provided depending on their performance.
- With the aim of providing accurate assessment, an accurate procedure considering the automatic evaluations and students' self-assessments was added to the framework. The objective is not only to evaluate the students' performance but also to determine to what extent the students improved their GSD skills.
- With the objective of facilitating the generation of training scenarios, a community website was developed (http://global.lero.ie/community). Researchers and practitioners can now collaborate with their expertise in GSD by contributing with GSD patterns and scenarios. Patterns define problems and strategies on how to tackle them. Realistic GSD scenarios in which the patterns occur are also gathered in such a manner that

they can be used to design new training scenarios aimed to train specific patterns by means of VENTURE. The following kinds of patterns are considered:

- Cultural patterns: tendency to always respond “Yes”, attitude to authority and respect, use of communication styles (direct or indirect, formal/informal, succinct/elaborate), deaf effect (reluctance to ’hear’ bad news in some cultures), etc.
- Linguistic patterns: False-friends (when a word looks or sounds similar in two different languages, but differs in meaning).
- Other patterns regarding coordination, communication and software processes. E.g. patterns regarding requirements engineering or project management in GSD.

The method followed made it possible to evaluate the framework from different perspectives in an iterative manner. The initial expert evaluation appraised the framework as a concept, and guided the development of the first prototype, Version 1, whereas the second evaluation assessed Version 1 of the tool. The feedback from the experts was again used in the development of Version 2. The next chapter presents Version 2 of VENTURE, which is followed by a Field Study that evaluates VENTURE as a training and assessment tool. This final evaluation will test whether the tool actually delivers real learning.

Chapter 6

VENTURE Tool

Once we had described the framework, it was then necessary to prove that it was possible to develop useful GSD simulations by applying the framework, and VENTURE Tool was used for this purpose. The architecture of Venture Tool has been designed in order to provide customizable GSD simulation based training oriented towards cultural, linguistic and GSD related problems. These customization capabilities facilitate the design of accurate scenarios based on real world problems. The development considers all the components of the framework presented in the previous section, including a chat and an email simulator. This development is based on the .NET platform (.NET Framework 4.0).

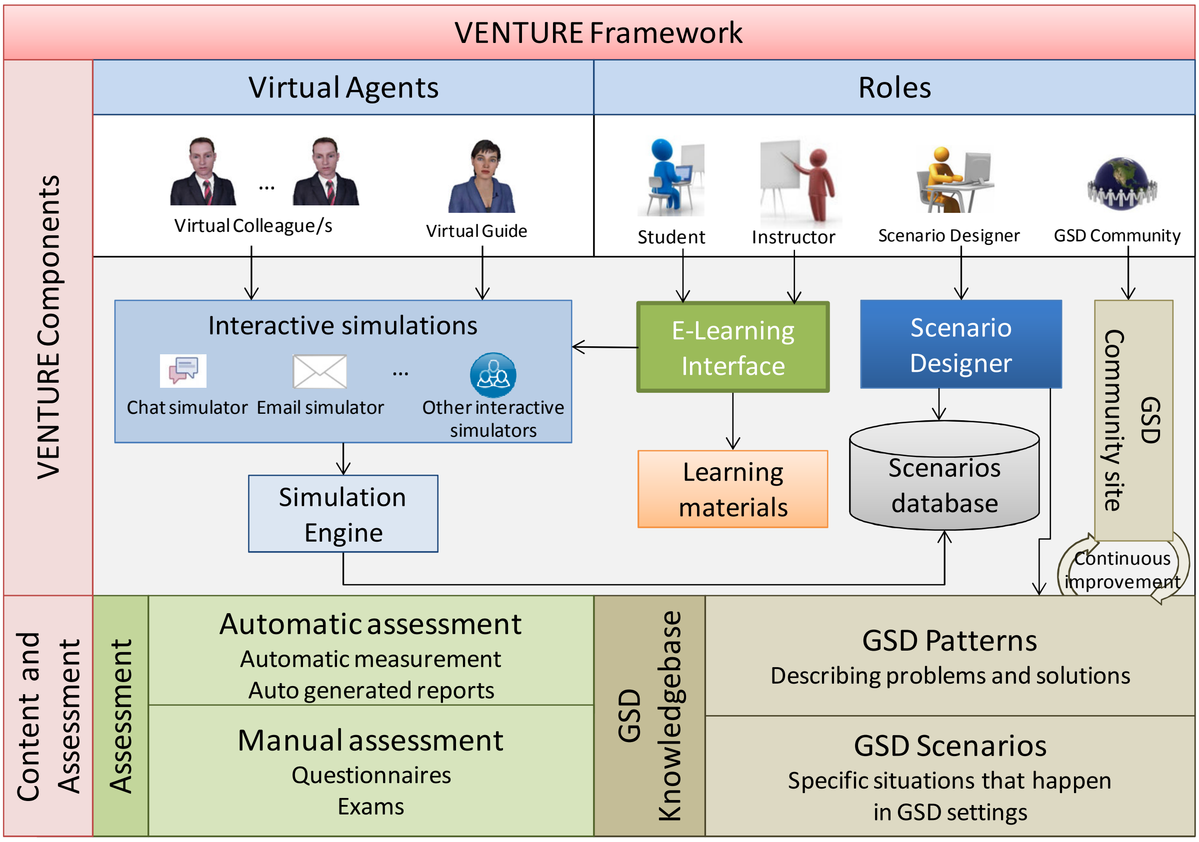


Figure 20. VENTURE Framework

The design of this framework is intended to cover the training needs discovered in the state-of-the-art chapter, and the subsequent implementation of the framework covers these requirements. Firstly, the analysis of the literature study led to define the requirements shown in Table 21, which guided the design of both the framework and the tool presented in this chapter. After analyzing the results of the Expert Evaluation, the requirements gathered in Table 22 subsequently served to define the latest version of the tool (Version 2).

This chapter describes how the different modules of the framework (summarized in Figure 20) have been implemented in this tool by providing solutions to the requirements gathered in the state-of-the-art section and from the Expert Evaluations. The description of the different modules presented here follows the same structure as in the previous chapter.

Table 21. Requirements from the state-of-the-art study

Requirement	Description
RQ1	Provide simulation based training: Simulation solves many problems found in traditional training environments, making the training more independent and focusing on specific problems. It allows common and repeatable experiences to be reproduced for each student.
RQ2	Minimize interactions with real participants: Interaction with Virtual Agents avoids these interactions.
RQ3	Focus on interaction: Communication and collaboration have proved to be a major issue in GSD. This implementation is based on chat and e-mail simulation.
RQ4	Provide training in the skills required in GSD: communication, culture, language and soft skills required in GSD interactions. Interactions with Virtual Agents that speak and show gestures serve to teach language and cultural differences.
RQ5	Contextual training: Simulations must have the ability to deal with different topics and provide rigorous orientation towards cultural and communication difficulties based on real world problems.
RQ6	Provide immediate feedback: The Virtual Guide provides opportunities for self-reflection and feedback.
RQ7	Provide adjustable training adapted to the students' needs: the scenario designer makes it possible to create customized training simulations.
RQ8	Cultural awareness focused on cultural dimensions. The scenario designer allows a predefined set of rules, organized into the different dimensions, to be applied in the simulations.
RQ9	Accessibility and availability: Courses are always available for access through the web by means of an e-learning platform.
RQ10	Provide automatic assessment: Interactions are tracked and an automatic report is provided at the end of the simulation, including a separate score for each skill trained. This allows us to both minimize the instructors' effort and provide the students with rapid feedback.
RQ11	Tracking and monitoring capabilities. The e-learning platform provides surveys and questionnaires, and the students' performances in simulations are available for both students and instructors.
RQ12	Scalability: The architecture allows the system to be easily adapted to a variable number of students.

Table 22. Requirements from Expert Evaluations

Requirement	Description
RQ13	Usability and user's experience: The possibility of pausing the simulations and including hyperlinks in the simulations so that Virtual Agents could provide links to documents was considered.
RQ14	Detect and penalize expletives. A blacklist will detect offensive language.
RQ15	Recognize different ways of expressing the same thing. Automatic match of synonyms.
RQ16	The time required to create new training scenarios should be as short as possible. This is addressed by automating some of the tasks related to the management of the rule database.
RQ17	Provide detailed assessment report about the specific problems that the student had during the simulation.
RQ18	Ability to provide a broad set of real world problems. A knowledgebase including real world GSD patterns and scenarios was designed in order to cover this requirement.

6.1 Roles

The interfaces presented in this chapter are used by the different roles that are defined in the framework. Both roles and interfaces are depicted in Figure 21. The following roles are defined in the implementation:

- **Student**: The students use the simulators to interact with Virtual Agents and confront typical GSD problems. An e-learning platform provides them with access to theoretical material and simulations. Automatic assessments and interaction logs can also be accessed by students through the e-learning platform, thus allowing them to reflect on their results.
- **Instructor**: The instructors organize the groups and provide access to the students, assigning them tasks according to their needs. They do this by accessing the e-learning platform, through which they can interact with students synchronously by chat and asynchronously by e-mail, forums and news lists. They can track the students' progress and review their assessment, which is automatically generated during the simulations.
- **Scenario Designers**: These are in charge of creating the simulations or updating existing ones. The scenario designer interface allows them to have access to a GSD knowledgebase in order to apply previous experiences and reuse previous scenarios. Those scenarios that are stored in the database will be available to be assigned to the students.
- **GSD Community**: In order to promote the development of a representative knowledgebase, a community site has been made available to the GSD community. A knowledgebase for gathering patterns and scenarios in GSD has also been implemented and made available to the community. By using this interface, the GSD Community contributes with the knowledge and experiences required to implement accurate GSD simulations.

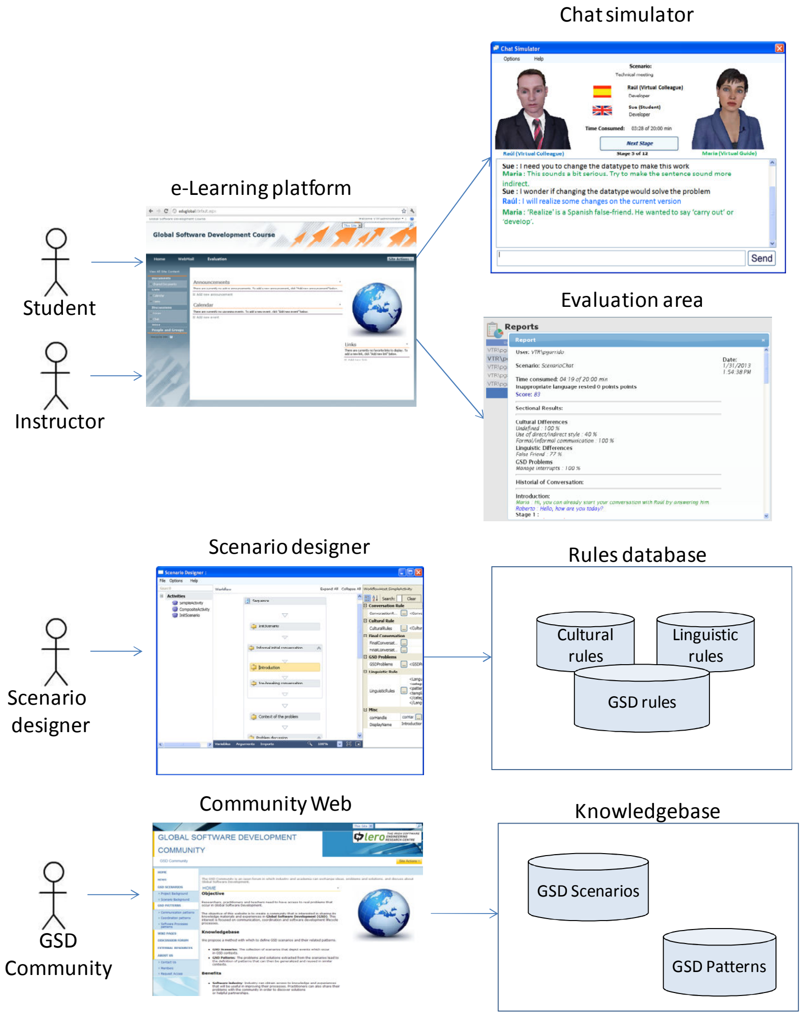


Figure 21. VENTURE Tool

6.2 Virtual Agents

Virtual Agents have been implemented in order to interact with the students by using a chatbot engine. The chatbot engine interprets AIML (Artificial Intelligence Markup Language) [344] in order to recreate real world interactions. AIML is a pattern language in which a *pattern* is a string of characters intended to match one or more user inputs, and a *template* specifies the response to a matched pattern. As the example below shows, the patterns may also contain wildcards, which match one or more words:

```
<category>
    <pattern>How many holidays *?</pattern>
    <template>You must consider two holidays this month</template>
</category>
```

In the case of simulating a chat conversation, Virtual Agents are displayed by means of avatars that can make gestures. This functionality has been implemented by using the avatars provided by Haptek (www.haptek.com). These avatars can also speak by using a text-to-speech engine (Speech SDK 5.1 for Windows), which serves to provide awareness as regards the accents of the different cultures and to improve the listening skills of non-native speakers, which is particularly useful during the provision of GSD training.

The chatbot engine is based on an implementation of the Alicebot engine (http://www.alicebot.org/) which works under the .Net framework. The detection of synonyms has been specifically implemented in order to automatically recognize the different ways in which the students may express the same thing. The chatbot engine has additionally been extended in order to recognize the different gestures that may be made during the simulations according to the context. During a simulation, trainees will interact with two types of Virtual Agents that have been implemented according to the framework:

- The **Virtual Guide** can interact at any stage of the interaction in order to guide the trainee or explain specific problems or circumstances. Moreover, in those cases in which the trainee is detected to have made a mistake, the Virtual Guide will interact to provide immediate feedback. Rules will be defined for this purpose which will consider predefined problems (as explained in the following section).
- The **Virtual Colleague** plays a role in a GSD scenario, interacting with the student by asking and answering questions, providing opinions, ideas or suggestions according to de definition of the training simulation and the AIML information defined at any stage.

The current implementation considers the intervention of a Virtual Guide and a Virtual Colleague in each simulation. According to the framework definition (see Figure 20), more Virtual Agents could participate in the interactions. However, the current implementation only considers a Virtual Guide and a Virtual Colleague as this is sufficient to simulate the scenarios that we can represent.

6.3 Scenario designer

As defined in the framework presented in the previous chapter, a scenario designer has been developed. This designer is needed in order to create any kind of simulation, in this case, chat and e-mail simulations, although the same designer should be valid for new future simulators.

The *scenario designer* is a client application that allows the simulations to be designed in a graphical manner, based on either synchronous or asynchronous means. Windows Workflow Foundation (WWF) has been used to develop the designer. WWF is a framework that enables users to create workflows and execute them. These workflow capabilities have been used to model the interactions as a set of activities composed of a sequence of contextual interventions. It consists of a namespace, an in-process workflow engine, and designers. WWF is appropriate to fulfill the goals of the framework as it provides an extensible model and designer with which to build workflows and to define custom activities that encapsulate workflow functionality. This capability to define custom activities has been used to represent the different kinds of activities of the workflows, in order to consider (in this case) cultural, linguistic and conversational knowledge.

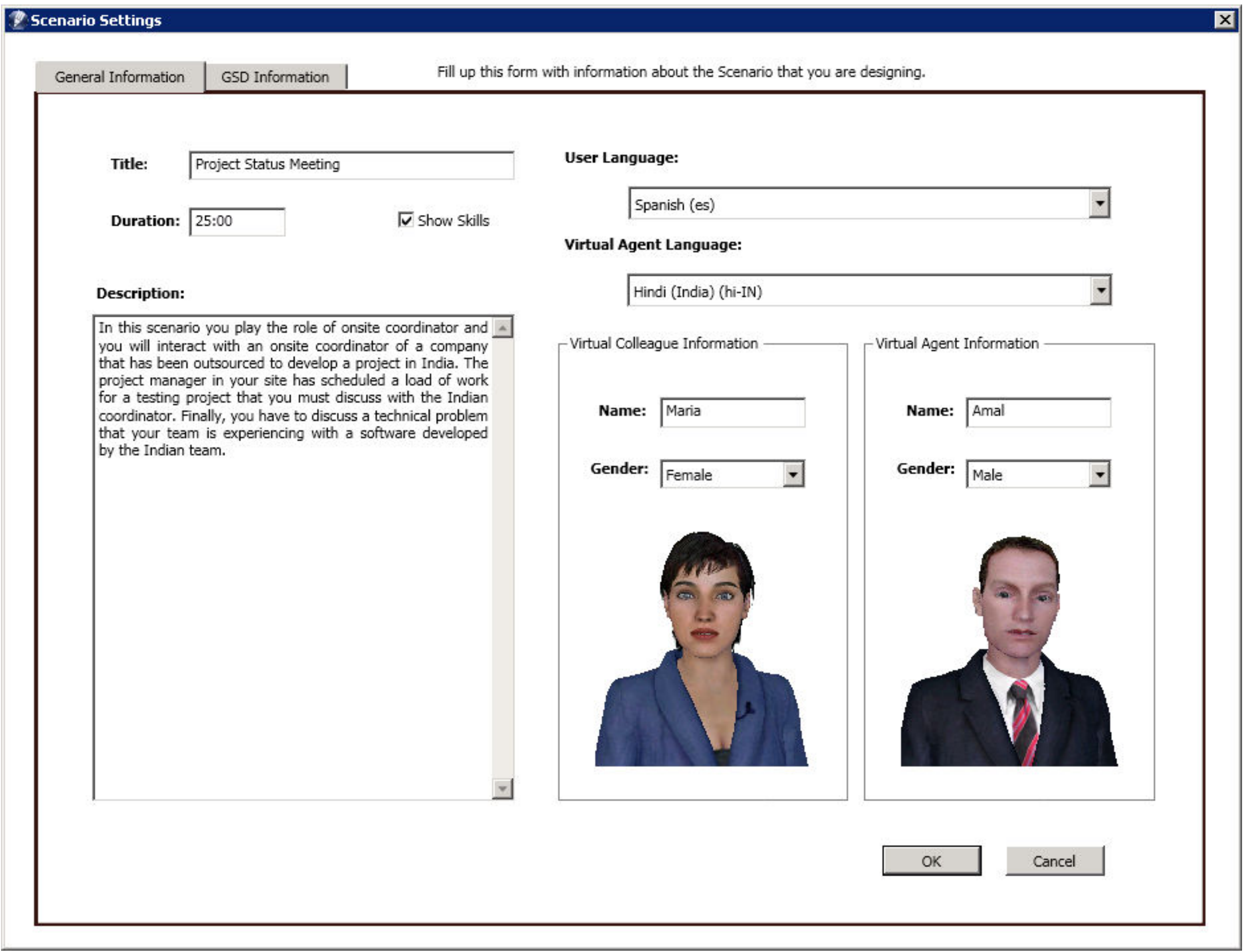


Figure 22. Edition of scenario general details

As a first step, the definition of a simulator requires the context of the simulation to be set up. In order to do this, the scenario settings can be defined as shown in Figure 22, including: a descriptive title, a description explaining the context and objective of the simulation, details about the Virtual Agents involved (including their native languages), and the native language of the student who intends to use the scenario (this can be omitted, signifying that the scenario is intended for any kind of user regardless of his/her native language or nationality).

Other settings related to the GSD project on which the simulation is focused can also be configured according to the settings screen shown in Figure 23, including the following settings:

- Project Settings
  - Project name
  - Domain of the project
  - Sites involved
  - GSD approach: module-based, phase-based, follow the sun
- Roles
  - Role of the student
  - Role of the Virtual Agent
- Project Status
  - Phase
  - Development status
  - Economic status
- Current Deliverable

- Deliverable name
- Deadline
- Status of the deliverable
- Description

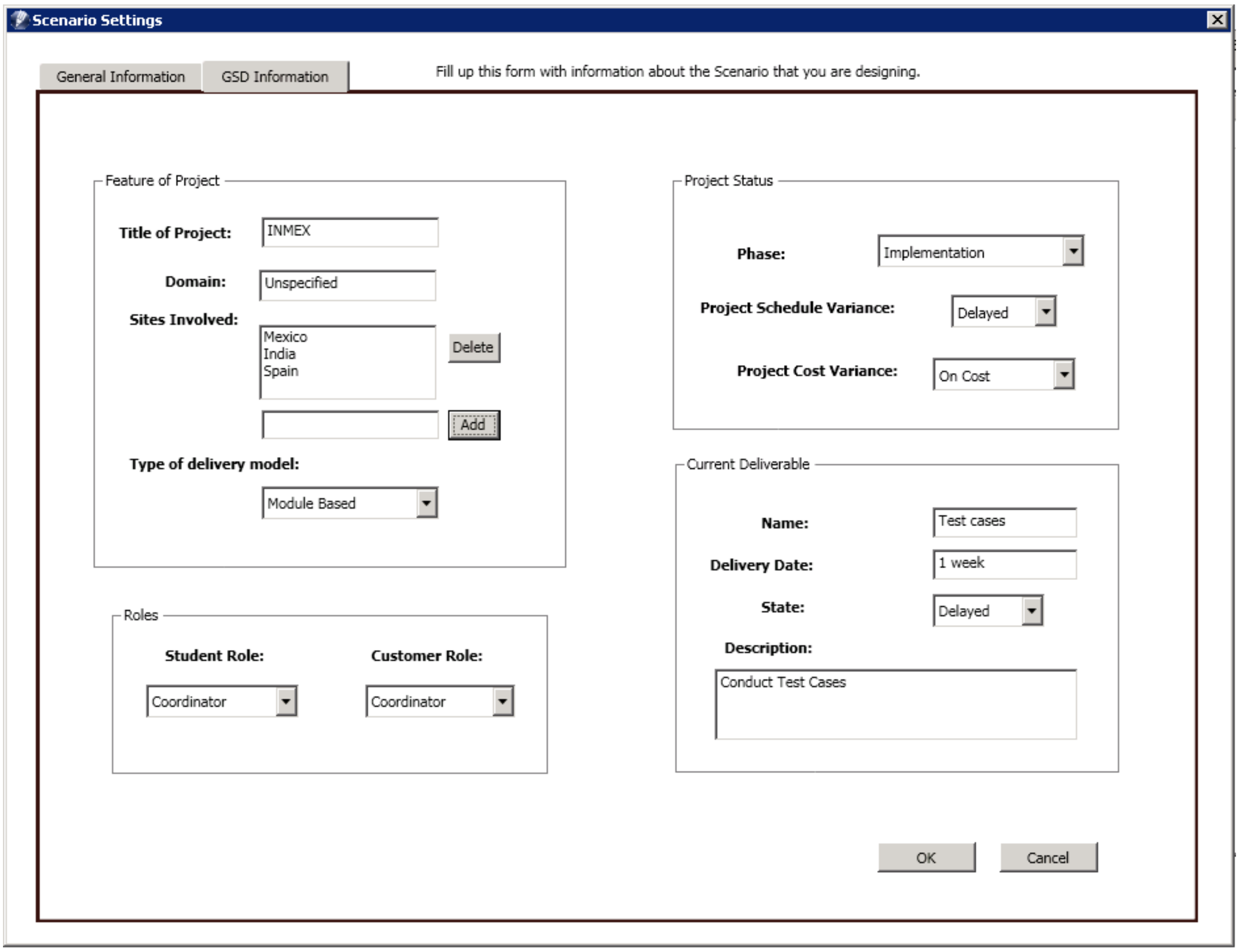


Figure 23. Edition of scenario details on GSD

The main feature of the *scenario designer* is the ability to define a sequential workflow of an interaction (synchronous or asynchronous). Windows Workflow Foundation has been used to provide the ability to define workflows. These workflows contain the different activities that will define the stages of the interaction, and will therefore define a specific context of the interaction. Figure 24 shows the interface of the scenario designer, in which the activities are composed of the sequential workflow. These activities can be added to the definition of the simulator by dragging and dropping them into the scenario sequence.

The activities that are part of the workflow must be defined at a high level of granularity, signifying that each interaction that the student is intended to make must be part of a separate activity containing the specific contextual information.

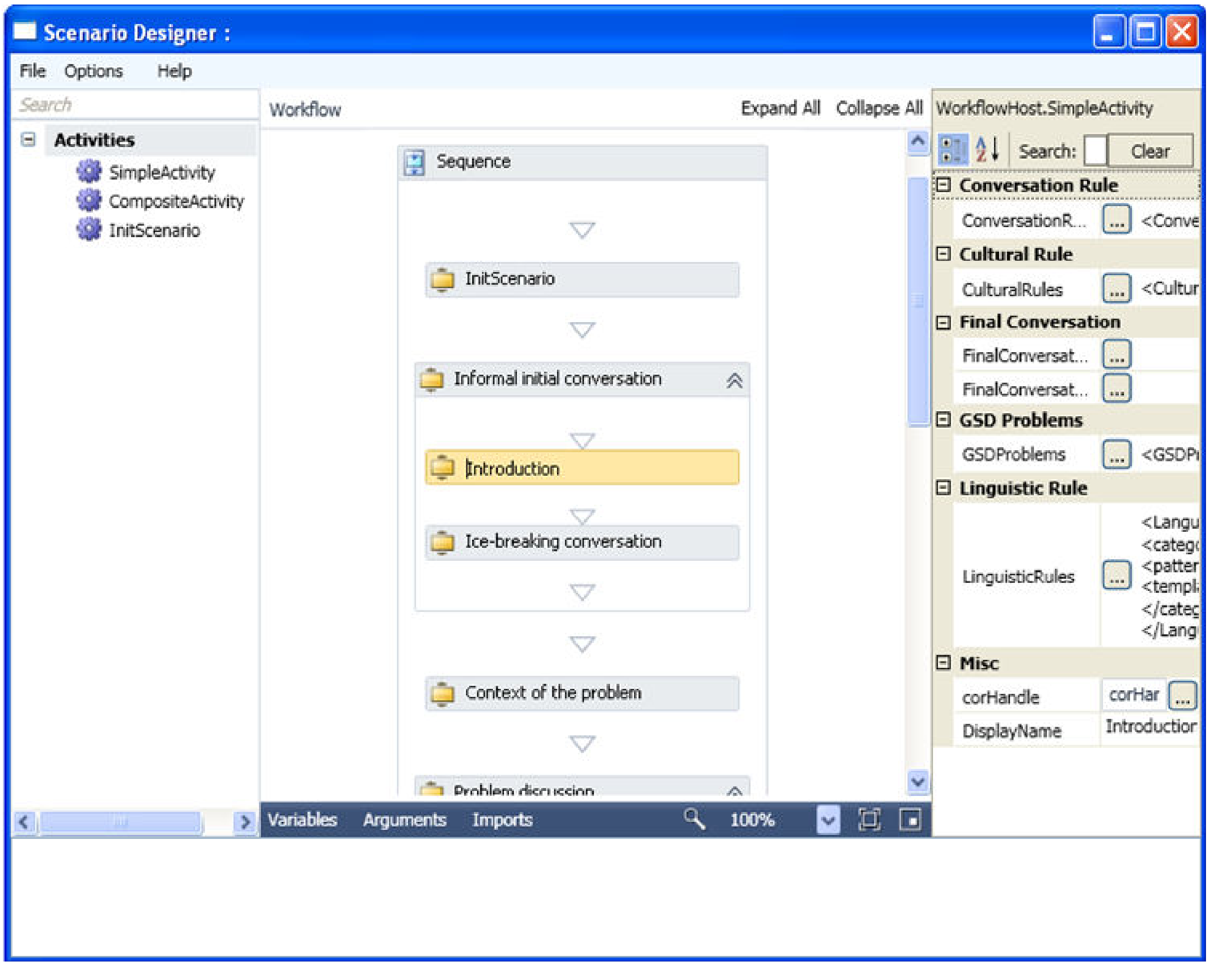


Figure 24. Scenario designer

The advantage of using Windows Workflow Foundation is that it permits custom activities to be defined, which in this case serve to store information that specifically applies to the GSD context of the interaction. Custom activities are defined with the following content:

- *Conversational knowledge*: required for the Virtual Colleague to answer the students' questions. This is defined by using AIML.
- *Cultural, linguistic and GSD rules*: The implementation of the framework has been focused on cultural, linguistic and GSD-specific interaction problems. These rules can be imported from the *Cultural, Linguistic* and *GSD-related problems knowledgebases*. Rules consist of a pattern describing the incorrect interactions expected from the students and a template with the feedback that they will receive from the Virtual Guide when the rule is fired. Scenario designers can therefore manage the knowledgebases by adding new rules or editing the existing ones.

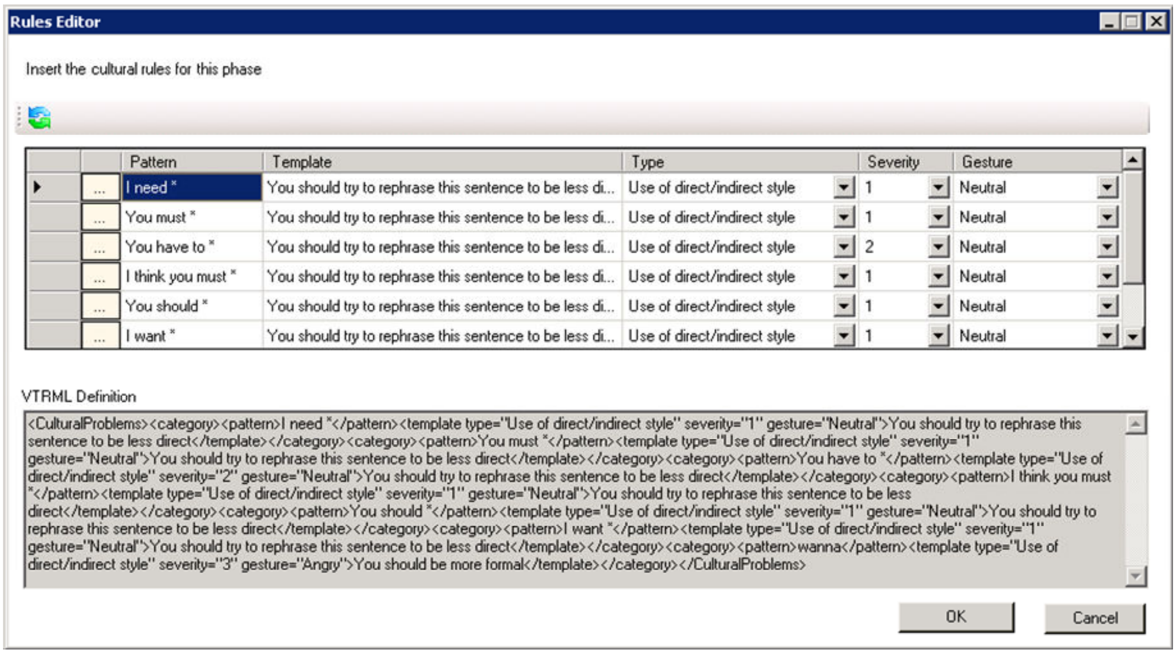


Figure 25. Rules edition

Figure 25 depicts the edition of a cultural rule for which, along with patterns and templates, the following attributes can be defined for each rule:

- Type: type of cultural problem in this case (or linguistic or GSD-related problem in the case of the edition of these kinds of rules).
- Severity: a numeric value used for evaluation purposes, indicating the penalty that the student will receive if this rule is triggered.
- Gesture: each of the Virtual Agent's answer templates can have an associated gesture. For example, the Virtual Agent can show disagreement, concern, sympathy, etc., as is shown in the following example.

A *compositeAcitvity* has also been defined as a means of grouping *simpleActivities* according to thematic areas. Figure 26 shows a fragment of a meeting workflow. The workflow starts with an "Introduction" activity, in which the Virtual Guide will introduce the participants and present the Virtual Agent. This initial activity is completed when the student greets the Virtual Agent, and the "Document request" activity then begins automatically. "Document request" is a *compositeActivity* whose content is detailed on the right-hand side of the Figure 26. It includes the Virtual Agent's conversational knowledge and the cultural and linguistic rules for this context. For each student intervention, the text introduced is checked by considering the patterns defined for cultural and linguistic problems.

With regard to conversational knowledge, this stage begins with the participation of the customer, who is requesting something from the student. The student must then interact by asking what the customer expects from him/her in order to provide an answer based on the AIML definition. In this case, the customer is expecting a negative sentence in order to provide an answer that will finish the stage in order to allow the next one to begin.

If the student provides a positive sentence, the activity includes a cultural rule concerning the Mum Effect which is triggered in order to correct this behavior. In this case, the Virtual Guide encourages the student to answer with a negative sentence in an attempt to postpone the delivery of the document. With regard to language knowledge, this stage also detects the pattern "actual", which is classified as a false friend mistake that the Virtual Guide will correct.

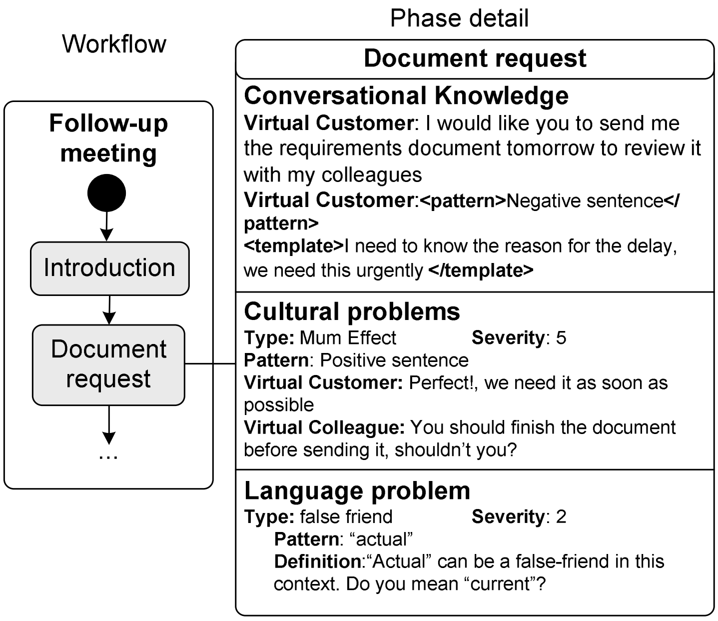


Figure 26. Definition of a training scenario

In this example the simplification "negative sentence" and "positive sentence" is used in order to avoid representing all the positive or negative patterns that the system actually considers. Examples of negative patterns in this context are: *can't *, *impossible*, *difficult*, etc., and the sentence is considered to be positive when none of these patterns are matched.

The definition of these workflows is based on XAML (Extensible Application Markup Language). This is a declarative language that allows all the elements required for the definition of the training scenarios to be defined. Figure 27 shows an example of the definition of a workflow which is based on an XAML automatically generated by the scenario designer, and which can be interpreted and executed by the Workflow Engine. The definition of this meeting considers the indications of Clear et al. [345], in which the structure of a meeting in international collaborative settings is described.


```
<Workflow>
  <Sequence>
    <Phase Name="Introduction">
        <!--Not detailed in this example"-->
    </Phase>
    <Phase Name="Client request">
        <ConvesationalKnowledge>
          <category order="1" gesture="concerned"><!--Start the conversation (order=1)-->
              <template VirtualColleague="VirtualCustomer1">We would need to have our components
                                  working with the exporting functionality by next week</template>
              <template VirtualGuide="VirtualColleague1">We do not have experience in using these
                                               components. You should answer negatively </template>
          </category>
          <category gesture="thinking">
            <pattern>*information * components * time</pattern> <!--The student must mention the
              need for information about the components and also ask for more time-->
             <action>NextPhase</action> <!—Reaching the previous pattern the next phase continues-->
          </category>
        </ConvesationalKnowledge>
        <CulturalRules>
          <CulturalRule type="Assertiveness" severity="3">
            <category gesture="thinking">
              <pattern>DEFAULT</pattern><!—Pattern triggered if the previous one doesn't match-->
              <template VirtualColleague =" VirtualColleague1">You should explain the reasons better
                                                   and be more indirect</template>
            </category>
          </CulturalRule>
        </CulturalRules>
    </Phase>
    <Phase Name="Data layer document">
      <ConvesationalKnowledge>
        <category gesture="concerned"><!--VirtualCustomer1 starts the conversation-->
           <template VirtualColleague ="VirtualCustomer1">Ok, I will send you more information.
              Yesterday I did not receive your document with the proposal for the data layer. Did
              you have any problems?</template>
           <template VirtualGuide ="VirtualColleague1">You should mention that we had technical
                                                   problems</template>
        </category>
        <category gesture="angry">
          <pattern>* problems * after|later|tomorrow|next|soon|days</pattern> <!--The student
            explain the problems that s/he had and when is s/he going to send the deliverable"-->
          <action>NextPhase</action> <!—Reaching the previous pattern the next phase continues-->
        </category>
     </ConvesationalKnowledge>
        <LanguageRules>
          <LanguageRule type="false friend" severity="2">
            <pattern>red</pattern> <!--Incorrect use of the word "red"-->
            <template>"red" is a false friend in Spanish. It can be translated as "network" in
              English. Is that what you mean?</template>
            </LanguageRule>
        </LanguageRules>
        <CulturalRules>
          <CulturalRule type="Assertiveness" severity="3">
            <category gesture="thinking">
              <pattern>* problems *</pattern> <!--The student doesn't give detailed information"-->
              <template VirtualColleague ="VirtualColleague1">You should tell him when you are going
                                                  to send him the report </template>
            </category>
          </CulturalRule>
        </CulturalRules>
    </Phase>
  </Sequence>
</Workflow>
```

Figure 27. Definition of a simulator

The properties of the scenario are also included in its definition. Figure 28 shows a fragment of the definition of the properties for an example scenario.

```xml
<scenarioSetting>
  <title>Technical meeting</title>
  <duration>20:00</duration>
  <description>Description</description>
  <user>
    <name />
    <language>pt-PT</language>
    <role>Quality assurer</role>
  </user>
  <VirtualColleague>
    <name>Raúl</name>
    <language>es-ES</language>
    <avatar>RaulAvatar</avatar>
    <role>Project manager</role>
    <initConversation></initConversation>
  </ VirtualColleague >
  <VirtualGuide>
    <name>Maria</name>
    <avatar>MariaAvatar</avatar>
    <initConversation></initConversation>
  </ VirtualGuide >
  <gsdInformation>
    <projectName>ENGLOBAS</projectName>
    <domain>hospital</domain>
    <phase>Analysis and design</phase>
    <GSDApproach>Module Based</GSDApproach>
    <status>Delayed</status>
    <cost>On Cost</cost>
    <sites>
      <site>Sao Paulo</site>
    </sites>
    <deliverable>
      <name>Ap. Server</name>
      <description>Client application.
     </description>
      <deliveryDate>Next week</deliveryDate>
      <status>On time</status>
    </deliverable>
  </gsdInformation>
</scenarioSetting>
```

Figure 28. Definition of scenario properties

The definition of these scenarios is stored in the Scenarios Database, which stores them for their execution and future reuse. Predefined scenario templates can also be stored in this database in order to provide a mechanism that will facilitate the design of new simulations. These templates can be organized according to the thematic GSD areas that can be covered. For example, it would be possible to define a template with which to train ice-breaking conversations containing predefined conversational knowledge to allow a conversation to be started in the right manner when two specific cultures are interacting in a GSD project.

Developing this designer according to the framework definitions therefore helps GSD scenarios to be designed in an accurate manner, reproducing problems and situations that it would be difficult to reproduce by other means. Moreover, it allows these problems to be reused in new scenarios and consequently speeds up the design of GSD simulations. The cultural, linguistic and GSD-related problem knowledgebases used by the designer are explained in the following subsections.

6.3.1 Cultural problems knowledgebase

One of the most common problems in GSD interactions is related to cultural diversity. Interaction among peers from different cultural backgrounds entails certain difficulties that may delay and even lead to the failure of GSD projects [8]. The *cultural problems knowledgebase* is a repository which contains the description of cultural problems and recognizes differences that may affect communication and collaborative work.

The knowledge included is classified by the kind of problem being dealt with, along with a description of the difficulty, the relative seriousness of making each mistake and the feedback that the user will receive when the problem appears in the context of a conversation. The information contained in this knowledge can be managed by the *scenario designer*.

The different cultural problems are also classified according to the cultures involved. The problems will occur to a different degree or in a different manner depending on the cultures interacting. This knowledgebase therefore requires an in-depth study of the state-of-the-art concerning the problems that each particular culture may have in this respect.

6.3.2 Linguistic problems knowledgebase

Students and practitioners involved in GSD projects are usually non-native English speakers who must additionally interact with peers who, in many cases, also lack the language skills required in globalized environments [4]. It is therefore necessary to provide them with training on their main problems but also to make them aware of the problems that can be committed by their peers. The *Linguistic Problems Knowledgebase* is a repository that contains the description of the linguistic problems that can affect communication when participants interact with a non-native language.

As in the case of cultural knowledge, this information is also managed by the *Scenario Designer*, which classifies this knowledge according to the kind of linguistic problem that may appear, along with a description of the difficulty, the relative seriousness of making each mistake and the feedback that the user will receive when the problem appears in the context of a conversation.

These problems are classified according to the languages involved, as the kinds of mistakes will depend on the participant's native language and the language used for the interaction. The population of a trusty linguistic knowledgebase requires an in-depth study of these common problems.

Grammatical inaccuracies committed by the students during the interactions are also immediately corrected by using third party dictionaries and grammatical correctors for the target language.

6.3.3 GSD-related problems knowledgebase

The *GSD-related Problems Knowledgebase* is a repository of problems, issues or challenges in GSD that is classified by the following skills, which were obtained from the Systematic Literature Review on GSD education, and are classified according to the participants' nationality:

- Performance in the use of synchronous and asynchronous means of communication.
- Ability to communicate effectively using a common terminology and language.
- Informal communication and improvisation skills.
- Knowledge of language, cultural and ethical issues.
- Leadership and conflict resolution skills.
- Time management skills.
- Ability to think from the perspective of the other side and understand people from different backgrounds.
- Managing ambiguity and uncertainty. Ability to evaluate information critically.

- Knowledge of negotiation skills and contract writing in a common language.
- Collaborative work skills.
- Skills to gain the interlocutor's confidence and trust.

6.4 Interactive simulators

Both synchronous and asynchronous means of communication are commonly used in GSD. Of the various kinds of simulators considered during the definition of the framework, two simulators were eventually developed: a chat simulator and an email simulator. These interactive simulators are accessible through the e-learning platform and are presented in the following subsections.

6.4.1 Chat simulator

A *chat simulator* was developed with the goal of reproducing synchronous interactions. Windows Communication Foundation (WCF) has been used in order to communicate with the *workflow engine*. WCF is a tool that supports distributed computing when services have remote consumers. Clients can consume multiple services; services can be consumed by multiple clients in a concurrent manner. It is an interoperable platform that uses a WSDL (Web Services Description Language) interface which does not depend on the platform. Messages written by the student in the chat simulator are sent to the *workflow engine* which will then send back an answer according to the definition of the simulation.

The interface has been developed by using Windows Presentation Foundation (WPF). WPF is a graphical system created by Microsoft and released as part of .NET Framework 4.0 in order to render user interfaces in Windows-based applications. It provides a programming model with which to build applications and separates the user interface from business logic. WPF employs XAML, an XML-based language, to define various user interface elements.

By using the *chat simulator* students can chat with the Virtual Colleague and receive feedback and guidance from the Virtual Guide. Its main interface is depicted in Figure 29, which shows an example of an interaction in which a student from the United Kingdom called Sue is interacting with a Virtual Colleague from Spain called Raúl. Both of them play the role of software developers discussing, in this case, a technical problem. Maria is the Virtual Guide who is providing feedback and guiding Sue.

The students only need to communicate with the Virtual Colleague by writing in the text box. If the students get stuck or do not know how to interact, they can use the "Next Stage" button in order to pass to the following phase of the conversation, and receive a corresponding penalization.

During the simulation, the total number of stages and the current stage are displayed, and GSD related information regarding the context of the simulation is also displayed: project settings, sites involved, status of the project and information about the current deliverables that are being addressed.

In this example Maria provides Sue with feedback after she has interacted in too direct a manner, and after solving this problem, Raúl makes a common mistake often made by Spanish people when speaking English. This problem is also explained by Maria. Maria can therefore not only provide immediate feedback about a student's problems, but can also explain customs or problems in the Virtual Colleague's speech which may require clarification. The Virtual Guide (María) can also explain concepts regarding GSD or provide information that is specific to the scenario in order to guide the student in the topics s/he has to cover. In order to promote the students' effective interaction, the simulations have a limited amount of time.

Both Virtual Agents can make gestures according to the context of the conversation and speak aloud by imitating the typical accent of their nationality in order to promote awareness. This functionality is provided by Haptek (www.haptek.com) which provides the avatars that have been used in this implementation.

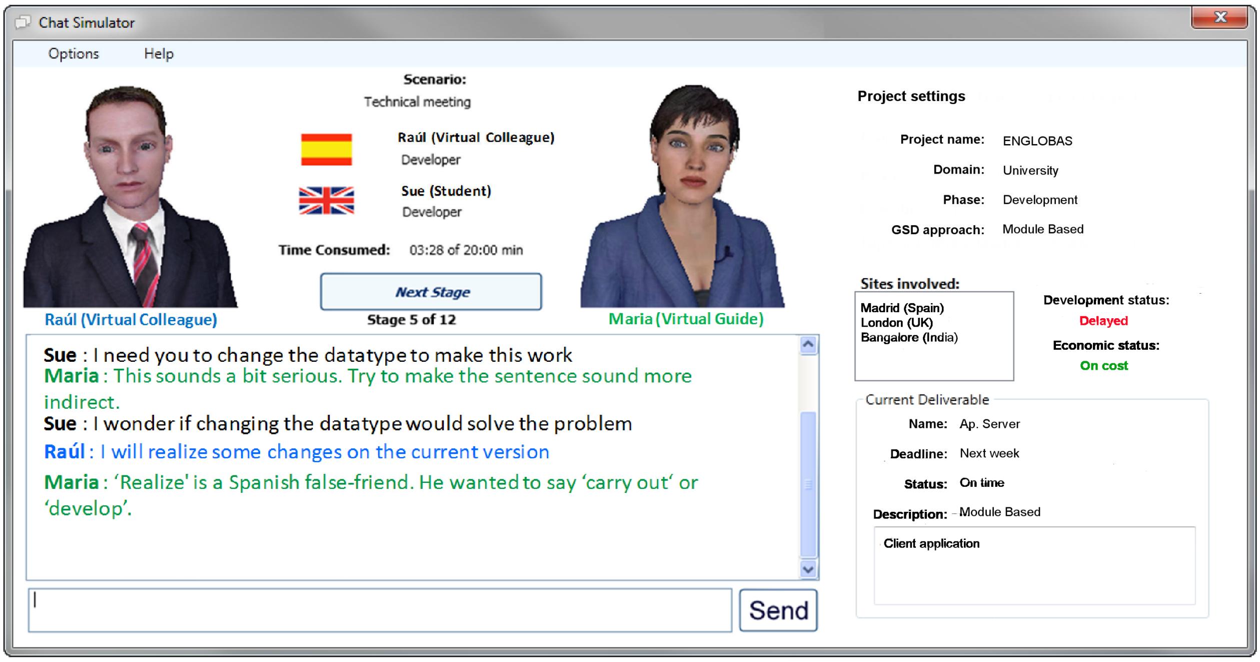


Figure 29. Chat simulator

Finally, a grammatical corrector was included, so that students can easily find the right expressions while they write in order to check inaccuracies before sending the message. This functionality is provided in order to facilitate its use by non-native English speakers, who are the main group of potential users considered by the framework.

6.4.2 Email simulator

An e-mail simulator was developed with the aim of reproducing asynchronous interactions, as this is the most frequently used means of asynchronous communication in GSD [346].

The email simulator is integrated into the e-learning platform. By using this simulator, students can receive messages from Virtual Agents and reply to them. This component can also be used by students and instructors to send and receive emails. Realistic delays in the answers can therefore be simulated. The procedure used to analyze the emails sent by the student is similar to that used in the chat simulator. In this case, the Virtual Guide will immediately notify the student with feedback and guidance by means of a message.

Figure 30 shows an example of an email simulation. In this scenario, the Virtual Colleague (Raúl) starts the interaction by asking the student (Sue) to send him a document. It is assumed that in a previous chat simulation, both Sue and Raúl have discussed a set of test cases and Sue has a document available. In his interaction, Raúl has committed a typical Spanish mistake, which is automatically explained to Sue by means of the Virtual Guide. When Sue replies, three mistakes are detected by the Virtual Guide and explained to Sue, so that she can reply again after correcting them.

It was quicker to develop this simulator after we had the *chat simulator* as it uses the same *workflow engine*. The same communication mechanisms (using Windows Workflow Foundation) are therefore valid for both simulators, and the design of the training scenarios is similar. Following the definition of the framework to create an *e-mail simulator* therefore had the advantage of the reusability of development and concepts. This has led us to believe that developing new simulators by applying the framework definition will similarly be a relative easy task.

Figure 30. Email simulator interface

6.5 Workflow engine

The *workflow engine* is part of the server side of the architecture and is responsible for executing the simulations. It does this by communicating with the chat and email simulators by sending and receiving messages based on Windows Communication Foundation (WCF).

The main component of the workflow engine is the Workflow Orchestrator, which guides the execution of the workflow defined with the *scenario designer*. Windows Workflow Foundation (WWF) has been used for this implementation. WWF provides a rehostable and customizable workflow engine that includes features with which to save the state of the workflows in a database. This is particularly useful for simulating asynchronous interactions in which communication may be delayed by days.

Simulations are launched when the student initiates a chat or an email conversation through one of the simulators. The simulators send a WCF message that is received by the *workflow engine*. At that moment, the *workflow engine* reads the definition of the simulation and establishes the parameters such as the time limit, description and GSD context of the scenario. It then starts orchestrating the sequential execution of the different stages by using the Workflow Orchestrator. Messages typed by the trainee are sent via WCF to the *workflow engine*, which interprets each stage into which the simulation is divided.

A Chatbot System processes the natural language by interpreting the AIML defined in these stages. The AIML language contains the information required to simulate an intelligent conversation with the Virtual Agents. The chatbot system used is an implementation of AliceBot (http://alice.pandorabots.com) for .NET.

As mentioned in the Scenario Designer section in this Chapter, the so called stages are defined as custom activities in WWF. These custom activities are defined by the *conversational knowledge* that will be used by the Virtual Colleague and the *cultural, linguistic and GSD rules* that will be used by the Virtual Guide. At each stage, the Workflow Orchestrator extracts the conversational knowledge, together with the cultural, linguistic and GSD-related rules defined. Having loaded this data, the process followed by the Workflow Orchestrator to execute a custom activity consists of the following steps:

1. Check whether there is a linguistic rule that can be triggered. If this is the case, the message returned will contain the feedback that the Virtual Guide will provide the student with. If no linguistic rules are fired, go to step 2.
2. Check whether there is a cultural rule that can be triggered. If this is the case, the message returned will contain the feedback that the Virtual Guide will provide the student with. If no cultural rules are fired, go to step 3.
3. Check whether there is a GSD related rule that can be triggered. If this is the case, the message returned will contain the feedback that the Virtual Guide will provide the student with. If no GSD related rules are fired, go to step 4.
4. Apply the chatbot engine in order to return a response considering the conversational knowledge definition. This response, after being received by the simulator, will be reproduced by the Virtual Colleague and will execute the next custom activity, going back to step 1.

During this execution, the engine automatically creates a report about the student's performance, considering the rules that he/she has fired and the severity of making each mistake. After completing all the activities in the simulation, the *workflow engine* stores this report and the log of the conversation in a database, thus permitting the instructor to review it later.

The workflow engine permits the simulators to provide training in a transparent manner, providing automatic assessment as required by the framework and allowing the students' actions to be tracked in real time.

6.6 E-learning platform

The main component of the architecture is the server side, from the point of view of the students, and the e-Learning platform from that of the instructors (see Figure 31). This platform has been developed using Microsoft SharePoint and facilitates the task of separating the contents into various courses oriented towards the different problems in GSD.

Microsoft SharePoint is a Web application platform that comprises a multipurpose set of Web technologies closely integrated with the Office suite. It provides central management, governance, and security controls, and it can be used to provide intranet and extranet portals, file management, collaboration, social networks, enterprise search and business intelligence.

It is a scalable platform that permits an easy web based customization and also provides a tool for in-depth customizations (SharePoint Designer). It is possible to extend its functionality by developing complementary applications and web parts. Web parts are sections that can be developed independently to provide new functionalities and which can be inserted onto a page. The implementation presented uses the free deployment called Windows SharePoint Services 2007, which includes all the features required for the scope of the framework.

Each course defined on this platform has its own theoretical contents which are adapted to the course objectives, and its own training scenarios.

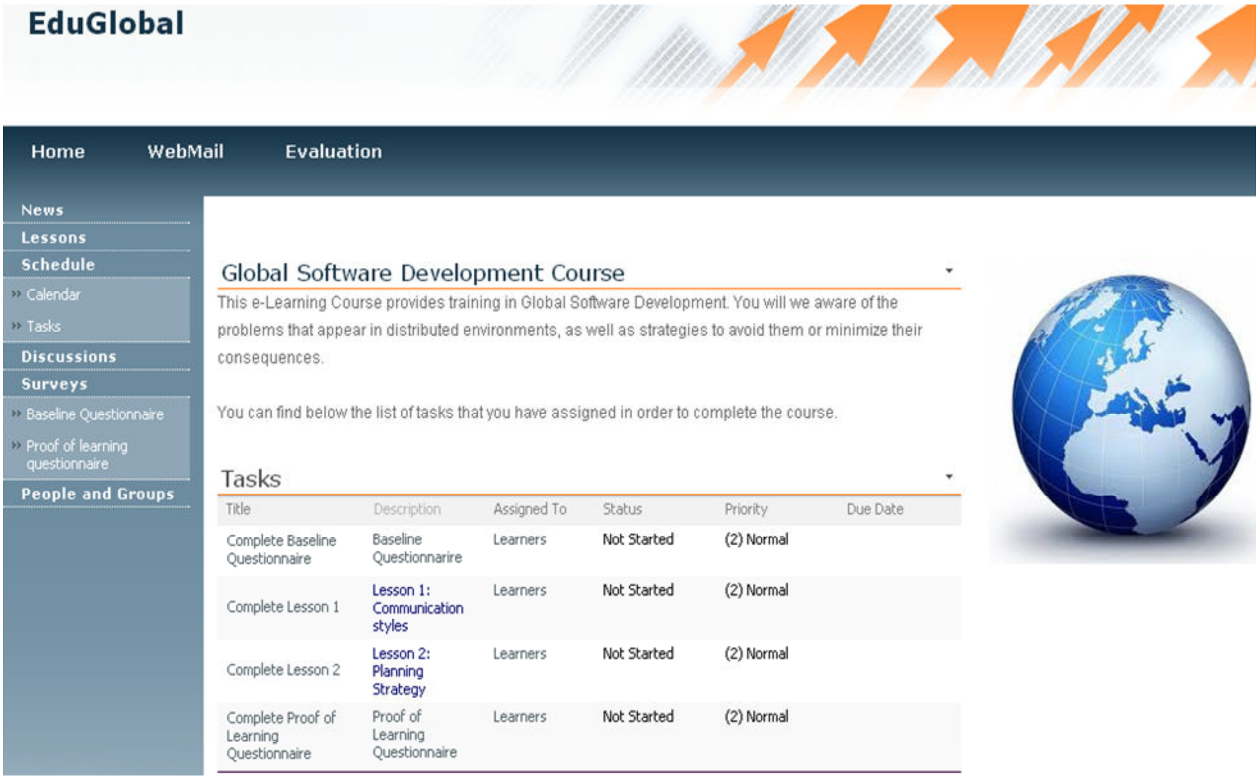


Figure 31. e-Learning platform

The e-learning platform provides the course resources and informating down facilities [332] by allowing the students to know their evolution and state. It also provides informating up facilities, since instructors can know, in real time, the status of the students' tasks and actions, and communicate with them. The following components are integrated into this platform:

- **News**: including information about the course: new assignments, deadlines, new materials added, etc.
- **Resource repository:** in which both the theoretical lessons and the simulators and artifacts are made available to the students.
- **Task area**: which serves to control and schedule the practical activities. The students can also upload deliverables in this area.
- **Forum and wiki**: through which students and instructors can keep in touch.
- **Evaluation area**: through which students can do exams, fill in questionnaires, and review the evaluation and the instructor's comments for these activities.
- **Chat and email communication**: with instructors and students.

The objectives of the framework are fulfilled with this platform, as instructors can schedule tasks and activities for the students and follow their evolution. They can manage the GSD training courses and be aware of the status of the tasks, whilst being able to communicate with the students. On the other hand, students can access the training materials and manage their assigned training scenarios, in addition to being able to access the information about their performance, deadlines, qualifications and historical actions. Students can also know which team members are online, receive news, share artifacts or fill out questionnaires and take exams.

6.7 Assessment

One important aim of the VENTURE framework is that of minimizing the assessment effort through the automation of the assessment procedure. This kind of assessment is important for the application of the framework in both universities and companies, and considers the student's self assessment based on questionnaires. The students are initially assigned a baseline questionnaire (see Appendix V) with which to identify their current skills. Based on the online questionnaire, the students' cultures and skills are determined on a scale of 1-5. Instructors can also obtain more information by conducting personal interviews with students . This allows the students to be categorized and appropriate learning materials to be assigned to them.

The next step is to study training materials and execute training simulations. For each training simulation, the students will complete a questionnaire at the beginning and another one at the end of the training session in order to measure their improvement after the simulation (see examples in Appendix VI and Appendix VII). These assessments also serve to adjust learning by the early identification of the students' problems and needs. After each lesson, the students receive the instructor's comments and future lessons can be adapted accordingly.

During the execution of the training scenarios, the students' interactions are automatically tracked and assessed. The *assessment database* stores the detailed scores of the participants after conducting the training scenarios, in addition to the scores from tests and questionnaires that are related to the courses. The logs of the interactions are also stored for review purposes. This information is made available to the students to enable them to know their own performance and also to the instructors in order to track and evaluate the students. The following subsection details the automatic assessment process applied in detail.

6.7.1 Automated assessment

Figure 32 shows a report indicating a student's score. In order to obtain the student's overall score, the average of the scores of all the simulations on the course are calculated. The results of the questionnaires that are conducted in relation to the content of the training scenarios is also computed with this aim. Students start with a perfect score of 100, where, if no errors are made throughout the interactions, their final score will be 100. However, each rule contained in the scenario is also defined by a severity mark, indicating the score that will be subtracted if the user fires such a rule.

The report includes information about the date and duration of the simulation, along with the number of times that the user gets stuck during the simulation and has to skip a stage. It also includes a final score and sectional results, detailing the scores obtained in each GSD skill that has been trained in the scenario one by one. The use of inappropriate or offensive language during the simulation can also take points away from the final score.

REPORT	
User: User 1	
Scenario: Project Status Meeting	**Date:** 8/2/2013 1:34:29 PM
Time consumed: 17:38 of 35:00 min	
Inappropriate language deduced 0 points	
Score: 77	
Sectional Results:	
Cultural Differences Use of direct/indirect style : 100 % Tendency to always agree : 0 %	
Linguistic Differences Overuse of certain verbs : 100 %	
GSD Problems Gain the team's confidence and trust : 100 % Misinterpretations and misunderstandings : 100 % Deal with difficult and conflictive situations : 66 % Formulate criticism/praise : 100 %	

Figure 32. Automatic assessment report

Figure 33 shows the log of a conversation, depicting how some rules are fired during a conversation, and detailing the type and severity value that will be applied by the automatic assessment. In this case the student (User1) is interacting with Amal and receiving the feedback from the Virtual Guide (María).

Instructors can therefore easily determine the areas that the students must improve, according to these results. The goal is to assign them an appropriate training module, focusing on their particular needs, or to make them repeat some of the modules in order to reinforce their learning.

The following procedure is followed in order to provide an automatic scoring during the simulations:

1. For each type of rule defined in the simulation, obtain its *MaxSeverity* value for each stage. The *MaxSeverity* value of a rule is the maximun value of its severity value in the stage.
2. Calculate the sum of all the *MaxSeverity* for each rule in the scenario according to Equation 1:

$$\text{SumTypeRule} = \sum_{k=0}^{n} (\mathbf{MaxSeverity_k})$$

Equation 1: Severity for each type of rule

   In which *n* is the number of stages.
3. *SumTypeRule* is therefore the value used to obtain the maximum score in a skill. Every time the user fires a rule, the severity of the rule will be deduced to SumTypeRule for the same type of rule.

It is thus possible to provide the student with a percentage of his/her performance in every GSD skill trained in the scenario. The system also measures the number of unsuitable or rude expressions detected during the conversation, and also when the user skips a stage.

Conversation Log:

…

Stage 9: Installer

User1: We need the deployment artifacts for Thursday

Amal: That´s perfect. We will have the installer finished by the end of Thursday, ok?

Stage 10: Schedule

User1: ok

Culture rules Fired: **Type of rule**: Tendency to always agree, **Severity**: 5

Maria: Amal is leading you to misunderstand his commitment. He has only agreed to finish the installer on time. However, you also require the documentation. Explain that you need both.

User1: We need both, installer and documentation

Amal: If you also need the documentation it would be great if we could have one more day.

Maria: Tell him that it is critical to have it on Friday morning. This must be finished because the new project will start next month and you need people available

…

Phase 25: Dealing with conflicts

Maria: Some previous problems derived from poor documentation quality were understood by your team as a lack of competence on behalf of the Indian team. From your perspective, you must let them know that they have to consider and anticipate these problems when they think the documentation is poor.

User1: Your team must let us know any problem about the documentation beforehand

GSD-related rules fired: **Type of rule**: Deal with difficult and conflictive situations, **Severity**: 5

Maria: This is a potential conflictive situation. You should moderate your language and let him know your intentions more indirectly. Use expressions such as 'It would be good if you…'.

User1: It would be good if you could anticipate problems with the documentation

Amal: I know, but it is not always easy to anticipate something that is evolving all the time.

…

Figure 33. Example of a conversation log

6.8 GSD community Web

The GSD pattern model defined in the framework has been provided with support through the implementation of a collaborative website (http://global.lero.ie/community). This implementation is based on Windows SharePoint, since it is a suitable platform with which to share, discover and discuss the patterns and scenarios that occur in GSD.

One interesting feature of SharePoint is that it provides the concept of lists. A list is a collection of pieces of information similar to a database table, with the benefit that it is easily customizable and integrates functionalities such as workflow integration, permission management and version history tracking. SharePoint includes some predefined lists including: announcement lists, blogs, contacts, discussion boards, surveys and tasks list. , SharePoint additionally permits the definition of document libraries, providing similar features for the storage and management of documents. A search engine also provides the ability to search within lists, documents and page content, and the centralized management web interface permits users, permissions and general settings of the application to be managed.

For this implementation, news, discussion boards and GSD-related resources are considered in the quest to promote the sharing of information and ideas, as depicted in Figure 34. The potential collaborators and users are instructors, researchers and practitioners with knowledge of GSD.

Figure 34. GSD community site

New participants can register in the community so that they can share their knowledge and participate in forum discussions. They can also propose improvements, share their problems or needs, and propose surveys or new features for specific purposes of interest to them and for the common good of the community. Figure 35 shows the definition of the background of a GSD project (right) and one of its associated scenarios (left) in which a Spanish analyst is interacting with a Portuguese customer.

Project Background

Project name	ORIGIM
Background	Industry
Description	
Collaboration mode	Intra-organizational
Distributed sites involved	4
GSD approach	Module Based
Size of the project (people involved)	15
Status of the project	On time
Methodology	Agile

Scenario

Project	ORIGIM
Scenario name	Requirements Elicitation
Description	
ISO/IEC 12207 processes	Acceptance
Tools used	Eclipse
Source of information	Audio recordings (meeting, telephone)
Participant 1 - Role	Analyst
Participant 1 - Culture	Spain
Participant 2 - Role	Customer
Participant 2 - Culture	Portugal

Figure 35. Definition of projects and scenarios

Figure 36 shows instances of cultural patterns which represent real conversations in the context of the aforementioned scenario. The site also includes features with which to sort and group information by taking the different attributes into account, exporting information, version control management, search engine, workflow support and automatic notifications.

CULTURAL PATTERNS

New | Actions | Settings

Type	Pattern name	Problem	Analysis of the problem	Scenario
Content Type : Communication styles (3)				
	Addressing direct communication	**Participant 1**: I **will need to** have a meeting with Edwards in order to speak about security issues. **Participant 2**: Of course, we can schedule a meeting with him when you finish the requirements document.	Participant 1 should have been a little more indirect, using a more polite form for requests, such as "Could I have..." or "May I have..."	Requirements Elicitation
	Addressing informal communication	**Participant 1**: What information do you **wanna** store? **Participant 2**: We would need everything related to customers and their associated invoices.	"**Wanna**" is too informal. Participant 1 should use "**want to**"	Requirements Elicitation

Figure 36. Pattern instances

6.9 Architecture

Having described the components that are part of the implementation of the framework, this section details the client-server architecture in which these components are articulated as shown in Figure 37. An overview of the architecture as a software product is presented in http://global.lero.ie/venture.

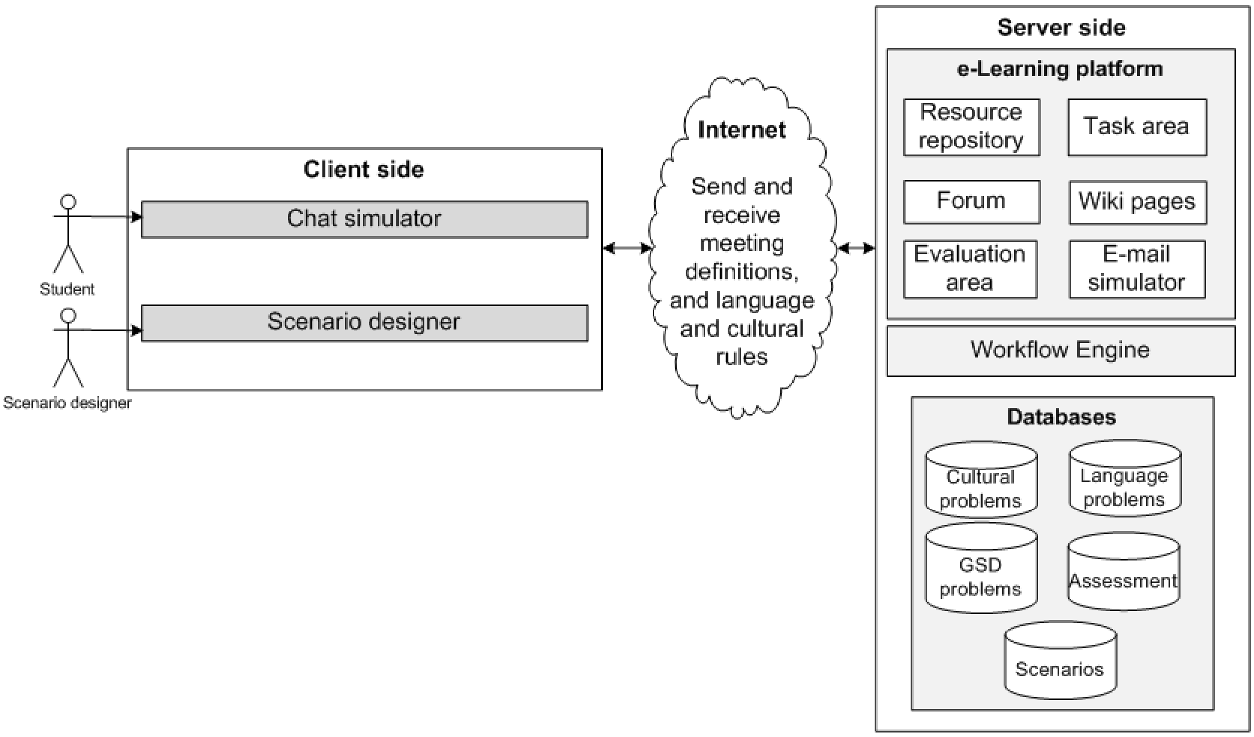


Figure 37. VENTURE architecture

The *e-learning platform* provides simulations that can be executed by the *chat simulator* (on the client side) or by the *email simulator* (on the server side). The *workflow engine* will orchestrate the execution of the scenarios contained in the *scenarios database* by sending and receiving messages to/from the aforementioned interfaces. The *assessment database* is accessed during the execution in order to track the automatic assessments. The *scenario designer* can access the *scenario database* that contains existing scenarios and also the *cultural*, *linguistic* and *GSD-related databases* in order to facilitate the creation of new simulators.

A client-server architecture was used to help comply with all the requirements of VENTURE framework. This architecture provides scalability, and therefore makes it easy to add and maintain new interactive simulators that can connect with the *workflow engine*.

6.10 Example of a training scenario

The training scenarios are designed to reflect typical problematic or controversial situations in GSD and take into account common language and cultural mistakes, where students are encouraged to find the solution to the problems by interacting properly. In this section a training scenario based on a real experience of members of a company is presented. In this training scenario, a Spanish user called Alberto plays the role of developer and has to interact with another developer from Germany called Georg. Georg has developed a webservice, and Alberto has (supposedly) developed an application that has to consume that webservice. However, Alberto has problems as regards consuming the service because it does not follow the WSDL standard. In this scenario, Alberto's task consists of explaining to Georg what the problem is and what changes are necessary in the webservice to solve the problem.

The training is intended to train the following specific skills: question formularization, negotiation skills, trust creation and linguistic problems in the context of the conversation. Before starting the simulation, the situation is explained to Alberto, so he has an idea of how he is going to interact in order to convince Georg, although the Virtual Guide will assist him at any moment.

The cultural dimensions of House [257] for Germany and Spain have been used to depict the differences in the cultural dimensions shown in Table 23 (on the 1-to-7 scale, where 1 is the lowest value of fulfillment for a dimension).

Table 23. Cultural dimensions Germany vs Spain. Summarized from [257]

House Dimensions/County	Germany	Spain	Difference
Assertiveness	4.66	4.39	0.27
Institutional Collectivism	3.97	3.87	0.1
In-Group Collectivism	4.16	5.53	**-1.37**
Future Orientation	4.41	3.52	0.89
Gender Egalitarianism	3.25	3.06	0.19
Human Orientation	3.30	3.29	0.01
Performance Orientation	4.42	4.00	0.42
Power Distance	5.48	5.53	-0.05
Uncertainty Avoidance	5.35	3.95	**1.4**

Bearing in mind that the cultural dimensions that differ most between these two cultures are in-group collectivism and uncertainty avoidance, Alberto must interact in a proper manner to cover the problems that these differences may entail:

- **In-group Collectivism** is the degree to which a community encourages and rewards the collective distribution of resources and collective action, including factors such as loyalty and cohesiveness of the individuals [257]. Members of individualistic cultures tend to be direct in their communication, expressing their inner opinions, whereas collectivist cultures tend to be more indirect. In this simulation Alberto will try to interact in an indirect manner, and the scenario will be focused on this. Figure 38 shows an example of a fragment of a conversation in which Alberto interacts in too direct a way and the Virtual Guide corrects him.

In this case, the Virtual Guide detects a direct intervention based on the detection of the patterns: “I need”, “you must”, “you have to”, etc. In the case of detecting one of these patterns in the context of this part of the conversation, the Virtual Guide will intervene to provide feedback. This rule is modeled in VENTURE as depicted in Figure 39, where the severity value is used for evaluation purposes, indicating the penalty that the user will receive if this rule is triggered.

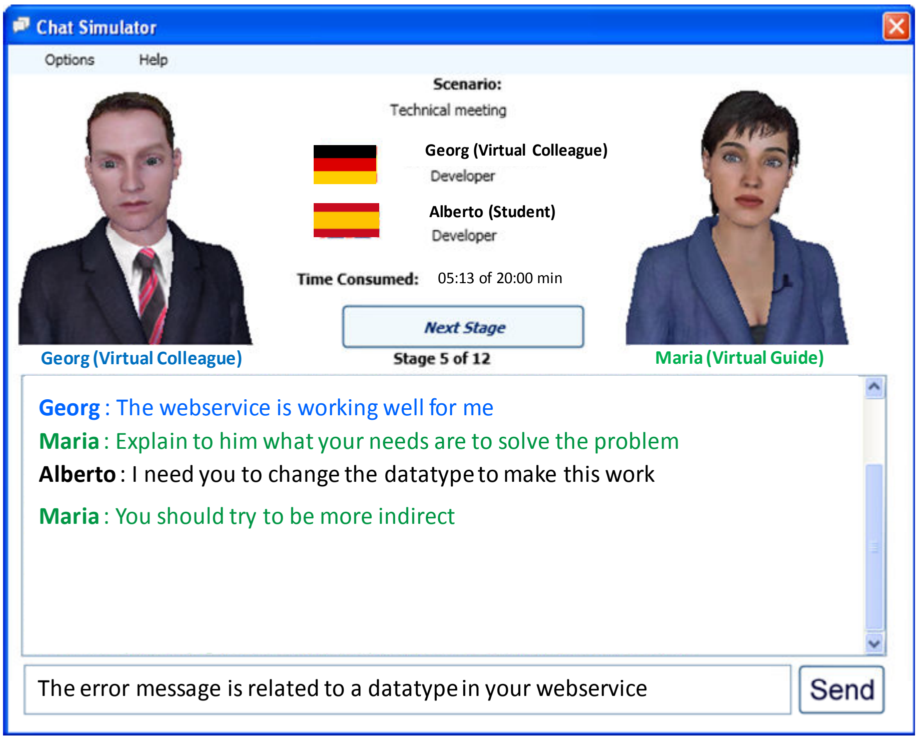


Figure 38. Simulation example

```
<CulturalRule type="direct-indirect style" severity="1">
    <pattern>"I need *"</pattern> <!—Formulate a request in a direct style”-->
    <pattern>"You must *"</pattern>
    <pattern>"You have to *"</pattern>
    <template>You should try to be more indirect</template>
</CulturalRule>
```

Figure 39. Definition of a cultural rule

- **Uncertainty Avoidance** is the degree to which individuals feel “comfortable” in new situations. Individuals tend to avoid uncertainty by relying on social norms, customs, and bureaucratic practices [257]. Individuals with high levels of uncertainty avoidance tend to seek more feedback than those who are more tolerant to uncertainty, either by asking questions or observing. As Germans are less tolerant to ambiguity, Georg could feel anxiety and stress if Alberto is not clear with his proposal for a solution. Short-term feedback is a proactive method that Alberto should apply to avoid these feelings. Alberto should also try to minimize the uncertainty by attempting to provide as much information as possible and avoiding, for example, misunderstandings or improper use of language. In the following fragment of a conversation, Alberto makes a mistake that could cause a misunderstanding (see Figure 40).

Alberto: Could you realize some changes in the webservice?
María: "Realize" is a false-friend in Spanish. Do you mean to "carry out"?
Alberto: Could you carry out some changes in the webservice?
María: You should try to be more polite using "please"
Alberto: Please, could you make a change in the webservice?

Figure 40. Training uncertainty avoidance

In this case, a language rule has been triggered in order to correct a false-friend mistake. Moreover, in the same conversation, the Virtual Guide has also detected that the formulation of a question has been too direct without using the word "please". These rules are modeled as shown in Figure 41:

```
<LanguageRule type="false friend" severity="2">
    <pattern>realize</pattern> <!--Incorrect use of the word "realize"-->
    <template>"Realize" is a false-friend in Spanish. Do you mean to "carry out"?</template>
 </LanguageRule>
<CulturalRule type="direct-indirect style" severity="3">
    <pattern>"!please + ?"</pattern> <!—Formulate a question without "please"-->
    <template>You should try to be more polite using "please"</template>
 </CulturalRule>
```

Figure 41. Rules definition

The Virtual Colleague (Georg) may also make mistakes during his interaction, signifying that Alberto must receive training in what he may confront in this respect. In the example shown in Figure 42, the Virtual Guide warns him about a mistake.

Georg: You must not work with this webservice until we finish our module
María: A typical German false friend consists of using "must not" when they really mean "not to have to". You still should work on it, as the deadline is too close.

Figure 42. Training false-friends

In this case, Georg uses "must not" incorrectly, as it is a false-friend, similar to the German "*muss nicht*", which means "do not have to", instead of "must not". So Alberto must know how to manage the uncertainty that this kind of situations generates. The training scenario can be designed to detect the pattern "must not" or "mustn't" in this specific context of the conversation and explain the problem to the student. Moreover, the text introduced by the user is automatically checked by a spelling dictionary which will provide him with feedback in the case of making grammatical mistakes.

6.11 Designing training scenarios

In the context of this thesis, three training scenarios were fully implemented by using the scenario designer. This experience has led to the following observations regarding the design of training scenarios.

Defining a training scenario can be a tedious task, as it requires thinking of a real situation in GSD in which different GSD problems appear, including cultural and linguistic differences. Moreover concepts regarding GSD should be explained by using additional training material, and questionnaires or other kinds of assessment artifacts should also be prepared. In order to formalize the process of developing training scenarios the following steps are defined:

1. Define the cultures involved in the virtual meeting.
2. Define a topic for the conversation, the settings for the project, and the sites involved in a GSD context.
3. Investigate cultural and linguistic problems that appear during interactions between the cultures and languages involved. Some help may be found in literature. For example the following references have been used for the definition of the training scenarios:
    - Craig [347]: presents seventy four realistic dialogues which represent common problems or misunderstandings that may occur when Americans interact with people from several regions.
    - Eden-Jones [238]: On practices for working effectively with culturally diverse clients and co-workers. This includes topics such as accepting cultural diversity, communicating effectively with culturally diverse people and resolving cross-cultural misunderstandings.
    - Cheng [348]: On intercultural conversations.
    - Warren [349]: On international communication.
    - Hosftede [256]: Detailing differences among the different cultures.
    - Specific bibliography regarding common errors of interacting in English committed by non-native speakers of other languages. For example García and Otheguy [350], Axtell [351] or Grzega [352].
4. Investigate GSD problems that could occur in similar settings and contexts.
5. Define the learning outcomes of the scenario
6. Write a script in which the problems investigated may appear, and the learning outcomes trained.
7. Formalize the script in order to separate the different stages and define cultural, linguistic and GSD-related rules according to the template in Table 24. The Virtual Agents' and the students' interactions should be detailed for each stage. In the case of the students, different kinds of entries and wildcards may be considered at this stage.
8. Create the scenario in the designer by following the aforementioned definition.
9. Test the training scenario with different people and make improvements according to the results. Typical problems with immature scenarios are:
    - The student does not know how to interact at a certain stage. In these cases, some help from the Virtual Guide may be sufficient.
    - The student introduces a valid statement that was not considered in the definition of the scenario and consequently the Virtual Colleague does not understand him. In order to solve this, a new entry should be included in the conversational knowledge section for that stage.
    - The student interacts in an incorrect manner but the conversation follows the flow without providing any feedback. In this case a new entry should be included in the corresponding section (cultural rules, linguistic rules or GSD-related rules) in order to provide the user with feedback.

Table 24. Example of the definition of a stage

Stage 25	**Problems with the Indian team**
Conversation	**Virtual Colleague**: These previous problems were understood by your team as a lack of competence on behalf of the Indian team. From your perspective, you must let them know that they have to consider and anticipate all these problems in order to avoid these situations in the future **User**: It would be good if you could somehow anticipate this kind of problems **Amal**: I know, but it is not always easy to anticipate something that is evolving all the time
Cultural problems	None
Linguistic problems	None
GSD problems	**Type of problem:** formulate criticism **User**: because of your delays/competence **Virtual Colleague**: Try to trust him and have a more relaxed conversation
	Type of problem: deal with difficult and conflictive situations **User**: should, must, have to **Virtual Colleague**: This is a potential conflictive situation. You should moderate your language and let him know your intentions more indirectly. Use expressions such as 'It would be good if you could…'

6.12 Conclusions

This chapter presents the set of software tools that provide full support for the VENTURE framework. The implementation achieved provides an e-learning platform, a chat simulator and a scenario designer connected to a knowledgebase of common problems. A workflow engine is also provided as the core for executing the simulations, providing automatic assessment mechanisms, and a GSD community Web permits the acquisition of GSD-related knowledge that can be used with simulation purposes.

The architecture presented provides flexibility, as the scenarios and functionality can evolve transparently with the end-users. It also provides scalability, since the server could be replicated or balanced in order to adapt it to different workloads. Another advantage of the architecture is its easy maintenance and interoperability, thus making it feasible to implement new simulators that could use the same workflow engine in the future, and serving to train students in new collaborative and communicative tools.

As is detailed in the next chapter, this software has gone through various stages of testing and evaluation with students, researchers and practitioners, which has served to improve its technical and functional aspects. This process has also been helpful as regards improving the effectiveness of the tool for GSD training activities, thus complying with the framework's objectives.

Chapter 7

Field Study

According to Sargent [353], **validation** is the substantiation that the components within the model's domain of application possess a satisfactory range of accuracy consistent with the intended application of the model. Carson and Robinson's definitions are pertinent to this study, in which validation is defined as "the process of ensuring that the model is sufficiently accurate for the purpose at hand" [354] or, according to Robinson [355], whether the right model is being built. **Verification** is defined by Davis [356] as the process of ensuring that the model design has been transformed… with sufficient accuracy, testing whether the model is built correctly [355], and ensuring that the model components are correct [353]. **Evaluation**, however, encompasses both validation and verification activities along with the model's quality, usability and utility assessment [357].

The Field Study presented in this chapter is an *evaluation* of the VENTURE framework that aims to test whether students, when interacting through the use of the VENTURE tool, improve their knowledge as regards cultural, linguistic and GSD-related problems. This chapter therefore evaluates VENTURE learning objectives through a validation exercise that tests the framework's learning outcomes. Questionnaires are simultaneously posed to *verify* that the chat simulator component is a good solution for GSD training. At the end of the chapter, quality and usability factors are also measured in order to complete the evaluation.

The participants in this Field Study are potential users of VENTURE tool, and originate from a population of students, researchers and practitioners. Data is collected via questionnaires, automatic assessments and logs of simulated interactions. An analysis of the quantitative and qualitative data derived is described in this chapter.

7.1 Background

There is no consensus in literature as to standardized evidence-based methods with which to validate virtual training platforms, and there is no methodology that is unique to any particular learning environment [358]. The method chosen must meet the specific objectives of the environment, which in this case is oriented towards the development of GSD and soft skills.

According to Mikropoulos and Natsis [359], most of the experimental studies that report on educational learning environments apply within-subject approaches, in which a pre and post evaluation of the learning approaches of the same group of students is conducted [360]. In a within-subjects design, each participant is tested under each condition. For example, there could be two conditions or scenarios in which the participants conduct both tests. The alternative to a within-subjects design is a between-subjects design. Between-subjects designs test each participant under one condition only, and there are two groups of participants; one group of participants is tested under the first condition or scenario and a separate group is tested under the second one.

One of the greatest advantages of a within-subjects design is that it does not require a large pool of participants as compared to the between-subjects design. A within-subjects design can also help reduce the errors associated with individual differences. This method is usually based on interviews or questionnaires whose objective is to measure students' perceptions of usability and usefulness [361].

When a training tool is employed, the evaluation can consist of measuring the students' performance by applying and comparing different configurations of the tool. For example, a within-subjects approach was followed by Vizcaíno [362] to evaluate a collaborative learning environment based on a simulated student agent. In this case, two different sessions using a different configuration of the environment served to compare the effects of applying the simulated agent.

Finally, Navarro and Hoek [223] presented a simulation-based tool with which to train Software Engineering processes. They evaluated the usage of the environment in both software engineering courses and a formal out-of-class experiment. In both cases, the participants/students were given an assignment to enact a certain number of simulations, after which they answered a set of questions concerning the concepts taught and their experience with the tool. A comparative experiment was also conducted to compare the effectiveness of the different methods that could be applied when teaching a specific set of software process concepts.

7.2 Evaluation method

In the case of this thesis, a **within-subjects design** has been applied, in which the comparisons are made between two or more results obtained from different circumstances involving the same participant. The within-subject design was chosen because it requires fewer participants than the between-subjects design, and as our experiment involved practitioners it was difficult to gather two representative groups with the same characteristics [363]. This method also reduces the error variance associated with individual differences that may occur in a between-subjects design.

The experiment was conducted by individual participants in two sessions of up to an hour each. These sessions tested the same configuration of the tool with different training scenarios. In VENTURE, both training scenarios provided training on similar skills, so that it was possible to compare the learning effects. A log of the interactions, automatic assessments and questionnaires was used to measure the two groups of participants' improvements, as detailed in this section. The following points were considered by the evaluation method:

Intervention: VENTURE training platform.

Objective: To test whether VENTURE enhances a user's ability to communicate with people from a different culture with a different first language.

Hypothesis: VENTURE can enhance users' knowledge as regards:

- how to communicate with people from a different culture.
- how to communicate with people with a different native language.
- GSD challenges and concepts, and the skills needed to tackle GSD problems.

Research questions: The following research questions are covered:

Q1. Can VENTURE enhance a user's knowledge and skills as regards how to communicate with people from different cultures?
Q2. Can VENTURE enhance a user's knowledge and skills as regards how to communicate with people with different native languages?
Q3. Can VENTURE enhance a user's knowledge as regards GSD challenges and concepts, and the skills needed to tackle GSD problems?

Outcomes of relevance: The specific relevant outcomes of this Field Study are organized according to the research questions, as presented in Table 25.

Table 25. Outcomes of relevance

Research questions	Outcomes of relevance
Q1	Participants' knowledge of cultural issues Participants' knowledge of solutions to solve cultural problems Participants' level of confidence and skills when interacting with people from a different culture
Q2	Participants' knowledge of linguistic challenges Participants' knowledge of solutions to solve idiomatic problems Participants' improvement as regards skills needed to tackle linguistic differences
Q3	Participants knowledge of GSD communication problems

Data collection methods: It was necessary to obtain the following data:

- *Participant´s opinion*. Based on the analysis of the survey results concerning their learning and experience with VENTURE.
- *Participant's performance by using VENTURE*. Using automatically generated logs and reports.
- *Participant´s level of learning*. Comparing pre-questionnaires and post-questionnaires.

This information was acquired by using the mixed-methods approach [364] and considering the following data collection methods:

- Automatic assessment generated and stored by VENTURE.
- Logs of the simulated interactions.
- Questionnaires (presented in the following Section). In order to check that the participants had understood the scenario and improved their knowledge. The surveys were administered online through the e-learning platform.

Sample: The participants in this study were students, researchers and practitioners, who are potential users of the framework. As the training provided in the simulations was related to the Mexican and the Indian cultures, participants from these two cultures were excluded from the sample.

7.2.1. Questionnaires

The objective of this study was to prove the learning effectiveness of the platform. Quality and user experience factors were also measured by following the PSSUQ (Post-study System Usability Questionnaire), originated by IBM. The description provided by Sauro and Lewis [365] was followed (Appendix VIII – Usefulness measure). User satisfaction and stakeholder satisfaction measures were also aligned according to ISO/IEC 25022 [366], including usefulness, trust, pleasure and comfort measures (Appendix VIII – Trust, pleasure and comfort measures).

The following questionnaires were applied:

- *Baseline Questionnaire:* designed to collect the participant's demographic information and background (see Appendix V).
- *Pre Training Questionnaire*: including questions on the specific knowledge of what is taught on the course (i.e., their current knowledge of GSD issues; confidence levels in communicating with people from other cultures in the job) (see Appendix VI).
- *Post Training Questionnaire*: whose objective was to check whether there was an improvement in the participant's knowledge. This included the same questions as the Pre Training questionnaire, but also included questions about the participants' perception of their improvement (see Appendix VII).

- *Opinion Questionnaire*: this was conducted at the end of the experimental sessions and obtained information on how the student felt about the experience of using the simulated training environment, including the usability of the system (see Appendix VIII).

7.2.2. Training scenarios

The following literature has been used in order to develop the training scenarios: [347], [256]. Moreover, experts on cultural diversity in GSD and practitioners involved in GSD were consulted in order to extract potential conversations and problems when designing these scenarios.

The following training scenarios were designed and applied during this study:

- **Scenario A**: In this scenario, the user plays the role of an onsite coordinator of an offshored team working for a Mexican client. The Mexican team has scheduled a set of tasks consisting of a set of test cases that were scheduled for three to four weeks. Their idea was that user´s team should finish the majority of them during the first two weeks and leave the last two weeks as a buffer for issue resolution.
  However, the Mexican coordinator does not communicate this point clearly, leading the user to misunderstand the deadline and the serious implications that it entails. Not meeting the deadline could result in the Mexican team losing faith in the user´s team capabilities. It is therefore important for the user to anticipate any problem that could cause a delay.
  During the conversation, the user must also discuss a technical problem that a member of his/her team is experiencing with a web service developed by the Mexican team. The user has to address this issue in a polite manner, and attempt not to be too assertive when raising issues that relate to what s/he thinks is the cause of the problem. An example of a conversation in this scenario is shown in Appendix X.
  Learning outcomes: Having interacted with VENTURE, the user should learn the following:
    - How some cultures have a tendency to always say "yes".
    - Different communication styles
    - The "Mum" Effect
    - Knowledge of Mexican customs
    - Typical problems a Mexican experiences when speaking English
    - Negotiation
    - How to avoid misunderstandings

- **Scenario B**: in which the same number and kinds of rules and GSD patterns applied in Session A are trained by using a different scenario (in a different context).

In this scenario, the user plays the role of an onsite coordinator for a company that has contracted the services of an Indian outsourced team. The project manager on the user's site has scheduled a set of tasks consisting of a set of test cases whose schedule must be discussed.

Not meeting the deadline could lead to serious problems for the user´s company. It is therefore important for the user to motivate the Indian team and anticipate any problems that could cause a delay.

During the conversation, the user must also discuss a technical problem that one of the team members is experiencing with a web service developed by the Indian team. The user has to address this issue in a polite manner, and attempt not to be too assertive when speaking about the reasons for the problem. An example of a conversation in this scenario is shown in Appendix XI.

Learning outcomes: Having interacted with VENTURE, the user should learn the following:
- Different communication styles
- The meaning of unproductive productivity
- Knowledge of Indian customs

- Typical problems Hindi speakers experience when communicating in English
- How to deal with conflicts
- Negotiation
- How to raise counter-arguments
- How to avoid misunderstandings
- How to avoid the escalation of errors

7.3 Procedure

The participants were introduced to the evaluation by being provided with a written explanation of VENTURE along with the objectives of the course. The participants were split into two groups: G1 and G2. Both groups conducted the same tasks but in a different order (Evaluation G1, and Evaluation G2). This order is explained below, and is represented in Figure 43.

Both groups initially filled in a *Baseline questionnaire* as the first task in the study. G1 then participated in Evaluation G1. Evaluation G1 consisted of filling in a *Pre Training Questionnaire* for *Scenario A*, after which *Scenario A* was executed, and finally a *Post Training Questionnaire* for *Scenario A* was filled in. After this, G1 participated in *Session B*, which consisted of filling in a *Pre Training Questionnaire* related to *Scenario B*, after which *Scenario B* was executed, and finally a *Post Training Questionnaire* on *Scenario B* was filled in.

The order of the sessions was different for G2, who participated in Evaluation G2, signifying that they first executed Scenario B and then Scenario A. After completing both sessions, both groups of participants (G1 and G2) completed the *Opinion Questionnaire*.

The reason for changing the order in which the two groups (G1 and G2) participated in the scenarios was to test whether there was any bias in the difficulty ratings of the two scenarios. For example, if all the participants who participated in Scenario A followed by Scenario B showed improvements, this might have been because Scenario A was more difficult than Scenario B; however, if Scenario A was always more difficult for the participant (independently of the order), it would not have been possible to *compare* the two scenarios for any proof of learning.

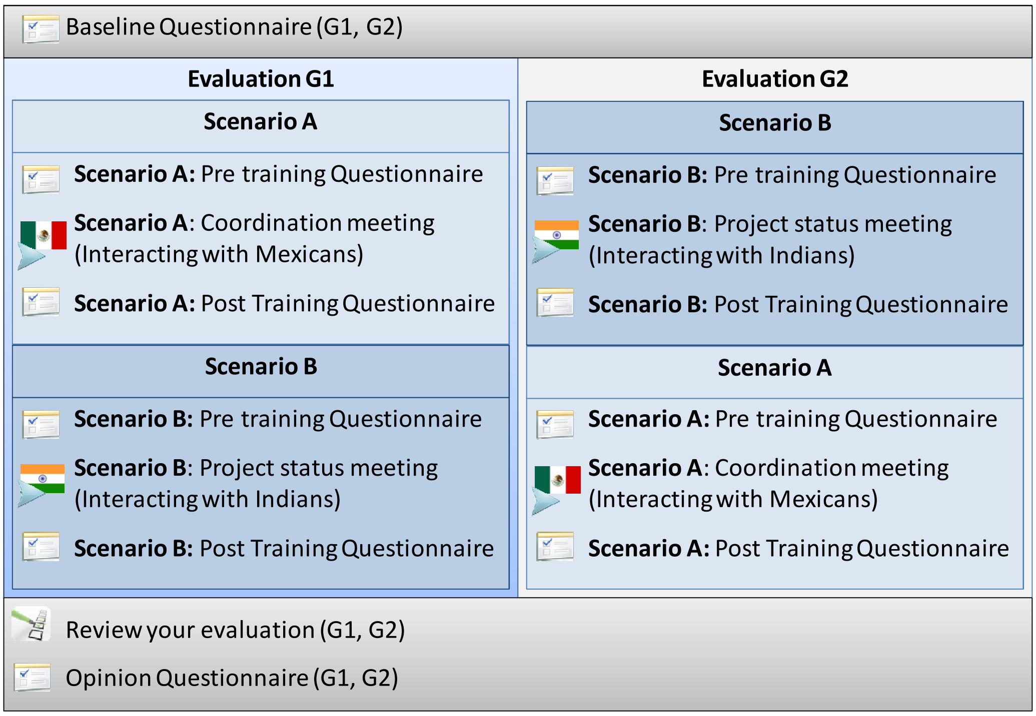
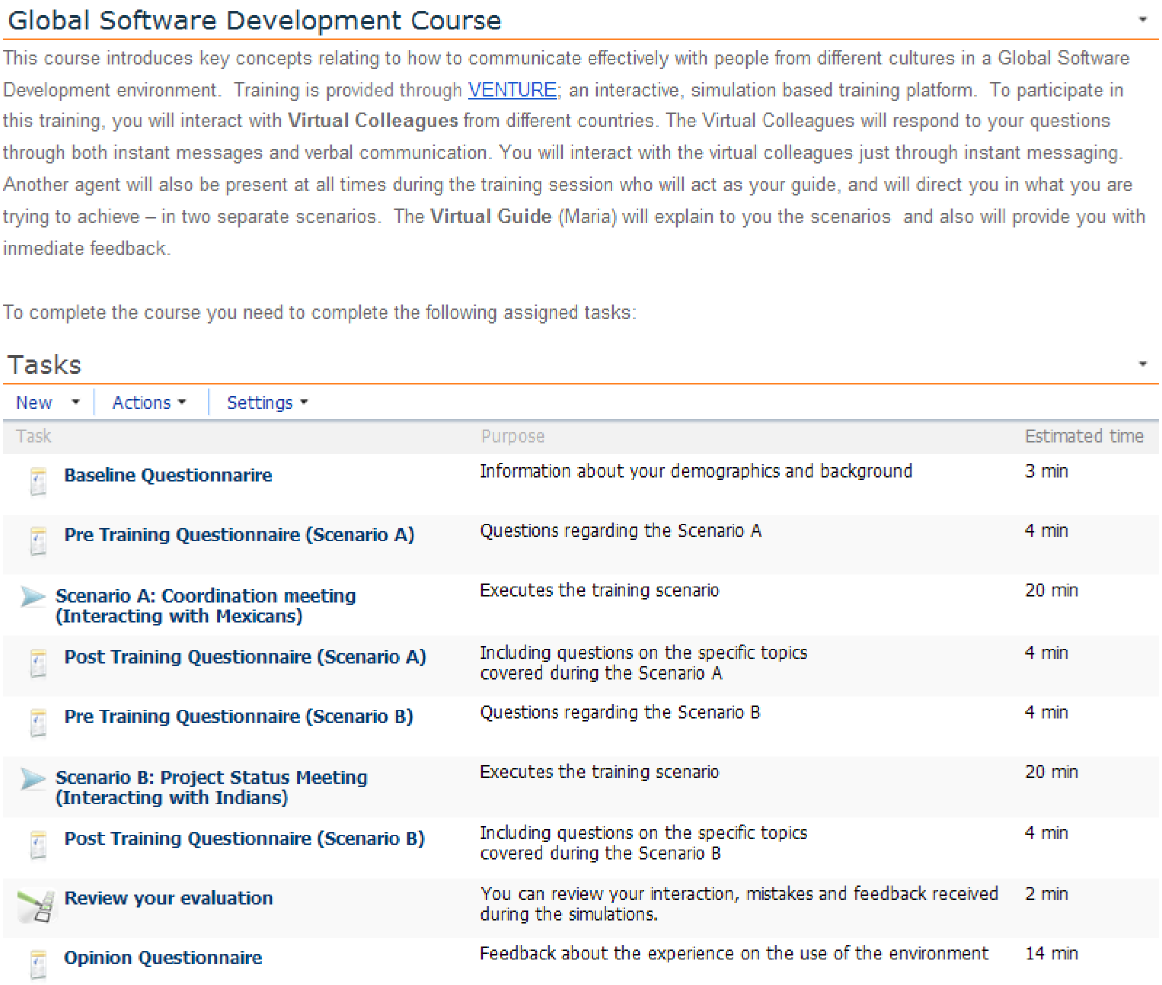


Figure 43. Flow of tasks for the participants

Figure 44 shows the screen that the participants accessed online, detailing the tasks that they had to complete with their average estimated time.



Global Software Development Course

This course introduces key concepts relating to how to communicate effectively with people from different cultures in a Global Software Development environment. Training is provided through VENTURE; an interactive, simulation based training platform. To participate in this training, you will interact with **Virtual Colleagues** from different countries. The Virtual Colleagues will respond to your questions through both instant messages and verbal communication. You will interact with the virtual colleagues just through instant messaging. Another agent will also be present at all times during the training session who will act as your guide, and will direct you in what you are trying to achieve – in two separate scenarios. The **Virtual Guide** (Maria) will explain to you the scenarios and also will provide you with inmediate feedback.

To complete the course you need to complete the following assigned tasks:

Tasks

New Actions Settings

Task	Purpose	Estimated time
Baseline Questionnarire	Information about your demographics and background	3 min
Pre Training Questionnaire (Scenario A)	Questions regarding the Scenario A	4 min
Scenario A: Coordination meeting (Interacting with Mexicans)	Executes the training scenario	20 min
Post Training Questionnaire (Scenario A)	Including questions on the specific topics covered during the Scenario A	4 min
Pre Training Questionnaire (Scenario B)	Questions regarding the Scenario B	4 min
Scenario B: Project Status Meeting (Interacting with Indians)	Executes the training scenario	20 min
Post Training Questionnaire (Scenario B)	Including questions on the specific topics covered during the Scenario B	4 min
Review your evaluation	You can review your interaction, mistakes and feedback received during the simulations.	2 min
Opinion Questionnaire	Feedback about the experience on the use of the environment	14 min

Figure 44. Tasks assigned to participants

During the execution of the scenarios, an automated assessment is generated, revealing the cultural, linguistic and GSD errors committed by the participant (see Figure 45). This report is also used for analysis purposes.

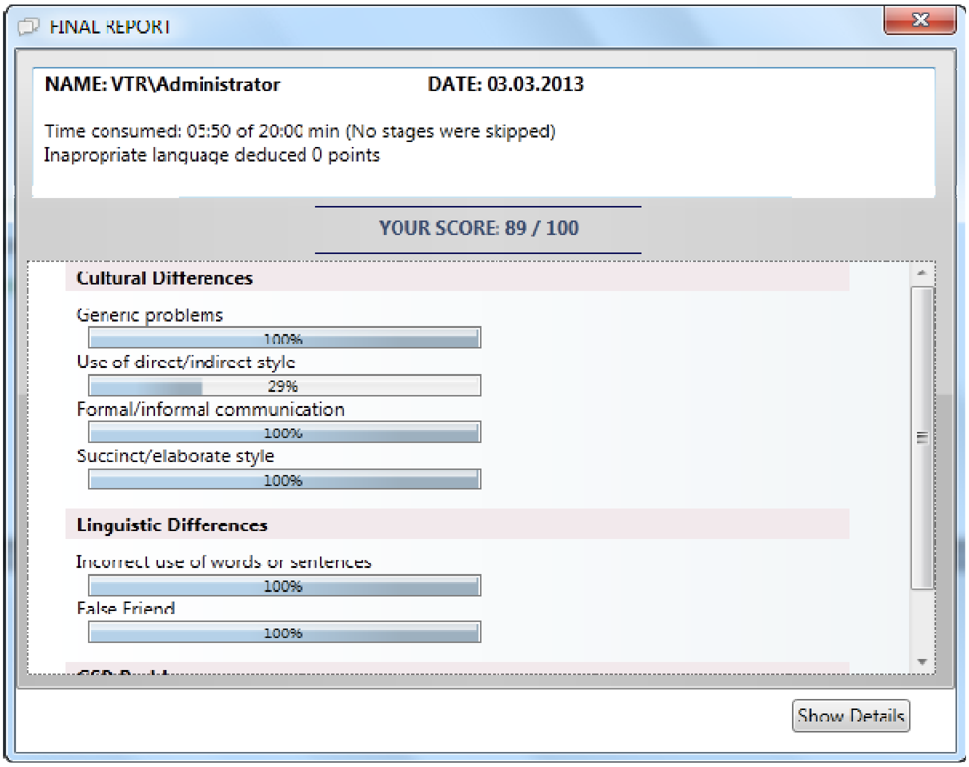


Figure 45. Automatic assessment report

7.3.1. Environment

The study was conducted by two interviewers from two different locations:

- Lero – The Irish Software Engineering Research Centre, University of Limerick, Limerick, Ireland.
- Alarcos Research Group, University of Castilla-La Mancha, Ciudad Real, Spain.

The participants at these locations had access to a computer that had been specially prepared to conduct all the steps required. The Chat Simulator was installed in this computer and access to the e-learning site (http://global.lero.ie/EduGlobal/) was granted to all the participants. Those participants who did not have access to these locations conducted the process online by remotely accessing one of the computers at either Lero or Alarcos, and their actions were monitored by means of TeamViewer (www.teamviewer.com).

7.4 Pilot Study

Prior to conducting the Empirical Study, a Pilot Study was conducted. The main objective was to check that it was feasible to answer the research questions by means of the evaluation method proposed. The second objective was to improve the quality of the process, making it more effective and reducing the amount of work the participants would have to do. This objective involved concentrating on the following points [367]:

- Check question sequencing and flow
- Evaluate the overall layout and design of the scenarios
- Refine response categories
- Check adequacy of written instructions

Five participants, whose characteristics are shown in Table 26, conducted the Pilot Study during the period 10/07/2012 to 12/07/2012. The participants were first introduced to the training framework and then received instructions on how to conduct the study, which they executed later without any external help. They took part in an initial version of the evaluation method described in this chapter, which was iteratively improved through this process. In order to obtain the feedback from the participants, a questionnaire was answered at the end of each session (see Appendix IV). After analyzing the feedback provided by each participant, immediate improvements were made to the process, questionnaires, and training scenarios, which were tested by the following participants.

Table 26. Participants in the Pilot Study

Participant	Role	Nationality	Age
1	Expert on Teaching	Irish	36
2	Researcher	Mexican	49
3	Researcher	Mexican	49
4	Researcher (Expert on GSD)	German	34
5	Researcher	Spanish	24

7.4.1. Results of the Pilot Study

The following subsections detail the main improvements that were made to the questionnaires, scenarios and the evaluation process as a result of this Pilot Study.

Questionnaires

During the Pilot Study, the need to include some negative questions in the questionnaires also arose. It was detected that most of the questions in the initial version of the questionnaires were formulated in a positive manner, which could lead to some unreliable result biases. For example, the length of the evaluation process and the number of questions can introduce 'fatigue effects' [368], which can lead to inaccurate answers, and in the case of Likert scales, respondents have a tendency to always tick the same choices.

After conducting the study, one of the participants provided a negative perception as he confessed that he did not like e-learning systems. Instead, he preferred traditional classes rather than learning by interacting with a machine. In this respect, he admitted that his responses might be biased by his particular way of thinking. In order to consider this kind of mindset, it was decided to include the following question in the *Baseline Questionnaire*: "BQ16. I do NOT like using e-learning systems", and also the following question in the *Opinion Questionnaire*: "QQ1. Using this tool has changed my perception of the usefulness of e-learning systems".

In these cases, the participants, rather than attempting to provide the most accurate answers, also tend to settle for merely satisfactory answers [369]. In order to avoid participant acquiescence, the questionnaires were reviewed so as to include negative questions [298]. In some cases it was also decided to formulate a particular question in a positive way and later formulate the same question in a negative way. This serves to check that the participants are coherent in their answers. An example of this technique, which can be found in the *Opinion Questionnaire*, is shown below:

OQ1. I have increased my knowledge of the Indian culture.
OQ2. I have increased my knowledge of the Mexican culture.
OQ5. I have NOT increased my cultural knowledge.

Scenarios

The same participant who did not like e-learning systems, also suggested that he did not read the textual explanations provided in the platform. Someone who uses an e-learning platform is normally expected to have a high degree of interaction rather than reading too much. In order to solve this point, those texts explaining the objective of the training scenarios were summarized and better formatted in order to make the information easier to digest.

Since all training was timed, it was also detected that those participants with a low level of English took longer to conduct the training scenarios than the fluent English speakers. In one case, one of these participants was asked to repeat one of the scenarios as he could not finish it because of the time limitation (30 mins for both scenarios), and a considerable improvement in his performance was observed the second time, in addition to an improvement in the final score obtained during the automatic assessment. This led us to believe that learning had occurred; however, this claim will be accurately demonstrated in this chapter. The time needed to complete the scenarios was increased by 5 minutes in order to give people with a lower level of English a better opportunity to finish the scenario.

Minor improvements were implemented in the definition of the training scenarios after observing how the participants interacted. The feedback provided by the Virtual Guide was also improved in those cases in which it was detected that some participants would need more help.

Evaluation process

After completing the Pilot Study, one participant indicated that s/he had felt pressurized, because this process was like an exam for him/her. After considering this comment, the participants were told that their function was to evaluate the system and that they should not feel the pressure of being examined, since they were in fact examining the framework. This explanation was, as a result of the pilot study, included in the first introductory steps of the process for the real study.

It was also observed that the time required to conduct the study was too long for a single session, and some of the participants might feel too tired. This led us to define two separate sessions for the executions of Scenarios A and B. The option of shortening the training scenarios was discarded, as it was necessary to include sufficient training material to prove that any learning had occurred.

Finally, the automatic assessments and logs of the conversations were reviewed in order to check that the information was stored in a consistent manner to be able to conduct the final process.

7.5 Settings

This section provides insights into the specific settings in which the evaluation was conducted.

7.5.1 Exclusion criteria selection

Before analyzing the results it was necessary to filter those evaluations that were not concise or were not useful for conducting the subsequent analysis. The questionnaires therefore considered some questions designed to detect inconsistencies in the answers. If the participant answered in a different sense to any of these questions, his/her responses were discarded from the sample used in the analysis. The following questions in the Opinion Questionnaire served to achieve this objective:

OQ1. I have increased my knowledge of the Indian culture.
OQ2. I have increased my knowledge of the Mexican culture.
OQ5. I have NOT increased my cultural knowledge.

OQ24. The e-learning platform is easy to navigate and use.
OQ28. I became confused when using the e-Learning platform.

OQ56. I feel comfortable using this system
OQ59. I feel uncomfortable when using this system

By applying this method two participants were excluded from the final sample. One of the participants strongly agreed that he had gained knowledge about the Indian and Mexican culture; however, he also strongly agreed that he had not increased his cultural knowledge. Moreover, in one of the simulations, the same participant only completed 80% of the scenario, which was not considered sufficient to be relevant for its analysis. Another participant was removed from the final sample as he scored the same amount in questions OQ56 and OQ59.

7.5.2 Sample

VENTURE framework is oriented toward students and inexpert practitioners who need specific training in GSD. This section describes the sample that has been chosen in order to be representative of the intended population. The sample comprised thirty-four participants who completed both scenarios, and answered all the questions in the surveys consistently. These participants were selected from different cultures including Brazil, Bulgaria, China, Finland, Iran, Ireland, Italy, Pakistan, Spain and the United States. Figure 46 details the number of participants from each country, showing a high representation of Spanish and Irish participants. Ten different countries are represented in the sample.

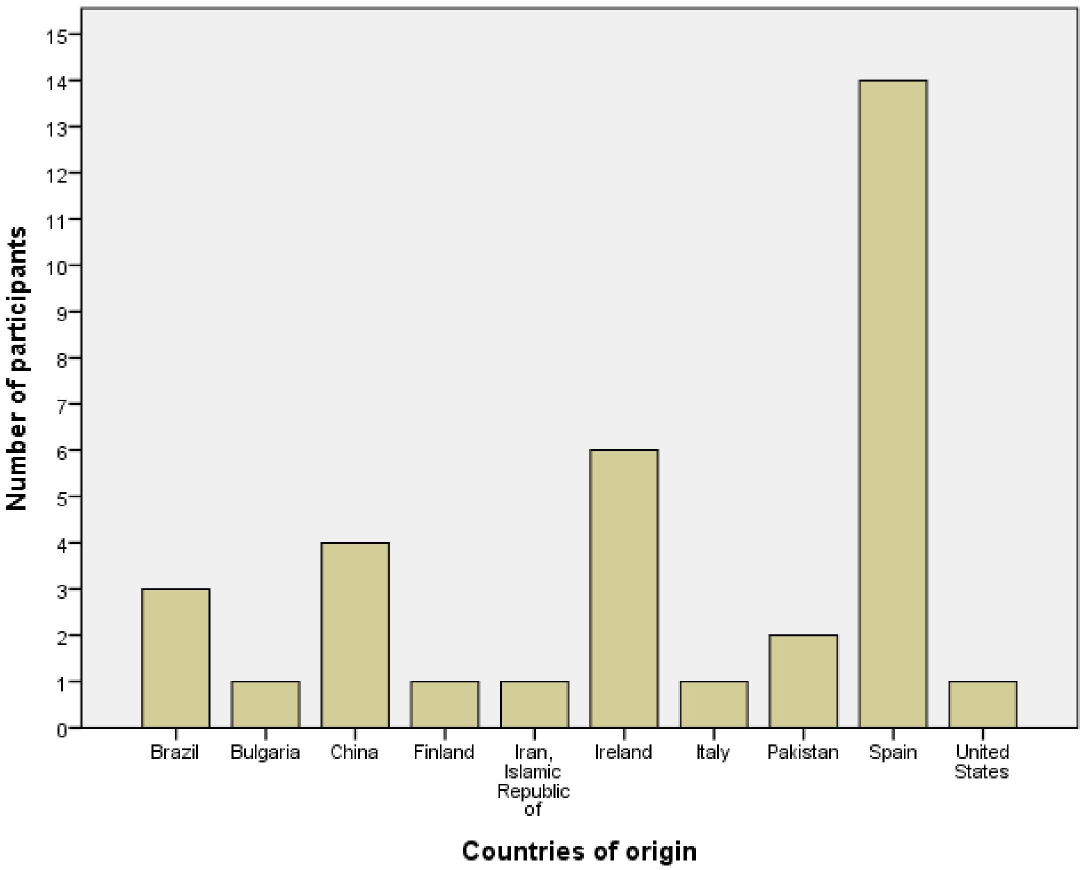


Figure 46. Distribution of participants by countries of origin

Knowing the culture and native language of the participants is important for this research because they have to interact with Virtual Agents from different cultures in English. Figure 47 reports the native language of these participants. For all those whose native language is not English, their second language is in fact English. The level is very diverse, although the participants measured their level according to their own perception and without following any standard.

As one of the scenarios considers the Mexican culture, whose main language is Spanish and the other scenario involves a Hindi native speaker from India, this may have some influence on the results as the number of Spanish speakers in the sample is high. However, this limitation is only related to linguistic learning. In order to avoid any bias as regards the cultural learning, Mexicans and Indians were excluded from the sample.

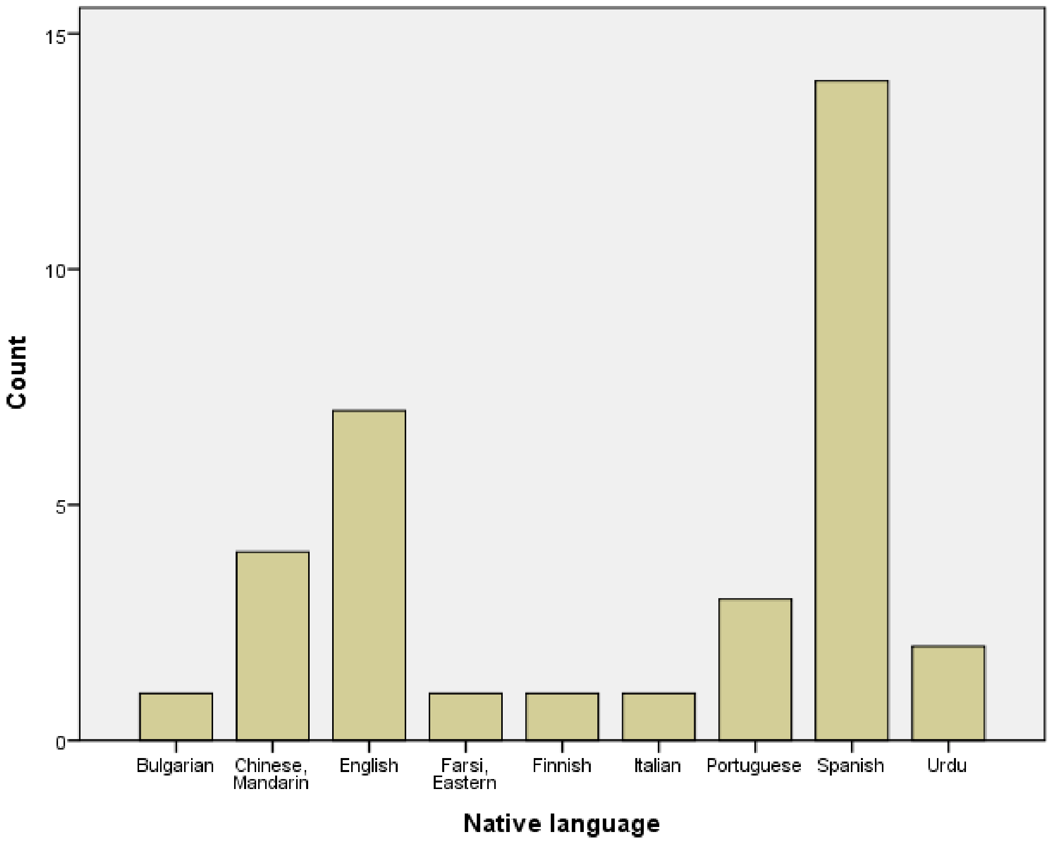


Figure 47. Native language of the participants

Figure 48 depicts that 29.4% of the sample were female compared to 70.6 % of male participants, meaning that females are under-represented in the sample population, although this may be representative of real GSD populations in which males tend to be predominant.

Participants: A stratified sample of participants was selected in order to include:

- Computer Science students.
- Practitioners with experience in Software Development or distributed work environments in IT.
- Researchers from a Software Engineering background with different levels of knowledge as regards GSD. This group also considers lecturers from a Software Engineering background with different levels of GSD knowledge.

This sample reflects the type of user the tool is targeting (students and practitioners). Researchers were also considered, as their perspective is also valuable when validating a research based tool, and lecturers because of their experience in education environments.

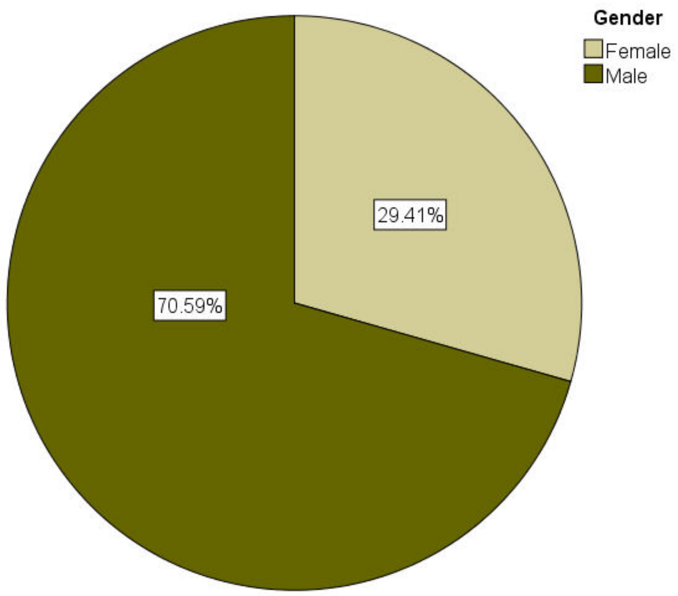


Figure 48. Distribution of participants by gender

Figure 49 depicts the percentage of participants by concentrating on their roles. Some of the researchers are lecturers, although this distinction is not expressed in this diagram. It must be noted that some of the researchers had previous experience in companies and some practitioners had experience in research, which is why the "Practitioner/Researcher" category has been included. The majority of the sample (16 participants) is composed of students, since they are the main target of the VENTURE framework. Researchers are the second group in number with 12 participants, the rest of the sample is made up of practitioners and researchers with experience in companies (or current practitioners with research experience).

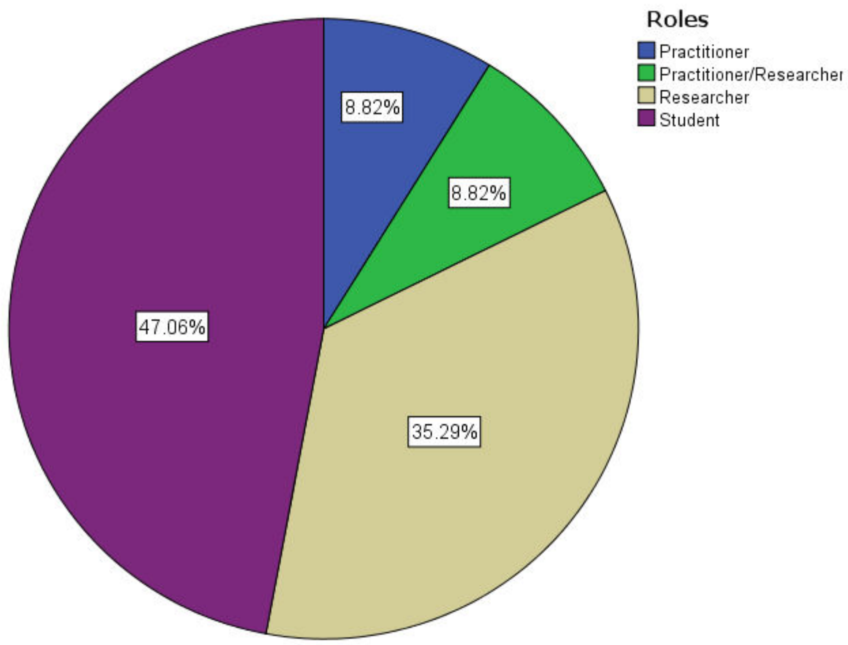


Figure 49. Distribution of participants by roles

With regard to the participants' age, and bearing in mind that students are the most important group, the mean age is around 30 years old. Table 27 shows details of the distribution of ages in the sample, whereas Figure 50 represents this distribution in a boxplot.

Table 27. Age of the participants

	Statistic
Median	31.0000
Variance	77.645
Minimum	18.00
Maximum	49.00
Range	31.00

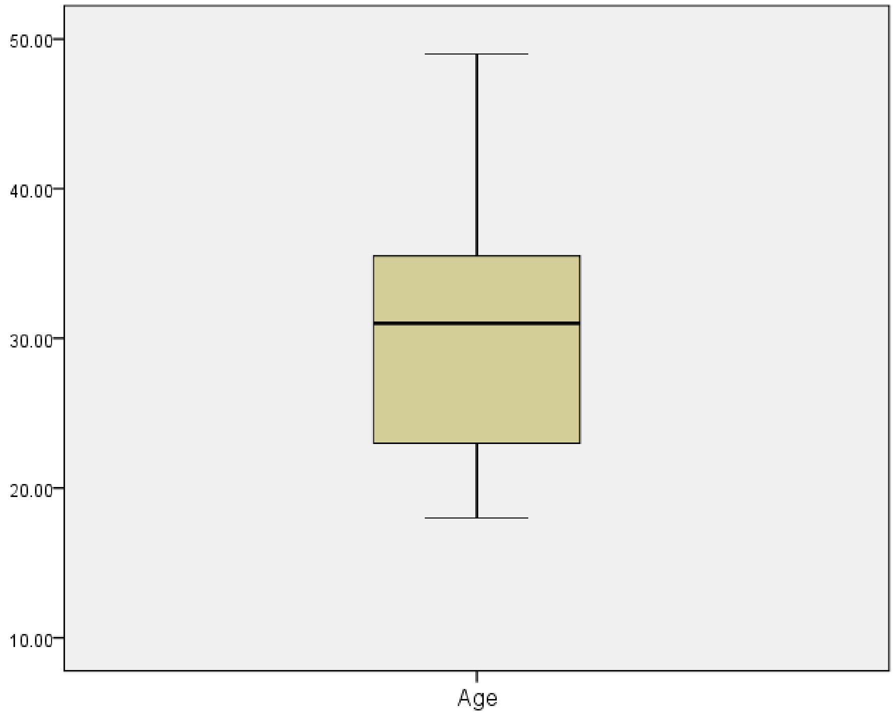


Figure 50. Distribution of age of the participants

Figure 51 represents how the ages are distributed by concentrating on the different groups of participants who provided their age.

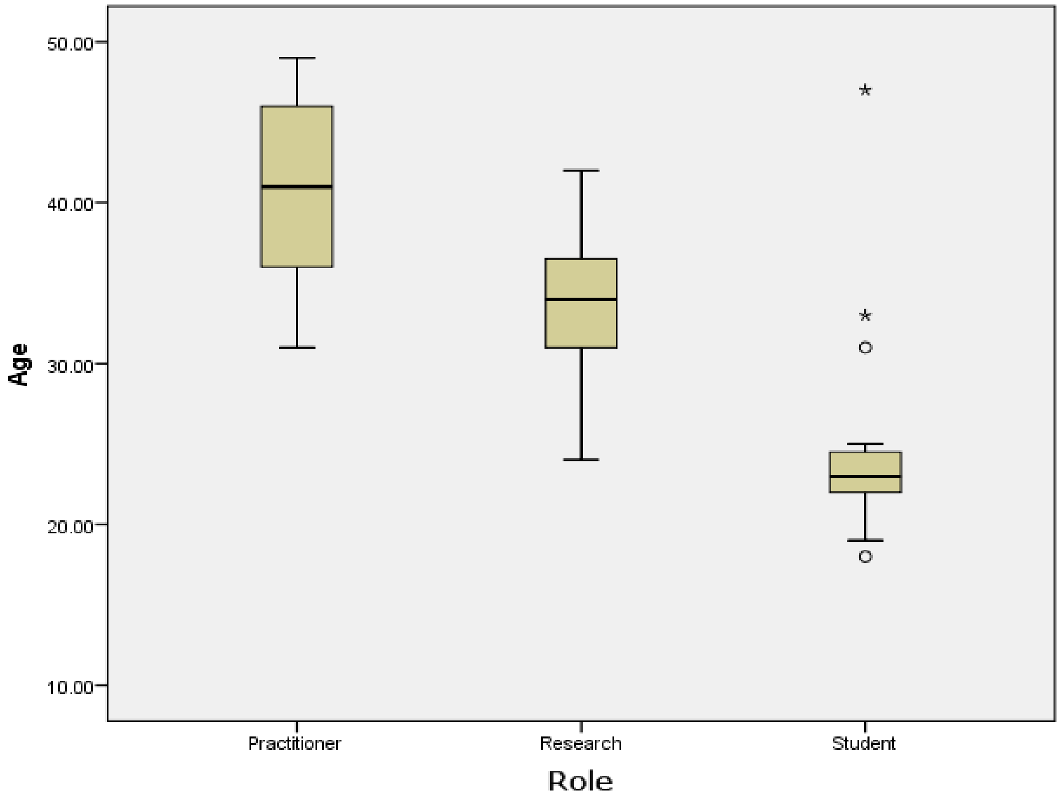


Figure 51. Distribution of age of the participants by roles

Those participants with experience in GSD are grouped in Table 28 according to their roles, and their months of experience is also detailed. Figure 52 shows the box plot of this distribution, revealing that practitioners are the most experienced group, followed by researchers, whereas the students, as expected, do not have any experience, which is why they should show a higher learning curve in this analysis.

Table 28. Participants' months of experience in GSD

Role		Statistic
Practitioner	Median	6.0000
	Variance	3088.250
	Minimum	.00
	Maximum	115.00
	Range	115.00
Researcher	Median	7.0000
	Variance	429.467
	Minimum	.00
	Maximum	48.00
	Range	48.00
Student	Median	.0000
	Variance	25.000
	Minimum	.00
	Maximum	10.00
	Range	10.00

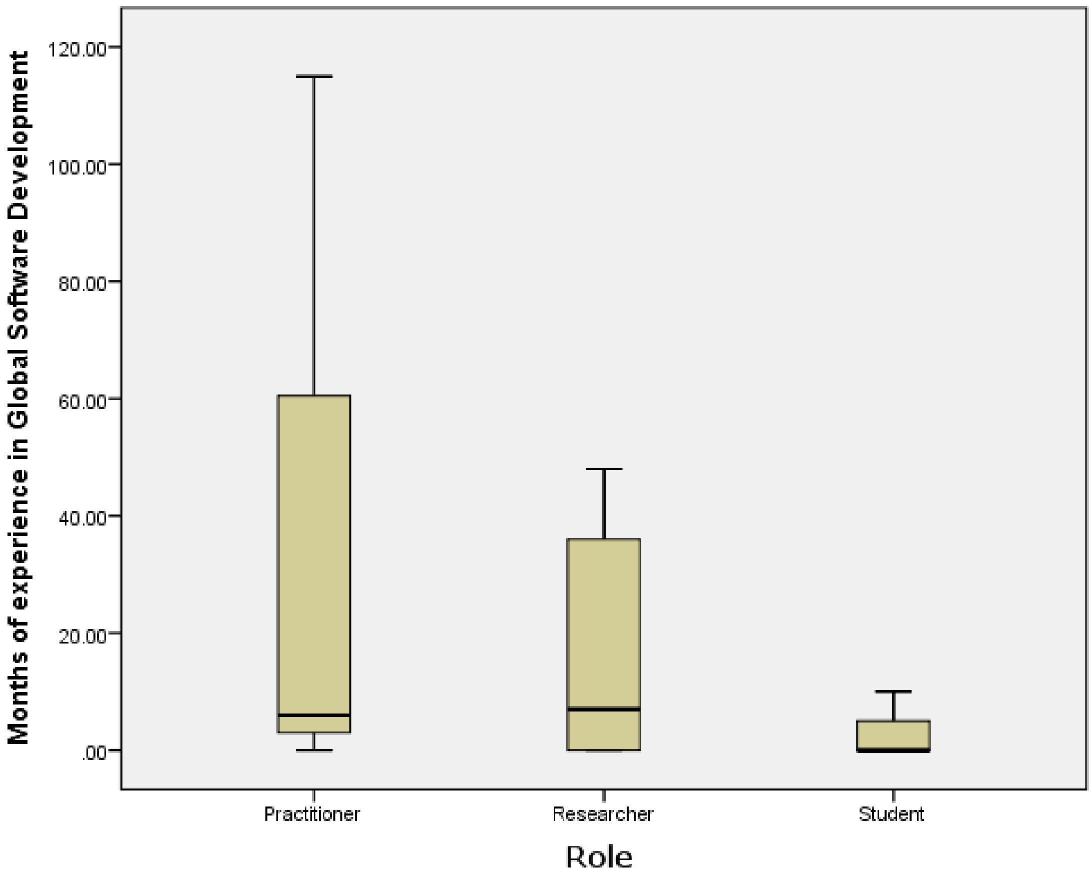


Figure 52. Participants' months of experience in GSD role

The practitioners interviewed represented a cross-section of organization sizes in software development, as shown in Table 29.

Table 29. Characteristics of the practitioners' companies

Size of the company (no. persons)	Size of the IT department (no. persons)
350	300
10	5
2000	1600
70	20

Those participants with experience in GSD projects additionally provided the nationalities of the members they have interacted with. As can be seen in Table 30, the different languages and cultures that the participants have interacted with on a daily basis, represent 30 countries, and this also reveals that two of the participants had previous experience in dealing with Indians, and one of them had interacted with Mexicans, which are the two nationalities with which the participants were trained in the scenarios.

Table 30. Nationalities with which participants have interacted on GSD projects

Participant	Nationalities
1	Argentina; Bulgaria; Croatia; Czech Republic; France; India; Ireland; Kazakhstan; Netherlands; Philippines; Poland; Portugal; Romania; South Africa; Spain; Vietnam
2	Argentina; Spain
3	Australia; Canada; Pakistan
4	Brazil; Canada; Switzerland
5	Chile; India; Turkey
6	Ireland; Israel; Italy; Netherlands; Saudi Arabia; Switzerland; United Arab Emirates; United States
7	Mexico; Spain
8	Spain; United Kingdom; United States

7.6 Results

The results from the survey, pre post questionnaire, baseline and opinion questionnaire and the logs of the interaction obtained through the evaluation process are analyzed in this section by addressing the research questions. All the statistical methods applied in this section are non-parametric, as tests for normality show that the distributions do not follow a normal distribution. All the normality tests are presented in Appendix IX.

7.6.1 Analysis of the interactions

The participants' interactions, including the log of the conversation, rules fired and scores obtained are now analyzed to explore relevant outcomes for this study.

Scenarios difficulty

At an initial stage it is interesting from a statistical point of view to observe whether there is any difference in the difficulty of two scenarios that were involved in the study. A potential difference could have an impact on the learning achieved in the scenarios.

Figure 53 shows a box plot that visualizes the scores obtained for both scenarios. Only two outliers are observed (both for Scenario B). For both scenarios, the students have usually scored over 60 out of 100, although a higher performance can be appreciated in Scenario A. However, it is necessary to check whether this difference is significant.

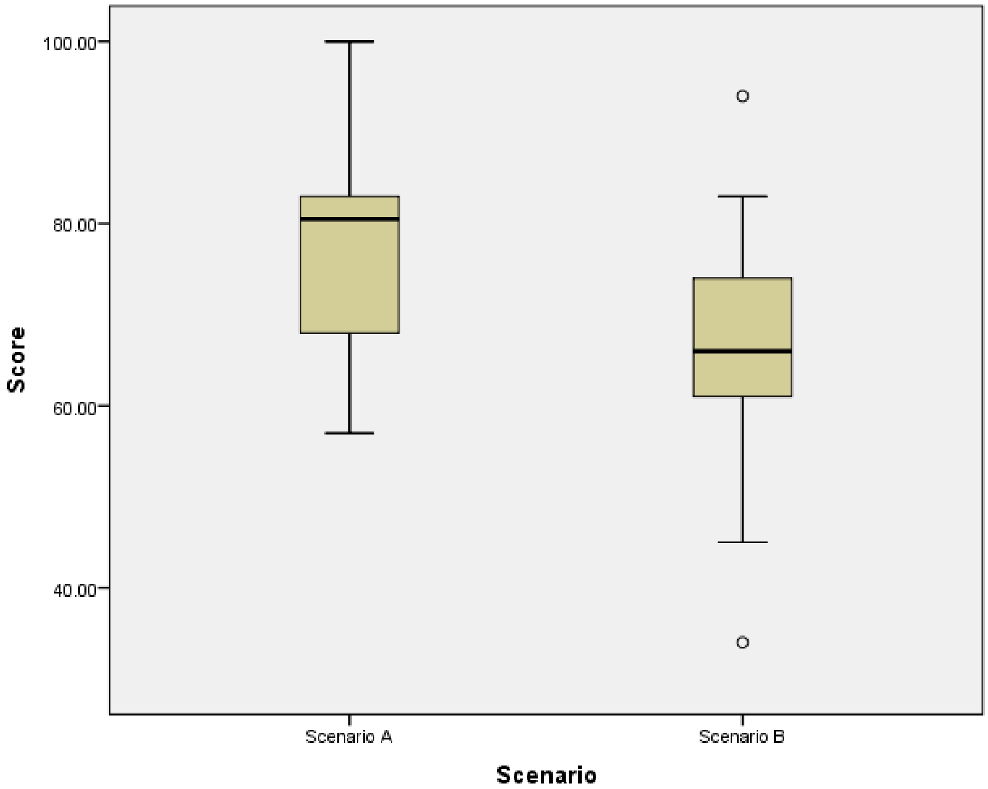


Figure 53. Scenarios difficulty (box plot)

The non parametric Mann-Whitney test was applied to compare whether the difference in the scores obtained for both scenarios was significant. Table 31 indicates that Scenario B is more difficult than Scenario A, as it has a lower score (of 66%), as shown by the median rank.

Table 31. Scenarios' difficulty (ranks)

	Scenario	N	Median
Score	Scenario A	34	80.5000
	Scenario B	34	66.0000
	Total	68	

Table 32. Scenarios' difficulty (test statistics)

	Score
Mann-Whitney U	265.500
Wilcoxon W	860.500
Z	-3.845
Asymp. Sig. (2-tailed)	.000

Regardless of the order in which the scenarios were enacted, Scenario B always produced a lower score. Table 32 shows the actual significance value of the test, providing the test statistic, U value, and the asymptotic significance (2-tailed) p-value, which leads to the following conclusion:

Conclusion: The score was statistically higher in **Scenario A** (U = 265.5, p =<0.001), so the null hypothesis, which states that there is no difference between the two scenarios can be rejected. There is a difference and it is highly likely that this is not a random occurrence.

These results must be taken into account for the subsequent statistical analysis, as it will not be meaningful to compare all the scores in both scenarios. However, in order to overcome this problem, another dimension can be used to compare the results in both scenarios. Both scenarios were designed to include the same number of rules of each type. It is therefore possible to check each rule independently in both scenarios to test whether any learning has occurred from one scenario to another.

Table 33. Rules difficulty (test statistics)

	Mann-Whitney U	Wilcoxon W	Z	Asymp. Sig. (2-tailed)
Formulate criticism	527.000	1122.000	-1.758	.079
Tendency to always agree	17.000	612.000	-7.847	.000
Use of direct/indirect style	447.000	1042.500	-1.729	.084
Overuse of certain verbs	255.000	850.000	-5.043	.000
Deal with conflictive situations	221.000	816.000	-5.403	.000
Misinterpretations and misunderstandings	339.500	934.500	-3.973	.000
Gain the team's confidence and trust	441.000	1036.000	-2.541	.011
Ambiguity/lack of information	85.000	680.000	-7.021	.000

Table 33 details the Mann-Whitney test statistics for the different rules. These tables are summarized below for each rule:

Formulate criticism: upon observing the means (in Table 33) it would appear that both scenarios are quite similar (U = 527.000, p =0.079)

Tendency to always agree: the means are different (U = 17.000, p =<0.001)

Use of direct/indirect style: both scenarios have similar means (U = 447.000, p =0.084)

Overuse of certain verbs: Differing means (U = 255.000, p =<0.001)

Deal with conflictive situations: Differing means (U = 221.000, p =<0.001)

Misinterpretations and misunderstandings: Differing means (U =339.500, p =<0.001)

Gain the team's confidence and trust: Differing means (U =441.000, p =0.011)

Ambiguity/lack of information: Differing means (U =85.000, p =<0.001)

Upon considering these results, those rules with a p-value > 0.05 are: Formulate criticism and Use of direct/indirect style. For these two rules, it is inferred that there is no significant difference between scenarios. Statistical comparisons can therefore be accurately made by using these dimensions. However, after checking both dimensions, it was found that the rule "Formulate criticism" has been fired too few times, so it is not a good comparer. The "Use of direct/indirect style" will be used for these comparisons.

Conclusion: Despite the difference in the difficulty of the scenarios, a statistical analysis can be conducted for the scores for the rule “Use of direct/indirect style”, since rules were fired in an equivalent manner in both scenarios, independently of the order in which the scenarios were enacted by the participant.

Proof of learning after the second execution

This section aims to discover whether there is any improvement in the participants’ performance when they execute the second scenario (which can be either scenario A or B, depending on which order they executed the scenarios), after the experience of having executed the first one. In order to achieve this objective, the rule “Use of direct/indirect style” will be analyzed. The first step was to check that the “Use of direct/indirect style” rule had been fired following a non-normal distribution (proved in Appendix IX), and a non parametric statistical method was therefore required.

The next step consisted of comparing both scenarios using the rule “Use of direct/indirect style” by applying the **Wilcoxon Test**. The objective was to compare whether there was any difference in how the rules were fired when the second scenario was executed. It was therefore necessary to consider the following two cases: when Scenario A was enacted before Scenario B and when Scenario B was enacted before Scenario A.

Table 34. Use of direct/indirect style Scenario A – Scenario B (descriptive statistics)

	N	Median
Use of direct/indirect style Pre	17	1.0000
Use of direct/indirect style Post	17	1.0000

Table 35. Use of direct/indirect style Scenario A – Scenario B (Wilcoxon test)

	Use of direct/indirect style Post – Use of direct/indirect style Pre
Z	-1.645
Asymp. Sig. (2-tailed)	.100

Table 34 shows that after executing Scenario A, there was an improvement in how the rule “Use of direct/indirect style” was fired during Scenario B. However, when applying the Wilcoxon signed-rank test, whose results are shown in Table 35, the p-value obtained is greater than 0.05. It is not, therefore, possible to conclude that the sample median differs from the hypothetical median. This means that the difference in how the rules were fired in both scenarios is not significant and it is not possible to conclude from this data that there was any improvement when executing the second scenario.

In the case of reversing the order of the scenarios, Table 36 shows that the number of times that the rule was fired, increased during the second execution. Moreover, as in the previous case, the p-value 0.138>0.05 (see Table 37) lead to the conclusion that there is no significant difference in how the rule was fired in both cases.

Table 36. Use of direct/indirect style Scenario B – Scenario A (descriptive statistics)

	N	Median
Use of direct/indirect style Pre	17	1.0000
Use of direct/indirect style Post	17	1.0000

Table 37. Use of direct/indirect style Scenario B – Scenario A (Wilcoxon test)

	Use of direct/indirect style Post – Use of direct/indirect style Pre
Z	-1.485
Asymp. Sig. (2-tailed)	.138

Figure 54 represents the means obtained in the previous steps, comparing the results for both scenarios in a graphical manner when they were enacted first and when they were enacted second. The rule "Use of direct/indirect style" was better when Scenario A was executed first, but was worse when Scenario B was executed first.

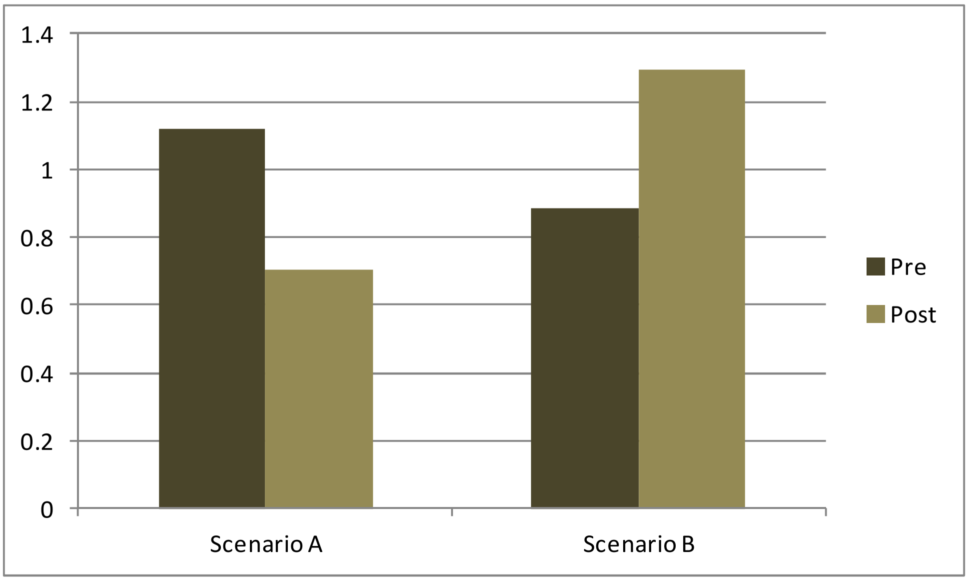


Figure 54. Results obtained in the execution of the "Use of direct/indirect style" rule

In order to detail these results, Figure 55 and Figure 56 represent the distribution of the number of times that this rule was fired. Figure 55 presents the behavior in the case of the first execution when Scenario A and Scenario B were enacted first. Figure 56 presents the behavior for the second execution when Scenario A and Scenario B were enacted second. An interesting piece of information from Figure 55 is that when Scenario A is enacted first, almost all the participants had the same problem with the rule. All of them fired it at least once (the medians are similar and equal to one in both cases). However, when Scenario A was enacted second (see Figure 56), this pattern disappears, and many participants did not fire the rule. This result can be attributed to a certain level of learning that may have taken place when enacting Scenario B first.

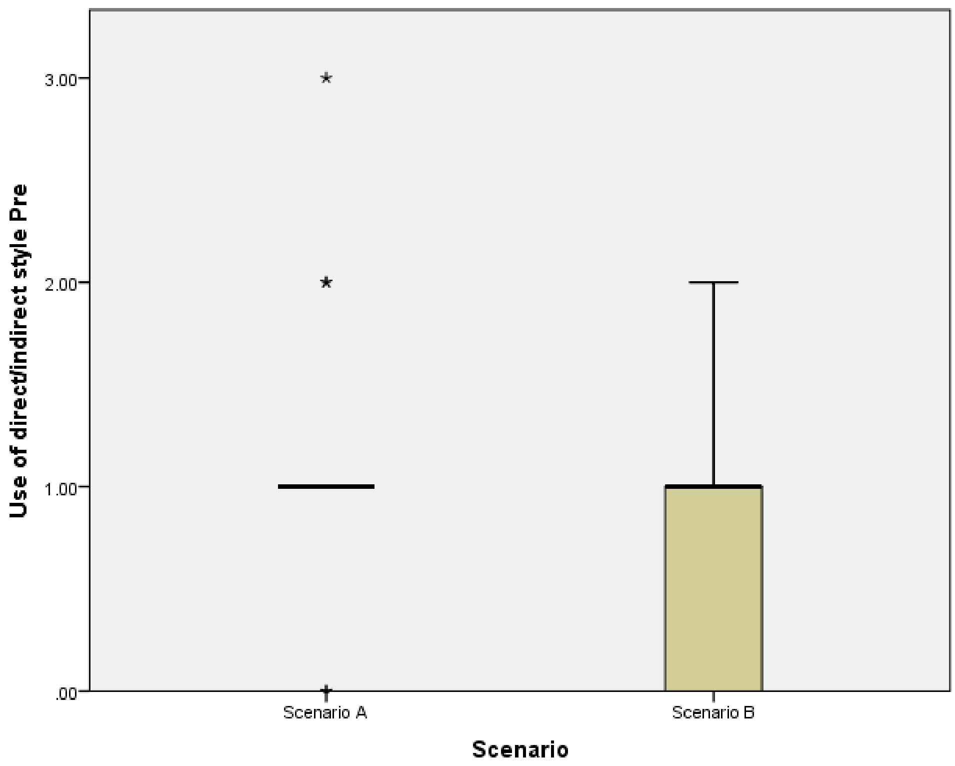


Figure 55. Rules fired in the first round (distribution boxplot)

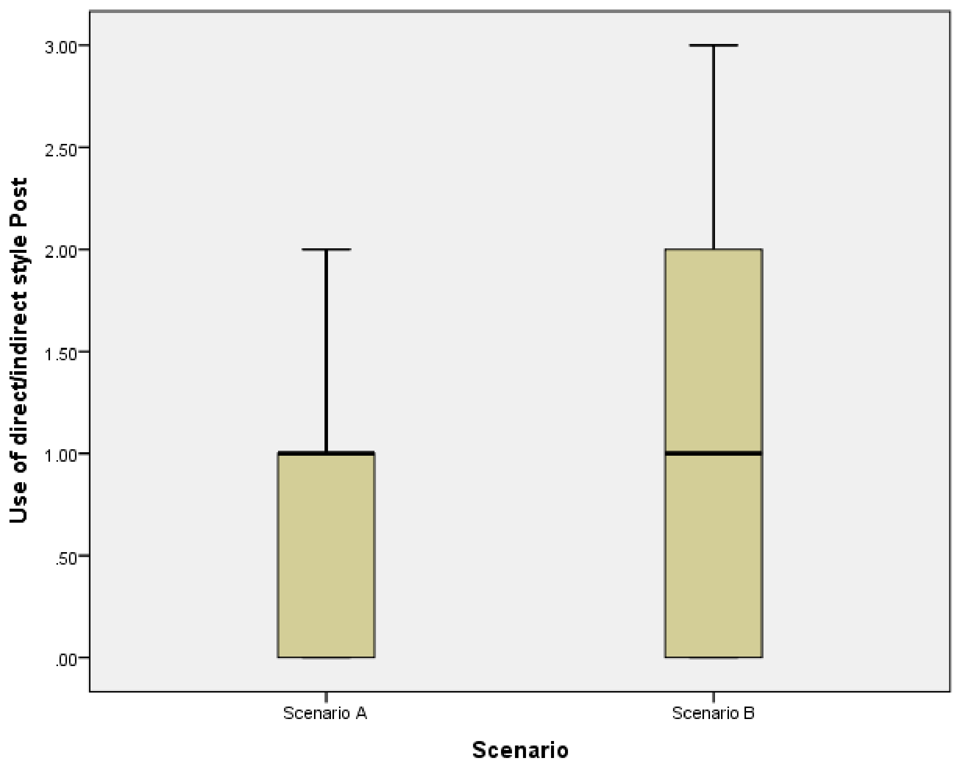


Figure 56. Rules fired in the second round (distribution boxplot)

The following observations arose from the analysis of this data:

1. The rule "Use of direct/indirect style" was more likely to be fired in Scenario A. Because of the context of the interaction in this scenario, this rule was not easy to avoid even in those cases in which participants had the experience of having enacted Scenario B first.
2. The context of the simulations is different in both scenarios. Although learning could occur during the execution of the first scenario, this does not mean that the participant will be able to detect the same problem in a completely different context involving different cultures.
3. One of the problems of using the Wilcoxon test is that it has little power when applied to small samples. When samples are reduced, the Wilcoxon test tends to obtain p-values of over 0.05, and extending the study with more participants might therefore provide these results with more relevance.

Conclusion: It was more likely to make a mistake in the "use of direct/indirect style" in Scenario A because of the context of the interaction. As the context in both scenarios is different, comparing the learning from one scenario to another is not viable by testing the output from the automated assessment method.

7.6.2 Can VENTURE enhance a user' knowledge and skills as regards how to communicate with people from different cultures?

We shall now analyze the pre and post questionnaires in order to study whether learning was achieved by using this framework. This section covers those questions regarding the cultural learning. Appendix IX shows the Shapiro-Wilk tests, which conclude that the answers do not follow a normal distribution. The comparisons are therefore conducted by applying non parametric tests.

Table 38 and Table 39 provide a statistical summary of the responses for the questions concerning cultural awareness before and after executing the scenarios for Scenario A and Scenario B, respectively. It can be appreciated that in both cases there is an improvement in the mean and the median of the answers provided. It must be noted that the scores shown in these tables range from 1 (strongly disagree) to 5 (strongly agree).

Table 38. Comparison of pre and post cultural questions - Scenario A (statistics)

	Median Pre	Median Post
Some cultures tend to always agree, others tend to be more critical and argue, providing reasons. I am aware of which of these styles is more common for Mexicans	1.50	4.00
I am aware that Mexicans tend to value family more than work	2.00	5.00
I am aware of Mexicans' respect for the chain of command	1.50	4.00
I am aware of Mexicans' willingness to work extra hours	1.50	4.00
I know the concept of the "Mum Effect"	1.00	4.00
I know which communication style it is more appropriate to use with a Mexican (direct or indirect)	1.00	3.82
I know how to start an icebreaking conversation with a Mexican	2.00	5

Table 39. Comparison of pre and post cultural questions - Scenario B (statistics)

	Median Pre	Median Post
I am aware of how people from an Indian culture feel about processes, rules and documentation	1.00	4.00
I am aware of Indians' punctuality when attending meetings or arriving at their workplace	2.00	4.00
I am aware of how open the Indian culture is as regards asking questions and seeking advice	1.00	4.00
I am aware of Indians' willingness to work extra hours/overtime	2.00	4.00
I am aware of how Indians can show concern when they do not agree with you	2.00	4.00
Some cultures tend to always agree, others tend to be more critical and argue, providing reasons. I know which of these styles is more common for Indians	2.00	4.00
I am aware of how the chain of command works in an Indian culture	1.50	4.00
I know the concept of "Unproductive Productivity"	2.00	4.00

Table 40 and Table 41 provide a comparison of the answers before and after the execution of Scenarios A and B, respectively. Negative ranks refer to those cases in which the question received a worse score in the post questionnaire than in the pre questionnaire. Positive ranks mean that the score was improved in the post questionnaire. In some cases the total number of answers differs for some questions, as the participants had the option of selecting a N/A (no answer) option.

The answers provided by the participants were considerably more positive in the post questionnaire for all the questions, indicating that they had the perception of having learnt by using the simulator.

Table 40. Comparison of pre and post cultural questions - Scenario A (ranks)

		N	Mean Rank	Sum of Ranks
Some cultures tend to always agree, others tend to be more critical and argue, providing reasons. I am aware of which of these styles is more common for Mexicans	Negative Ranks	0	.00	.00
	Positive Ranks	29	15.00	435.00
	Ties	2		
	Total	31		
I am aware that Mexicans tend to value family more than work	Negative Ranks	1	3.50	3.50
	Positive Ranks	28	15.41	431.50
	Ties	4		
	Total	33		
I am aware of Mexicans' respect for the chain of command	Negative Ranks	0	.00	.00
	Positive Ranks	30	15.50	465.00
	Ties	3		
	Total	33		
I am aware of Mexicans' willingness to work extra hours	Negative Ranks	1	3.50	3.50
	Positive Ranks	28	15.41	431.50
	Ties	4		
	Total	33		

		N	Mean Rank	Sum of Ranks
I know the concept of the “Mum Effect”	Negative Ranks	0	.00	.00
	Positive Ranks	28	14.50	406.00
	Ties	3		
	Total	31		
I know which communication style it is more appropriate to use with a Mexican (direct or indirect)	Negative Ranks	1	13.50	13.50
	Positive Ranks	30	16.08	482.50
	Ties	2		
	Total	33		
I know how to start an icebreaking conversation with a Mexican	Negative Ranks	1	5.00	5.00
	Positive Ranks	30	16.37	491.00
	Ties	2		
	Total	33		

Table 41. Pre and post cultural questions comparison Scenario B (ranks)

		N	Mean Rank	Sum of Ranks
I am aware of how people from an Indian culture feel about processes, rules and documentation	Negative Ranks	1	4.00	4.00
	Positive Ranks	28	15.39	431.00
	Ties	5		
	Total	34		
I am aware of Indians’ punctuality when attending meetings or arriving at their workplace	Negative Ranks	2	4.00	8.00
	Positive Ranks	24	14.29	343.00
	Ties	8		
	Total	34		
I am aware of how open the Indian culture is as regards asking questions and seeking advice	Negative Ranks	0	.00	.00
	Positive Ranks	29	15.00	435.00
	Ties	5		
	Total	34		
I am aware of Indians’ willingness to work extra hours/overtime	Negative Ranks	1	6.00	6.00
	Positive Ranks	24	13.29	319.00
	Ties	9		
	Total	34		
I am aware of how Indians can show concern when they do not agree with you	Negative Ranks	1	5.00	5.00
	Positive Ranks	29	15.86	460.00
	Ties	4		
	Total	34		
Some cultures tend to always agree, others tend to be more critical and argue, providing reasons. I know which of these styles is more common for Indians	Negative Ranks	2	5.00	10.00
	Positive Ranks	27	15.74	425.00
	Ties	5		
	Total	34		

		N	Mean Rank	Sum of Ranks
I am aware of how the chain of command works in an Indian culture	Negative Ranks	0	.00	.00
	Positive Ranks	31	16.00	496.00
	Ties	3		
	Total	34		
I know the concept of "Unproductive Productivity"	Negative Ranks	2	6.00	12.00
	Positive Ranks	25	14.64	366.00
	Ties	7		
	Total	34		

The next step consists of checking whether these differences are statistically significant. Table 42 and Table 43 show the results of the Wilcoxon test for the pre and post questionnaires for Scenario A and Scenario B, respectively. In both cases the p-value is lower than 0.05 for all the questions. The null hypothesis that there is no difference between the results in the pre and post questions cannot be accepted. It is therefore possible to assume that there was a significant improvement in cultural knowledge/awareness perceived by the participants after executing the simulations.

Table 42. Comparison of pre and post cultural questions - Scenario A (Wilcoxon test)

	Z	Asymp. Sig. (2-tailed)
Some cultures tend to always agree, others tend to be more critical and argue, providing reasons. I am aware of which of these styles is more common for Mexicans	-4.761[a]	.000
I am aware that Mexicans tend to value family more than work	-4.679[a]	.000
I am aware of Mexicans' respect for the chain of command	-4.863[a]	.000
I am aware of Mexicans' willingness to work extra hours	-4.673[a]	.000
I know the concept of the "Mum Effect"	-4.677[a]	.000
I know which communication style it is more appropriate to use with a Mexican (direct or indirect)	-4.658[a]	.000
I know how to start an icebreaking conversation with a Mexican	-4.814[a]	.000

a. Based on negative ranks.

Table 43. Comparison of pre and post cultural questions - Scenario B (Wilcoxon test)

	Z	Asymp. Sig. (2-tailed)
I am aware of how people from an Indian culture feel about processes, rules and documentation	-4.657[a]	.000
I am aware of Indians' punctuality when attending meetings or arriving at their workplace	-4.290[a]	.000
I am aware of how open the Indian culture is as regards asking questions and seeking advice	-4.770[a]	.000
I am aware of Indians' willingness to work extra hours/overtime	-4.267[a]	.000
I am aware of how Indians can show concern when they do not agree with you	-4.719[a]	.000
Some cultures tend to always agree, others tend to be more critical and argue, providing reasons. I know which of these styles is more common for Indians	-4.526[a]	.000
I am aware of how the chain of command works in an Indian culture	-4.924[a]	.000
I know the concept of "Unproductive Productivity"	-4.299[a]	.000

a. Based on negative ranks.

The participants' perceptions of their cultural learning were also requested at the end of the course, and are shown in Table 44. The scores report that after concluding the course they perceived that they had learnt about both cultures (Mexican and Indian).

Table 44. Cultural learning

Questions	Median	Mode
I have increased my knowledge of the Indian culture	4.00	4
I have increased my knowledge of the Mexican culture	4.00	4
I now have a better concept of how to adapt my style of speech to fit the context of the conversation	4.00	4
I have NOT increased my cultural knowledge	1.50	1
I would feel more comfortable interacting with people from Mexico	4.00	4
I would feel more comfortable interacting with people from India	4.00	4
Taking part on this course has increased my interest in other cultures	4.00	4

Conclusion: The participants' answers to the pre and post questionnaires show that, after completing each simulation, they have gained cultural knowledge and awareness. They also report that after concluding the course they perceived that they had learnt how to interact with both cultures (Mexican and Indian), and that they would therefore perform better in a real scenario.

7.6.3 Can VENTURE enhance a user' s knowledge and skills as regards how to communicate with people with different native languages?

The pre and post questionnaires for each scenario have been analyzed by following the same procedure as in the previous section in order to answer whether or not the participants perceived linguistic learning. Appendix IX shows the Shapiro-Wilk tests, which conclude that the answers do not follow a normal distribution. The comparisons are therefore conducted by applying non parametric tests.

Table 45 and Table 46 provide a statistical summary of the responses for questions regarding cultural awareness before and after executing the scenarios for Scenario A and Scenario B, respectively. It can be appreciated that in both cases there is an improvement in the mean and the median of the answers provided. It must be noted that the scores shown in these tables range from 1 (strongly disagree) to 5 (strongly agree).

Table 45. Comparison of pre and post cultural questions - Scenario A (statistics)

	Median Pre	Median Post
I am familiar with the concept of a false-friend	2.00	4.00
How many Spanish false-friends do you know?	1.00	3.00

Table 46. Comparison of pre and post cultural questions - Scenario B (statistics)

	Median Pre	Median Post
I am familiar with the kinds of mistakes that Hindi speakers make when speaking English	1.00	3.00

Table 47 and Table 48 provide a comparison of answers regarding the participants' perception of language learning before and after the execution of the scenarios. Negative ranks refer to those cases in which the question received a worse score in the post questionnaire than in the pre questionnaire. Positive ranks mean that the score was improved in the post questionnaire. In this case, the results are also positive, showing that the participants have a perception of having increased their knowledge after the execution of the scenario.

Table 47. Comparison of pre and post questionnaires - Scenario A (ranks)

		N	Mean Rank	Sum of Ranks
I am familiar with the concept of a false-friend	Negative Ranks	0	.00	.00
	Positive Ranks	22	11.50	253.00
	Ties	12		
	Total	34		
How many Spanish false-friends do you know?	Negative Ranks	1	4.50	4.50
	Positive Ranks	16	9.28	148.50
	Ties	17		
	Total	34		

Table 48. Comparison of pre and post cultural questions - Scenario B (ranks)

		N	Mean Rank	Sum of Ranks
I am familiar with the kinds of mistakes that Hindi speakers make when speaking English	Negative Ranks	2	8.50	17.00
	Positive Ranks	24	13.92	334.00
	Ties	8		
	Total	34		

The next step consists of checking whether these differences are statistically significant. Table 49 and Table 50 show the results of the Wilcoxon test for the pre and post questionnaires for Scenario A and Scenario B, respectively. In both cases, the p-value is lower than 0.05 for all the questions. The null hypothesis that there is no difference between the results in the pre and post questions cannot be accepted. It is therefore possible to assume that there was a significant improvement in the linguistic knowledge perceived by the participants after executing the simulations.

Table 49. Comparison of pre and post cultural questions - Scenario A (Wilcoxon test)

	Z	Asymp. Sig. (2-tailed)
I am familiar with the concept of a false-friend	-4.177[a]	.000
How many Spanish false-friends do you know?	-3.491[a]	.000

a. Based on negative ranks.

Table 50. Comparison of pre and post cultural questions - Scenario B (Wilcoxon test)

	Z	Asymp. Sig. (2-tailed)
I am familiar with the kinds of mistakes that Hindi speakers make when speaking English	-4.081[a]	.000

a. Based on negative ranks.

The participants' perceptions of their linguistic learning were also requested at the end of the course, and are shown in Table 51. The scores report that after concluding the course they perceived that they had learnt about both languages (Spanish and Hindi).

Table 51. Linguistic learning

Questions	Median	Mode
I have gained linguistic understanding with which to interact with people from other cultures	4.00	4

Conclusion: The participants' answers to the pre and post questionnaires show that, after completing each simulation, they have gained linguistic knowledge and awareness as regards the kinds of problems that occur in GSD. As a consequence, they also report that after concluding the course they perceived that they had developed the skills needed for a better understanding of people with different languages.

7.6.4 Can VENTURE enhance a user's knowledge as regards GSD challenges and concepts, and the skills needed to tackle GSD problems?

In order to study the learning as regards GSD-related concepts and challenges, the pre and post questionnaires are similarly studied. In this case, Scenario B considered explaining the concept of "Follow-the-Sun". Table 52 shows statistics regarding the answers provided by the participants. It must be noted that the scores shown in these tables range from 1 (strongly disagree) to 5 (strongly agree). The results show that there is an improvement in the knowledge of the concept.

Table 52. Comparison of pre and post GSD questions - Scenario B (statistics)

	Median Pre	Median Post
I know the concept of "Follow-the-Sun"	2.50	4.00

Table 53 provides a comparison of answers regarding the participants' perceptions of GSD learning before and after the execution of the scenario. Negative ranks refer to those cases in which the question received a worse score in the post questionnaire than in the pre questionnaire. Positive ranks mean that the score was improved in the post questionnaire. The results obtained are positive, showing that the participants have a perception of having increased their knowledge after the execution of the scenario.

Table 53. Comparison of pre and post GSD questions - Scenario B (ranks)

		N	Mean Rank	Sum of Ranks
I know the concept of "Follow-the-Sun"	Negative Ranks	0	.00	.00
	Positive Ranks	20	10.50	210.00
	Ties	14		
	Total	34		

The Wilcoxon test (non parametric) is applied in order to discover whether these differences are significant. Table 54 shows that the p-value is lower than 0.05 for that question. The null hypothesis that there is no difference between the results in the pre and post questions cannot be accepted. It is therefore possible to assume that there was a significant improvement in the GSD knowledge perceived by the participants after executing the simulation with regards to this concept.

Table 54. Comparison of pre and post cultural questions - Scenario B (Wilcoxon test)

	Z	Asymp. Sig. (2-tailed)
I know the concept of "Follow-the-Sun"	-3.962[a]	.000

a. Based on negative ranks.

The participants' perception of their GSD learning was also requested at the end of the course. The statistics as regards the scores provided are reflected in Table 55, showing that the perception of GSD learning is generally good.

Table 55. GSD learning

Questions	Median	Mode
I am more confident when confronting conflicts, misunderstandings and problems in Global Software Development	4.00	4
I have more knowledge as regards how to gain the confidence and trust of the person I am interacting with	4.00	4
I have learnt to detect missing information and misinterpretations of the person I am interacting with	4.00	4
I think this experience has increased my ability to create professional relationships with people who think differently from me	4.00	4
I have learnt new ways in which to manage conflicts and disagreements	4.00	4
I think this experience will make me more effective in collaborating and working within a multidisciplinary and multicultural team	4.00	4
I have acquired/reinforced new knowledge as regards Global Software Development	4.00	5

Conclusion: The participants report that after concluding the course they perceived that they were more aware of the kinds of problems that can occur in GSD. They also report that they have developed skills that will allow them to interact in GSD settings better.

7.7 Participants' perception

The analysis of the previous research questions may have been influenced by the participants' opinions of the scenarios and also of the system and procedure followed. This section analyses this perception in order to discover possible problems during the use of VENTURE.

The perceptions of the quality and characteristics of the simulations were initially gathered after each execution. Table 56 and Table 57 show the median for the scores obtained in the different questions for Scenarios A and B, respectively. The results show that the scenarios were clear and enjoyable for the participants and that they perceived the scenarios to be representative of GSD settings.

Table 56. Perceptions of scenario A

Questions	Median	Mode
The length of the simulation was too long	2.50	3
The story in the scenario was clear, concise and complete	4.00	4
The objective of the scenario was clear	4.00	5
The training scenario was enjoyable	4.00	5
The training scenario was engaging	4.00	4
The training scenario was difficult to understand	2.00	1
The training scenario was representative of a Global Software Development environment	4.00	4
The training scenario helped me to improve the skills required in Global Software Development	4.00	4
It was always clear how to interact with the Virtual Agents	4.00	4
Overall, I am satisfied with the scenario outcomes	4.00	4

Table 57. Perceptions of scenario B

Questions	Median	Mode
The length of the simulation was too long	3.00	3
The story in the scenario was clear, concise and complete	4.00	4
The objective of the scenario was clear	4.00	4
The training scenario was enjoyable	4.00	4
The training scenario was engaging	4.00	4
The training scenario was difficult to understand	2.00	1
The training scenario was representative of a Global Software Development environment	4.00	4
The training scenario was effective in helping me learn Global Software Development concepts	4.00	5
The training scenario helped me to improve the skills required in Global Software Development	4.00	4
It was always clear how to interact with the Virtual Agents	4.00	4
Overall, I am satisfied with the scenario outcomes	4.00	4

Conclusion: Both scenarios were perceived to be helpful as regards providing effective GSD training in an enjoyable manner. The outcomes are well valued and the interactions with the Virtual Agents acceptable.

Since one of the future objectives of this research is to apply the framework in real educational environments, its usability is an important point to consider. A usable tool will lead to a better acceptance in academic and industrial environments. In this respect, Table 58 shows that the perception of the usability of the system is acceptable by considering the median of the answers. However, this qualitative analysis does not provide feedback on possible improvements. The participants' suggestions as regards usability improvements are analyzed in the following section.

Table 58. Usability

Questions	Median	Mode
The system was simple to use	4.00	4
It was easy to learn to use this system	5.00	5
I was able to complete the tasks and scenarios quickly using this system	4.00	4
The system is stable and I did not receive error messages	5.00	5
The description of the scenarios is clear	4.50	5
The organization of information on the system screens was clear	4.00	4
The information was effective in helping me complete the tasks and scenarios	4.00	4
The interface of this system was pleasant	4.00	4
I liked using the interface of this system	4.00	4
This system has all the functions and capabilities I expect it to have	4.00	4
I feel comfortable using this system	4.00	5
I feel confident when using this system	4.50	5
This system performs the tasks that it was intended to	4.00	4
The information in this training is valuable	4.00	4
I feel uncomfortable when using this system	2.00	1
Overall, I am satisfied with this system	4.50	5

Conclusion: The system was perceived as usable, simple to use and clear, which means that it is easy to learn how to use it.

In order to be more specific, the participants' perceptions of the e-learning platform and the chat simulator were also requested. Table 59 and Table 60 show the mean of the results for both cases, showing that the experience was acceptable, obtaining a minimum of 4 out of 5 in the punctuation in all cases. If Figure 57, which describes the enjoyment of using the chat simulator of the different groups of participants, is studied it will be observed that the group that obtained the best scores is that of the students.

Table 59. Perceptions of the e-learning platform

Questions	Median	Mode
The e-learning platform is easy to navigate and use	4.00	4
The materials and explanations are simple and well organized	4.00	5
I always knew which activities I had to work on	4.00	4
I like the way this course is delivered in an interactive manner	4.50	5
The final reports are clear and help me to understand my problems and my score	4.00	4
I became confused when using the e-Learning platform	1.00	1

Table 60. Perceptions of the chat simulator

Questions	Median	Mode
The Chat Simulator can reproduce Global Software Development scenarios in an accurate manner	4.00	4
Simulated conversations provide awareness of the problems that can appear in Global Software Development	4.00	5
I felt involved in the scenario	4.00	4
The Virtual Guide can highlight and provide useful feedback on Global Software Development	4.00	4
I enjoyed using the Chat Simulator	4.00	4
I learnt more when I failed	4.00	4
The Chat Simulator is easy to use	4.00	4
I was able to use the simulation without getting lost	4.00	4

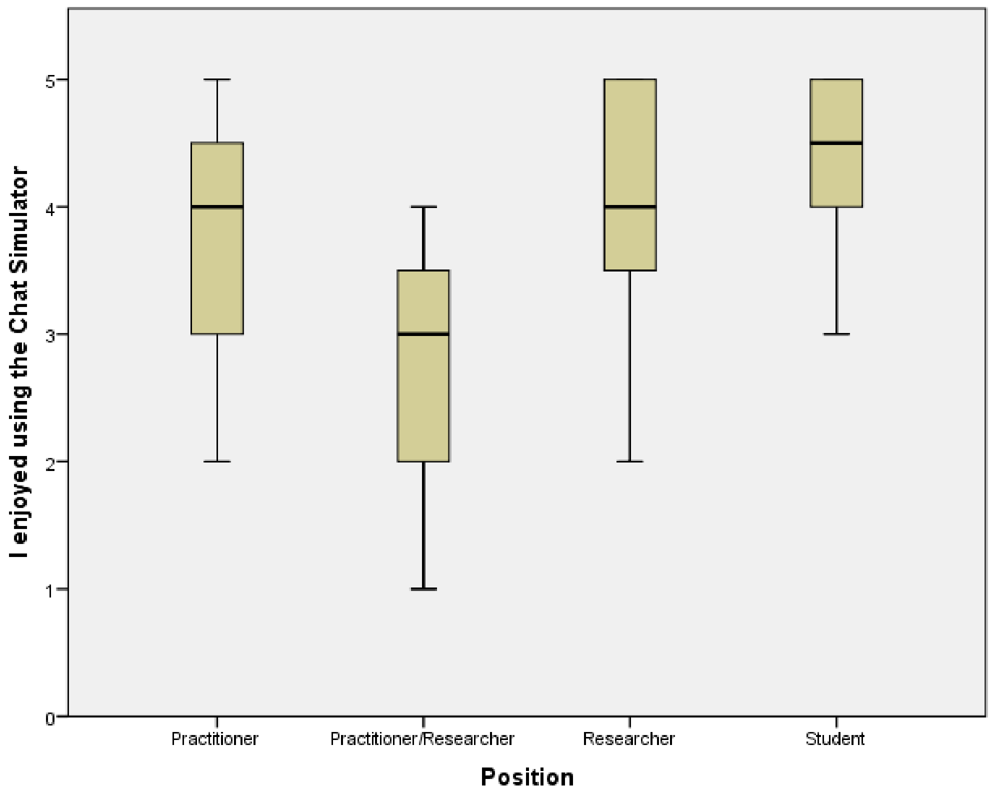


Figure 57. Perceptions of the chat simulator

The perceptions of the assessment method were gathered with the same aim of improving the experience and usefulness of the framework, as is reflected in Table 61 which shows similar positive results that were, in this case, more positive for the groups of students and researchers (see Figure 58).

Table 61. Perceptions of the assessment method

Questions	Median	Mode
I was aware of the learning outcomes at the beginning of the course	4.00	4
The assessment information is easy to understand and interpret	4.00	4
The feedback provided facilitated learning	4.00	4
The reports helped me to understand the topics trained	4.00	4
The scores I got seem fair	4.00	4
The assessment motivated me to improve my performance	4.00	4

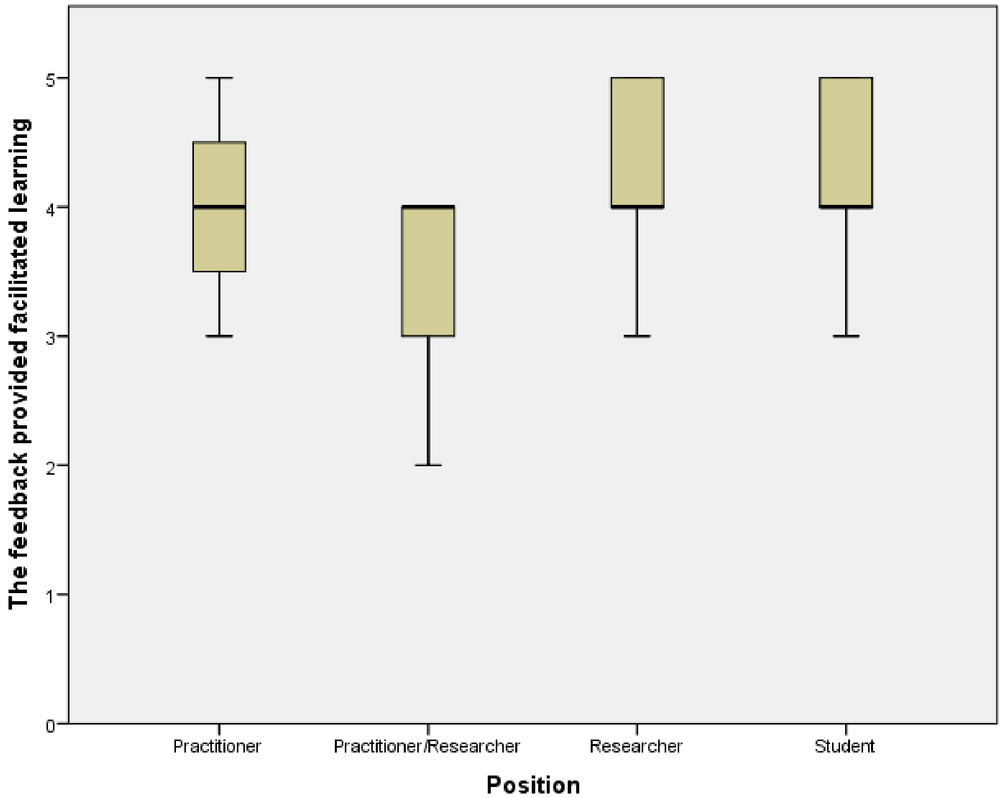


Figure 58. Perceptions of assessment method

Finally the overall perception of using VENTURE has been punctuated with a mean of 4 out of 5 in all the questions (see Table 62) and the satisfaction as regards the learning outcomes has obtained a similar overall punctuation (see Table 63).

Table 62. Overall engagement

Questions	Median	Mode
I enjoyed using VENTURE	4.00	4
This e-learning platform has the potential to be used by students/members of my university/company	4.00	4
I prefer participating on a training course with VENTURE than in traditional classes	4.00	4
Teaching Global Software Development is useful	4.00	4
Using this tool has changed my perception of the usefulness of e-learning systems	4.00	4

Table 63. Satisfaction as regards learning outcomes

Questions	Median	Mode
Overall satisfaction with the learning outcomes	4.00	4

Conclusion: The perceptions of the e-learning platform, the chat simulator and the assessment method are positive, and the learning outcomes of the experience are recognized by the participants. According to the participants' criteria, the system may have the potential to be implemented in educational environments.

7.8 Participant's comments

In addition to the quantitative analysis previously analyzed, the surveys also gave the participants the option of providing their opinion, which highlighted good points, weak points and feedback on future improvements. The analysis of those comments is presented in this section.

7.8.1 Good points

The good points reported are mostly similar to those found in the Expert Evaluations. The participants suggested that the application of the framework in certain learning environments (university and companies) could be achievable and beneficial.

The concept of a simulation-based platform was well received, as additionally supported in the quantitative analysis. In the words of one of the students who participated in the study: "It is a useful and interesting tool to help me to learn something about a different working culture background as well as to improve my English".

The Virtual Guide's guidance and feedback was also considered to be effective and useful for coping with the scenario. However, as one of the participants mentioned, sometimes the students do not want to do what the Virtual Guide is suggesting, and would prefer to follow the conversation according to what they think, but it is a good method as regards inducing systematic mistakes from the students or making them learn specific subjects.

7.8.2 Weak points

One of the main weak points is related to the artificial intelligence that Virtual Agents are supposed to show. On many occasions during the simulations, the Virtual Agents did not understand participants' interactions. All the participants experienced this at least once during the entire process. In these cases, the Virtual Colleague informed the users that they did not understand and asked them to re-state the message with other words. Figure 59 graphically represents how frequent these user misunderstandings were, showing that this issue appeared more frequently in Scenario A. However, the number of times that this issue occurred is not worrying, as users were able to complete the scenarios without any inconveniences other than the time that they lost. Moreover, it must be noted that these figures also include those cases in which the participants really interacted in a wrong way, in some cases because they made a spelling mistake and in others because they were testing the system.

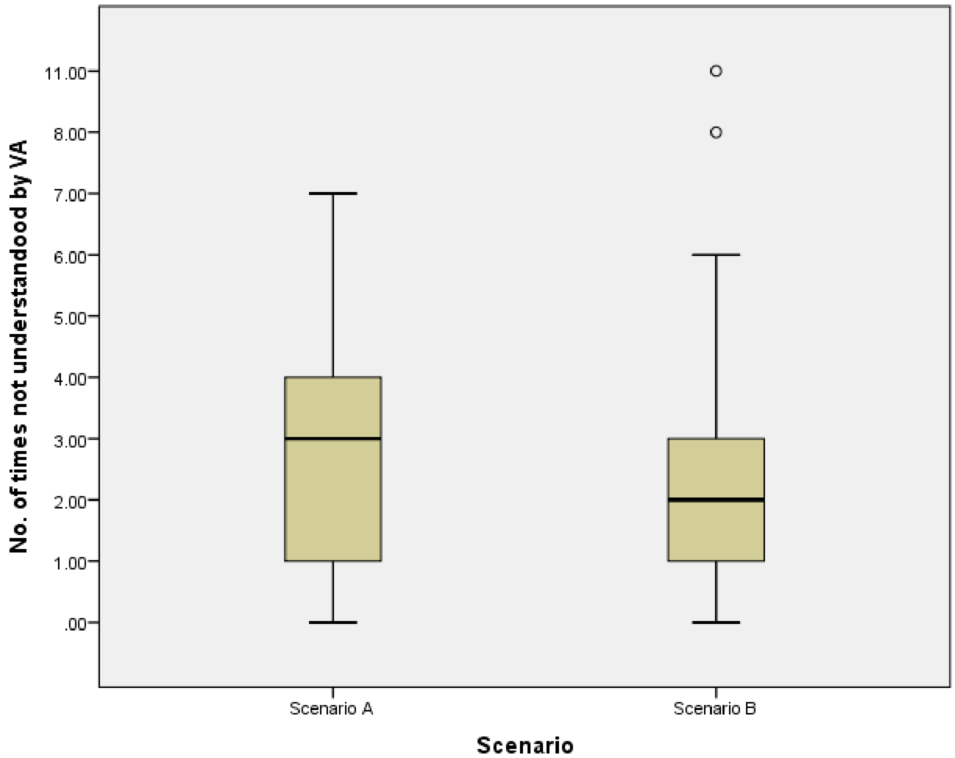


Figure 59. Number of times that the user was not understood by the VA

However, this issue signifies that in some cases the conversation can be misleading. The “Next stage” button was used on a total of seven occasions during the whole process by participants who got stuck and considered it necessary to skip to next stage of the interaction.

In the words of one participant “the virtual agent was a little inflexible”. Sometimes participants are led to express themselves in a way they would not usually do: “I was forced to use turns of phrase I would not normally use in a number of cases just to persuade the system to allow me to progress”. Moreover, sometimes the guidance can be misleading if the students lose the focus of the topic they are dealing with.

The idea behind using the framework is to iteratively improve the training scenarios by including the conversational knowledge required to understand the entries in the next iteration. However, no improvements were made during the length of the evaluation process to enable us to study exactly the same environment.

Some participants also thought that the length of the study was too long, which may have had some influence on their performance and the results of the surveys. In this respect, it was suggested that the simulation should be broken down into several small simulations dealing with specifics topics. However, the author did not consider the option of conducting an evaluation of this, as the motivation and engagement of the participants would decrease when executing a second scenario similar to the first one. In contrast, the evaluation method chosen permits the tiredness effect to be minimized, as the participants interact in two different contexts with two different cultures.

It was also found that the context in which the different problems occur in the conversation directly influence how the rules are fired, and comparing the learning by comparing the automatic scores of both scenarios is not a fair method by which to attain valid conclusions for this analysis. However, some participants whose level of English was not so advanced suggested that more time should be allowed to complete the scenarios, as they had difficulties in expressing themselves.

Finally, a lag in the typing was reported in some cases when the participants used the system remotely. However, this problem is not taken into consideration as the remote access was only part of the process in order to ensure that the participants executed all the steps in the evaluation correctly, and this problem would not exist in real learning environments.

7.8.3 Ideas for improvement

Of those participants who used the “Next stage” button, one of them commented that the current operation only goes to the next step of the conversation, thus causing one of the Virtual Agents to interact. An improvement to this could, however, be made by automatically sending the message that the participant should have written, thus permitting the participant to know how he/she should have interacted as expected by the Virtual Agents. This could solve some confusion in the case of students who may get lost during the interaction if they do not really understand the sense of a specific part of it.

One important issue reported in the previous section refers to the different ways of expressing the same thing. One of the participants reported that the Virtual Agent did not understand the subtleties of his speech. An iterative improvement to the simulations may be helpful to solve this problem. Since the logs of the interactions are stored automatically, it would be feasible to iteratively improve the scenarios by automatically looking for those parts of the interaction in which the students may have problems and manually editing the design of the scenario in order to consider that interaction in the future.

In relation to this point, one participant stated that to be able to move on to the next stage, on some occasions he had to repeat phrases that the Virtual Guide had used to guide him. This made him feel that it was possible to cope with some of the stages just by copying and pasting the Virtual Guide’s conversation. This issue must be considered when designing training simulations in order to provide an effective guidance by avoiding this effect.

With regard to the use of avatars, one participant suggested the inclusion of different avatars for the different cultures with typical facial features from the nationality they represent. The voices of the Virtual Agents when they speak to the user are similarly too synthetic, and the text-to-speech engine could be improved to consider different accents. Voice recognition was also suggested in order to provide telephonic interaction training in the future.

In the same line, one interesting feature proposed was that of simulating interactions with more than one person at the same time in order to provide training in stressful situations.

The inclusion of additional tutorials at the end of the simulation was suggested as a means to reinforce those points with which the student had more problems. In this respect, the inclusion of additional introductory material on the different cultures before initiating the simulation was also suggested. This point was not considered during the study in order to minimize the participants’ effort, but the simulator’s application in real learning environments should consider imparting this kind of concepts beforehand.

Finally, it was suggested that the simulator concept could be adapted in order for it to be integrated with development tools applied in companies, such as Eclipse or Visual Studio, thus simulating real interactions between developers. Although this possibility had already been considered in the framework, such an implementation could be the basis of future work.

7.9 Conclusion

This chapter presents an *evaluation* of the VENTURE framework which aims to test whether students, when interacting through the VENTURE tool, improve their knowledge on cultural, linguistic and GSD-related problems. This chapter therefore evaluates VENTURE learning objectives through a *validation* exercise that tests the framework's learning outcomes. The three main research questions that address these objectives are:

Q1. Can VENTURE enhance a user's knowledge and skills as regards how to communicate with people from different cultures?
Q2. Can VENTURE enhance a user's knowledge and skills as regards how to communicate with people with different native languages?
Q3. Can VENTURE enhance a user's knowledge as regards GSD challenges and concepts, and the skills needed to tackle GSD problems?

In order to provide answers to these research questions, the participants' performance was tracked during the simulations, and their knowledge and opinions were gathered using questionnaires. After analyzing this information, it can be concluded that the participants' perception is that the three points covered by the research questions: cultural learning, linguistic learning and GSD problems, challenges and concepts, can be learned by applying this framework.

Although there was a difference in the participants' performance in both scenarios, it was possible to discover that one of the rules defined in both scenarios could be used to conduct accurate statistical comparisons between scenarios. However, it was found that the context in which the different problems occur in the conversation directly influences how the rules are fired, and that comparing the learning by comparing the automatic scores of both scenarios is not a fair method by which to attain valid conclusions for this analysis.

The answers to the pre and post questionnaires on cultural differences show that after completing each simulation the participants have gained cultural knowledge and skills. They also reported that after concluding the course they perceived that they had learnt about both cultures (Mexican and Indian). The first research question was therefore covered with this analysis.

With regard to the second research question "Can VENTURE enhance a user's knowledge and skills as regards how to communicate with people with different native languages?", the participants' answers to the pre and post questionnaires on linguistic differences show that after completing each simulation they have gained linguistic knowledge and skills. They also report that after concluding the course they perceived that they were more aware of the kind of linguistic problems that can arise in GSD.

The analysis also showed that after concluding the course, the participants reported that they had perceived an improvement in their awareness of the kind of problems that can arise in GSD. They also reported having acquired the knowledge and skills needed to tackle specific GSD problems.

Questionnaires were also used to *verify* that the chat simulator component is a good solution for GSD training, and quality and usability factors were also measured in order to complete the *evaluation*,. The system was perceived to be usable, simple to use and clear. The perceptions of the e-learning platform, the chat simulator and the assessment method were positive, which is an important point if students and instructors are to accept this platform. In this respect, and bearing in mind that the learning outcomes were also recognized by the participants, it was also concluded that the system may have the potential to be implemented in educational environments, either academia or industry, according to the participants' criteria. Our future work will therefore be focused on achieving this objective of making VENTURE more robust and applicable in educational settings. One of the suggestions in this direction was to adapt the simulator concept in order for it to be integrated with tools applied in companies, which would be the basis of future work in order to facilitate its application in industry.

Some improvements should be made in order to achieve a more robust framework. For instance, after examining the logs of the interactions, it was found that adaptability to the participants' backgrounds is an important point to consider. As an example, it was noted that those participants who lacked skills in English writing took longer to complete the simulations. This background should therefore be taken into consideration when designing and assigning learning tasks.

Moreover, and although this was an expected outcome shared with the feedback from the Expert Evaluations, it was noted that in some cases the Virtual Agents were not sufficiently intelligent to understand some interactions. Although a greater effort in the design of the scenarios could help to solve many of these situations, the objective of a future implementation should be focused on minimizing the effort needed to design scenarios. Improving the Virtual Agents' intelligence will therefore need to be considered.

The main objectives of the future work should focus on avoiding situations in which the students get stuck when trying to express something in a way that the Virtual Agents are not expecting. To avoid this dead-end, the Virtual Agents' intelligence will need to be improved to detect a broader set of possibilities in an autonomous manner. Some ideas that relate to solving this challenge are:

- Automating the detection of grammar inaccuracies, notifying students about their mistakes and avoiding the situation of the Virtual Agents processing inaccurate expressions.
- Considering the different ways used to express something negative or positive. For example, on occasions saying “No” does not involve the use of a negative expression. Other indirect forms that could even depend on the context of the interaction could be used. This can be addressed both by automating the detection of certain expressions and providing automatic mechanisms in the scenarios designer in order to consider certain expressions in a guided manner during the design of the simulations.
- On occasions, the linguistic and cultural rules will not allow the students to express something that could be right just because they have used a keyword that has the potential to be part of a wrong interaction. However, the student could have used that keyword in a context that is right. This occurred several times during the simulations, and led us to believe that the definition of the rules will need to consider some conditions in which the rule would not be fired.
- In some cases in which the students' interaction is not important it could be considered that the flow of the interaction could follow the students' entry, regardless. The Virtual Agent could simulate understanding any interaction as long as no rules are fired. This would make the design of the scenarios easier in these cases, as it minimizes the amount of different interactions to consider when they are not really important for the simulation.

Finally, both scenarios were perceived to be helpful as regards providing effective GSD training in an enjoyable manner. The outcomes were well valued and the interactions with the Virtual Agents were considered acceptable by the participants. The inclusion of additional tutorials at the end of the simulation was suggested, and will be part of our future work as regards improving learning effectiveness.

Finally, weak points and ideas for improvements arose from the study of the interactions and participants' perception. This information will also be part of the future work detailed in the next chapter, which will be focused on improving the cost-effectiveness factor of the chat simulator, the e-learning platform and the framework by implementing the ideas for improvement analyzed. The objective of this last iteration for improving the framework is its future implementation in industrial and academic settings. Gathering a sufficient amount of GSD patterns and scenarios will be important as regards defining useful training material for the future.

7.10 Limitations

As a consequence of the application of the within-subjects approach there are some limitations regarding the results of this study. This section presents these limitations by considering the external, internal and construct validity of the research method applied:

- **External validity**. The number of participants, although sufficient for obtaining trusty statistical results, does not allow the results to be generalized to populations from countries and backgrounds not represented in the study. Also, the Spanish nationality is over-represented, and males are predominant over females in the sample. Although the sample included practitioners, there was not a sufficient amount of them for us to be able to draw accurate conclusions regarding differences in their learning and perception as compared to students and researchers. Increasing the number of practitioners could be an improvement for a future study.

- **Internal validity**. Effects which may threaten the internal validity of the within-subject experiment and which are present in this research include [362]:

    - Learning effect: As the same participants are involved in two training scenarios during the experiment, this may assist participants to learn how to use the environment in the first scenario; in consequence they will always perform the task more correctly in the subsequent session. In this study, the learning effect is controlled by changing the order in which the two groups of students participated in the scenarios.
    - Fatigue effect: However, given the difficulties of finding two different days for the participants, some of them conducted both sessions during the same day. Testing the *fatigue effect* is considered in this study by dividing it in two sessions and reversing the order of the sessions for the different participant groups (see Figure 1).
    - Persistency effect: In those cases in which the experiment is conducted on different days, a different level of persistency of the learning acquired in the first session may be observed.
    - *Hawthorne Effect* [370]: This refers to a psychological phenomenon that produces an improvement in human behavior or performance when they receive attention from superiors, clients, colleagues or the interviewers. In order to avoid this effect, in those cases in which the interviewers were present during the simulations, they kept their distance from the participant unless the participant required attention.

  The multiple ways in which the data were collected from the participant (including automated testing and subjective testing), would also make it difficult for the participant to falsify learning. However, the training sessions often took place during conferences and in university classrooms. Environments in which the tool may be used, such as organizations, are likely to be different.

  Only a limited set of examples were considered to test students' knowledge of culture, language and GSD. It is not therefore possible to extend the conclusions to other knowledge concepts or areas not dealt with in the evaluation.

  The participants' different levels of English may also have influenced some results. Finally, as the participants were contacted by the interviewers, this could have led them to be more positive in their answers or to answer what the interviewers expected from them.

- **Construct validity**. Conducting both sessions on the same day may also involve a limitation as regards proving the change in the participants' behavior, since this takes time to take place [371]. Although the sessions are delimited in time, owing to the fact that the participants train similar skills in both sessions, the *motivation effect* could also have an influence. In order to mitigate this problem, the scenarios were designed by considering different contexts and teaching different concepts. The fact that students have to achieve objectives and receive a score is a motivating factor. In addition, the participants had the option of conducting the sessions on different days. Students' learning approaches and outcomes can be influenced by a number of contextual variables beyond the actual design of a course. In this design, the students' interest in the learning task may differ [372].

Having conducted a pilot study, the author believes that the questions clearly addressed the research questions and they were easy for the participants to understand. What is more, any answers from the participants that did not make sense were detected thanks to the design of the questionnaires, and were removed from the study.

7.11 Acknowledgments

The evaluation procedure applied has been reviewed by several experts in teaching and learning from the Centre for Teaching and Learning at the University of Limerick. Experts in International Management and Organizational Behavior at the University of Limerick were also consulted as regards the design of the training scenarios. The participants' effort in the pilot study and the field study must also be acknowledged. This work was supported, in part, by Science Foundation Ireland grant 10/CE/I1855 to Lero - the Irish Software Engineering Research Centre (www.lero.ie).

Chapter 8

Conclusions

This chapter presents a summary of the results of this research and analyzes the achievement of the research goals proposed. It also outlines future lines of research and summarizes the resulting publications.

8.1 Analysis of research goals

In this thesis I have shown how the VENTURE framework is designed, developed and evaluated. This research has followed a mixed method methodology involving the participation of students, practitioners and researchers under realistic training conditions. The research conducted in this thesis was guided by an iterative and incremental approach through which the framework was iteratively evaluated and improved in order to address the research questions. Two Expert Evaluations and a Field Study were conducted, and the results indicate that the framework achieves the GSD training objectives.

The hypothesis and objectives of this thesis were presented in Chapter 1, and these goals are now analyzed in this section in order to evaluate whether they have been achieved. The first objective was to:

1. **Define a simulation-based framework with which to provide training on GSD interactions**

The first objective was addressed as follows:

- **A study of the state-of-the-art in GSD in order to extract the problems that participants and organizations have to confront in these environments**. This study is presented in Section 2.1, in which a Systematic Literature Review on GSD problems and solutions is described.
- **A study of the state-of-the-art as regards the concrete methods, techniques and tools applied to provide training in GSD**. This study is presented in Section 2.2, in which a Systematic Literature Review of GSD Education is presented. Learning environments, simulation approaches and methods for teaching and training GSD in universities and companies are discussed.
- **A study of the concrete interaction problems and on the knowledge and skills that students and practitioners need to acquire in order to efficiently interact in GSD settings**. Section 2.3 presents a study of the problems, and the knowledge and skills that are required in GSD. Section 2.4 complements the study by reviewing the environments presented in literature that provide training on the communication and interaction problems in GSD.
- **The design of an extensible framework that addresses the problems found in literature that will permit the training of GSD skills that are difficult to address with traditional training methods**. A simulation based framework for GSD training is presented in Section 4, based on the results of the above.

The second objective of this thesis was to:

2. **Develop the software required to support the training framework. This software must provide instant feedback, automatic assessment and be adaptable to different kinds of GSD scenarios.**

This second goal was addressed as follows:

- **Development of a simulator that can provide training on GSD interaction by using Virtual Agents who play a role in the simulation. Students will interact with these Virtual Agents in order to train as regards specific GSD situations and problems**. A framework based on the definition of these simulations is defined in Section 4. Section 6 describes an implementation of a chat simulator and an email simulator in the context of the framework.
- **Development of a designer with which to easily define these simulations**. In order to provide a mechanism that can be used to elaborate training tasks adapted to specific needs and focused on real world problems, a scenario designer is considered in the framework as described in Section 4. Section 6 describes an implementation of the scenario designer defined in the framework.
- **Integration of an automated assessment method that may minimize instructors' efforts and provide students with additional feedback**. An assessment method able to track the students and detect the areas on which they should focus is integrated into the framework described in Section 4. Section 6 describes an implementation of the assessment method defined in the framework.
- **Development of training simulations that could serve to prove the effectiveness of the framework as regards providing GSD training**. Two training simulations designed to provide immediate feedback were developed in order to validate the framework, as detailed in Chapter 7.
- **The evaluation of the framework on a course that will check whether this framework is effective**. Two Expert Evaluations were conducted in Chapter 5 with the objective of improving the framework. Chapter 7 presents the evaluation of the framework (Field Study) conducted on a course made up of two training scenarios. This evaluation checked that the framework can promote the awareness and learning of cultural linguistic and GSD related problems while improving communication and coordination skills.

The third objective of this thesis was to:

3. **Create a mechanism that allows real world scenarios to be gathered, which can be applied with training purposes.**

The third objective was addressed by:

- **A study of the state-of-the-art of GSD patterns that occur during interactions**. A literature review on GSD patterns was conducted in order to complete Chapter 4. This literature review is detailed in Appendix II.
- **Definition and implementation of a collaborative tool that would serve to gather GSD patterns and scenarios and share them with the GSD community**. A GSD collaborative knowledgebase based on GSD patterns and scenarios is integrated into the framework described in Section 4. Section 6 describes an implementation of the GSD community site defined in the framework.

Based on the attainment and fulfillment of the objectives by conducting the actions detailed above, the results of this research support the initial thesis hypothesis: that it is feasible to provide effective training on GSD interactions by reproducing realistic settings and scenarios that occur in these environments.

8.2 Contribution of this thesis

Based on the goals of the work defined above, the main contributions of this thesis are the following:

1. A Framework and a set of tools for simulation-based GSD training. This includes an architecture that provides interaction training based on Virtual Agents, in addition to the mechanisms with which to design these simulated interactions. More specifically, the framework and tools supporting the framework provide:

a. The ability to promote interaction practices by simulating GSD settings.
b. The ability to promote cultural awareness, showing the customs and behaviors of different cultures.
c. The ability to show linguistic problems and misunderstandings that may occur during multicultural interactions.
d. The ability to rigorously improve attitudes and reactions as regards real world problems that occur in GSD.
e. The ability to provide immediate feedback and guidance.
f. The capacity to register the interactions with evaluation purposes and provide the automatic assessment of GSD skills.
g. Monitoring and communication mechanisms with which to conduct the courses online.

2. A GSD pattern model that can be used to acquire real world problems in GSD that can then be used for pedagogical purposes. This model is intended to elicit knowledge by means of the collaboration of the GSD community and provide mechanisms with which to:

   a. Define GSD patterns and the scenarios in which these patterns may be contextualized.
   b. Share, review and improve the patterns and scenarios in collaboration with the GSD community.

Finally, the study of the state-of-the-art can also be considered as a contribution on itself, as two systematic literature reviews were conducted in the fields of GSD [232] and GSD education [373] which currently have around 85 and 17 citations, respectively.

8.3 Potential applicability

This framework makes a contribution as regards both practitioners and researchers by tackling several points which may be interesting in the following fields of research and practice:

- Global Software Development, collaboration and coordination of practices.
- Global virtual team research: communication, culture and language.
- Computing education research, education technology, distance learning and e-Learning.

The transferability of the conceptual and methodological findings to domains and field settings beyond that of Global Software Development education remains to be proven. Nonetheless, the work needed to adapt this framework to other fields has already been addressed by conducting a commercial feasibility study in Ireland with the support of Enterprise Ireland. This study has resulted in positive conclusions as regards a wider applicability of the framework to interaction training, and not only in the field of Global Software Development. Several fields of endeavor in the sphere of research and practice can be derived from the results of this thesis. Table 64 summarizes the most relevant domains of applicability according to the literature in the related research field that might to some extent be related to GSD training.

Table 64. Domains of applicability

Domain	Applicability	References
Technology-use mediation and technology facilitation	The study of interactions of different stakeholders through a variety of technologies.	[374], [375]
Global collaboration in tertiary education	Exposing students to real-world projects in order to encourage them to share, learn, and improve when collaborating.	[276], [376]
Management and coordination of distributed development or outsourced activities	Management and coordination activities may be the object of simulation through interaction activities.	[377], [378]
Open source software development teams and communities	Interaction in open source must frequently follow certain rules if security and privacy are to be provided.	[379], [380]
Computer Supported Collaborative Work, Computer Supported Collaborative Learning	Efficient communication, coordination, and process-aware collaboration are frequently studied in the field of virtual teams and teamwork.	[381], [382]
Group facilitators working in virtual and distributed team contexts	There is a strong interest in developing practical processes and techniques that facilitators could use in building and maintaining relationships in the online groups they work with.	[383]
Computing Education Research, Education Technology, Distance Learning and e-Learning	Technologies and methods used in distance learning or computing education.	[384], [385], [386]
Support for online communities	Support for online communities can be based on the contributions of this thesis.	[387], [388]

8.4 Future work

The framework is currently being applied at a Spanish university in a subject concerning Virtual Teams in order to provide GSD training. The same scenarios that have been used in the Field Study are being used on this course. During this course it has been detected that in order to improve the applicability of the tool in future courses, load tests should be conducted to determine the optimal number of concurrent participants supported by the tool. Efforts should be made to define the best approach with which to escalate the system in order to be able to provide a varying number of students with the service. The future objective is to apply the framework on this kind of courses on a regular basis, and this experience is therefore essential if the maturity required of the framework is to be achieved. Future work will also be focused on applying the framework at universities in other countries and on studying the training requirements and specific needs of these universities.

A Commercial Feasibility Study has been conducted in order to explore the commercial options for VENTURE. This study has involved meetings with three multinational companies that have expertise in virtual team management, training and cultural and linguistic diversity. These meetings have served to check that the problems that VENTURE is tackling are in fact real problems of these kinds of companies. Although the companies have not provided concrete figures about economic costs, as they were confidential, they have acknowledge that they are investing in order to tackle these problems by different means. Cultural issues are one of the most conflictive points for companies, and may lead to an important amount of rework, and it has indeed been recognized that it is difficult to train or promote awareness of these issues. This study has served not only to obtain feedback from the commercial point of view but also to provide insights into the future work that, according to the experts, would improve the VENTURE framework from a commercial perspective.

After conducting the Commercial Feasibility Study, our immediate future work will be focused on conducting the subsequent commercialization activities with the support of Enterprise Ireland (the governmental organization responsible for the development and growth of Irish enterprises in world markets), for which an application is now under develpment. Future work may consider applicability to other fields related to GSD training, as presented in the previous section, but also more diverse fields such as:

- Customer assistance training: Based on the procedures and practices of the company.
- Job interview training: Training interviewees how to behave or how to react to typical questions that they might need to answer during a job interview.
- Patient assistance training: How to deal with patients in clinics, hospitals or psychiatrists' surgeries.
- Training for children with communication problems or as a treatment for people with diseases such as autism.
- Emergency training: Providing the best response to critical problems based on real scenarios (police, ambulance or firefighters called to assist at emergencies).

This work has also presented two Expert Evaluations and a Field Study in which weak points and ideas for improvement have been detected. Our future work will be focused on providing answers to these points. The following subsections detail the set of tasks that have been planned in order to extend this research work in the future. This work will be principally focused on the simulations, the scenario designer, the GSD patterns and scenarios and the framework's assessment method.

8.4.1 Simulations

Both the Expert Evaluations (1 and 2) and the final evaluation (Field Study) have identified a number of ideas that could improve the simulations conducted in VENTURE. In this respect, the primary technical improvements were those shown as follows:

- **Simulations interface:**
  - Voice recognition was suggested as a possible enhancement to allow telephonic interaction training.
  - The avatars should be more realistic and should, for example include facial features from the nationality that they represent and use voices that represent their characteristic accents.
  - Users should be aware of their scores immediately after making each mistake rather than only being shown them at the end.

- **Simulation engine:**
  - Artificial intelligence: the Virtual Agents should be able to understand participants' interactions in a more flexible and accurate manner.
  - The possibility of the students to get stuck during the interactions should be minimized. The detection and provision of more accurate feedback when this problem occurs should be automated to avoid confusion. The functionality of skipping to the next stage of the interaction when necessary should provide the students with feedback as to what kind of interaction was expected from them.
  - Automate the detection of potential problems in the simulations. In the same way that a report is generated to assess the students, automatic reports should also inform the scenario designers about potential problems in the simulations enacted by the students. For example, the number of times that the students got stuck or had to skip a stage of the conversation can be reported, in addition to when the Virtual Guide had to intervene too often or repeat the same advice more than once.
  - Possibility of simulating interactions with more than one Virtual Colleague at the same time in order to provide training in realistic stressful situations.

- **Integration:**
  - Integration with existing communication tools used by companies, such as Lync (for instant messaging) or Outlook (e-mail). The idea would be to provide practitioners with instant feedback in their day-to-day

work when interacting under specific settings. For example, a chat interaction could be guided by a Virtual Guide by considering the specific location, role of the participants and purpose of the communication.

- Integration with development tools such as Eclipse or Visual Studio could be considered in order to simulate real interactions between developers by using the mechanisms that these tools provide, such as workitems. A workitem is a means of keeping track of the tasks, issues, bugs, requirements, etc. or other information items that a software team needs to address during the development cycle. Their creation, assignment and the interactions required to solve them could also be simulated or guided by Virtual Agents by using VENTURE.

Most of these ideas came into being during the validation of the chat simulator. However, the improvement and validation of the email simulator is also an important future work that should follow the same process conducted with the chat simulator, and which will eventually generate more ideas for future improvements.

8.4.2 Scenario designer

Although the scenario designer was not tested in the Field Study, the Expert Evaluations led to the following recommendations for improvements:

- **Usability and operational improvements:**
  - The designer interface must be improved by simplifying the design as much as possible and improving its usability in order to make it available to non-technical designers.
  - Possibility of choosing different paths of the simulation depending on the user's responses. It would be helpful to make users aware of the consequences of their actions.
  - Ability to test the specific stage of the conversation that is being designed. For example, when designing a chat simulation, the chat simulator could be executed only for that stage in order to test that part of the interaction by avoiding executing the entire simulation.
  - The designer should provide automatic guidance and suggestions based on the GSD patterns and scenarios contained in the knowledgebase. The effort of creating new scenarios should be minimized by facilitating the use of existing knowledge.
  - Rules should consider exceptions that will permit the use of certain expressions in certain contexts. For example, a cultural rule can detect that the interaction "I need the document tomorrow" is direct because it contains the keyword "tomorrow". However an interaction like "we can delay it until tomorrow" should be accepted as appropriate as it is more indirect. This kind of exceptions should therefore be taken into consideration in the design.

- **Content related improvements:**
  - A representative number of training scenarios and GSD rules should be developed to cover a broader area of knowledge. One of the participants interviewed considered it useful for training cultures such as Koreans interacting with Japanese people. Interactions between these cultures must consider political conflicts and thus follow certain rules to avoid problems during the interactions that could be trained.
  - The interaction policies of companies and the rules used to accomplish regulations could likewise be taken into account when designing simulations.
  - In the same way that cultural differences can be implemented in the simulations, the different kinds of personalities involved in an interaction could also be considered. Practitioners can interact with a number of difficult personalities such as people who are hostile-aggressive, complainers, silent and unresponsive super-agreeable, negativists, know-it-alls and indecisive. Bramson [389] presents techniques on how to cope with these difficult personalities, showing real world examples. These kinds of coping strategies could be designed in a future work. This would allow students to identify these

personalities and acquire techniques and capabilities with which to resolve conflicts and promote effective interactions in these situations.

- It is a real need for companies to provide training not only as regards interacting in English. Moreover, the Virtual Agents that simulate that they are people from specific countries should be able to respond in their supposed native languages. For example, during the interactions with Mexican Virtual Agents, some participants initiated the conversation by saying “hola” (“hello” in Spanish), which the Virtual Agent did not understand.

8.4.3 GSD patterns and scenarios

With regard to the GSD patterns and scenarios required to design accurate training scenarios, one important limitation as regards obtaining a representative knowledge base is the community’s willingness to participate and share experiences. Future work will consider this problem by gathering together training scenarios by conducting confidential interviews with practitioners, and also through the use of role play. Role play was found to be a successful method for identifying scenarios and patterns in Coplien and Harrison [336].

The next step in relation to this work is to create a community of researchers and practitioners that is interested in sharing their knowledge, or wishes access to the information. This will be achieved by inviting companies and the authors of relevant papers in GSD literature to participate in this knowledge-sharing.

Efforts will be made to extract new problems and scenarios and contextualize them in the GSD field, making them available to the community. Finally, the structure of the repository should be validated by GSD experts and the resulting knowledge should be applied in academia and industry. Future work will focus on defining critical success factors that are considered essential in the assessment of the patterns and scenarios. The feasibility of applying an adapted version of the Q-PAM evaluation technique [390], should be studied in order to analyze its feasibility. The general idea of Q-PAM is to use scenarios as test cases which are analyzed against the patterns. These patterns and scenarios can also be validated by proving their effectiveness when they are applied in training environments to develop specific skills.

8.4.4 Assessment

Future work concerning the application of VENTURE in universities and companies will serve to refine the assessment process. The lessons learned will be used to increase the automation of the assessment without compromising the validity of the process as a whole.

Issues such as ensuring students’ engagement and improving the learning outcomes may be key success factors in the eventual effectiveness of the application. Future work will therefore, as suggested by some participants, be focused on improving the final assessment reports, including additional tutorials that could serve the students as regards reinforcing those points with which they had more problems.

In a long-term future when a sufficient number of scenarios might be available, the assessment should also be able to detect which is the most suitable training scenario for each student based on their background and performance.

8.5 Output from the research program

The results of this thesis are described in the following sub-sections.

8.5.1 Invention disclosure

An invention disclosure of the VENTURE tool was achieved at the University of Limerick. This was a previous step required for the commercialization objectives that are currently underway and that intend to adapt VENTURE to the potential market in order for it to be commercialized. The invention disclosure is referenced as follows:

M. J. Monasor, A. Vizcaíno, M. Piattini, J. Noll and S. Beecham. VENTURE (a simulated e-learning training platform) – Invention Disclosure, UL ref: 2006190. University of Limerick.

8.5.2 Commercialization

A Commercial Case Feasibility Support Grant conceded by Enterprise Ireland served to contract the independent consultants who conducted interviews at three multinational companies. These interviews served to obtain insights into the commercialization potential of the tool and to define a business model adapted to the needs of the potential customers. The reference of this grant is:

Commercial Case Feasibility Support Grant. Ref: CF 2013 3752Y, Enterprise Ireland, 2013.

The results of this study were positive, and a Commercialization Fund Project Support has now been applied for through Enterprise Ireland.

8.5.3 Publications

The results of the research performed in this thesis have been published in various peer-reviewed forums. The list of publications to which the author has contributed is listed below. Table 65 shows each paper's contribution to the various research subjects covered in this thesis. It divides the publications into various categories: Indexed international journals (JCR), international journals (IT); international conferences (IC); book chapters (CHP); Latin-American conferences (LAC) and technical reviews (TR). Submitted work is also mentioned (SUB).

Indexed International Journals

JCR-1. [391] **Monasor, M.J.**, A. Vizcaíno, and M. Piattini, Cultural and linguistic problems in GSD: a simulator to train engineers in these issues. Journal of Software Maintenance and Evolution: Research and Practice (Special Issue on Global Software Engineering), 2011. 24(6): p. 707-717. ISSN/ISBN: 2047-7481
Impact factor JCR (2011): 1.273

International Journals

IJ-1. [232] **Monasor, M.J.**, M. Piattini, and A. Vizcaíno, Challenges and Improvements in Distributed Software Development: A Systematic Review. Advances in Software Engineering, 2009: p. 1-16. ISSN/ISBN: 1687-8655

IJ-2. [392] **Monasor, M.J.**, A. Vizcaíno, and M. Piattini, Improving Distributed Software Development in Small and Medium Enterprises. The Open Software Engineering Journal, 2009: p. 26-37. ISSN/ISBN: 1874-107X/10

International Conferences

IC-1. [393] **Monasor, M.J.**, A. Vizcaíno, M. Piattini, J. Noll, and S. Beecham, Simulating Global Software Development processes for use in Education: A Feasibility Study, in 20th European Conference, EuroSPI, F. McCaffery, R.V. O´Connor, and R. Messnarz, Editors. 2013, Springer: Dundalk, Ireland. p. 36-47. ISSN/ISBN: 978-3-642-39178-1

IC-2. [394] **Monasor, M.J.**, A. Vizcaíno, and M. Piattini, Providing Training in GSD by Using a Virtual Environment, in International Conference on Product-Focused Software Development and Process Improvement (PROFES). 2012: Madrid, Spain. p. 203-217. ISSN/ISBN: 978-3-642-31062-1

IC-3. [395] **Monasor, M.J.**, A. Vizcaíno, and M. Piattini, VENTURE: Towards a framework for simulating GSD in educational environments, in 5th International Conference on Research Challenges in Information Science (RCIS). 2011: Gosier, Guadeloupe, France. p. 469 - 478. ISSN/ISBN: 978-1-4577-1839-7

IC-4. [396] **Monasor, M.J.**, A. Vizcaino, and M. Piattini, A Framework for Defining Simulators with Which to Train Global Software Development, in International Conference on Software and Data Technologies (ICSOFT). 2011: Seville, Spain. p. 261-264. ISSN/ISBN: 978-1-4577-1839-7

IC-5. [397] **Monasor, M.J.**, A. Vizcaíno, and M. Piattini. Training Global Software Development skills through a simulated environment. in 5th International Conference on Software and Data Technologies (ICSOFT). 2010. Athens, Greece. p. 271-274.

IC-6. [398] Garrido, P.J., A. Vizcaíno, J. Andrada, **M.J. Monasor**, and M. Piattini, A Tool for Decisions Management in Distributed Software Projects, in 6th REMIDI Workshop in the **International Conference on Global Software Engineering Workshops (ICGSE)**. 2012: Porto Alegre (Brazil). p. 22-27. ISSN/ISBN: 978-1-4673-2625-4

IC-7. [373] **Monasor, M.J.**, A. Vizcaíno, M. Piattini, and I. Caballero, Preparing students and engineers for Global Software Development: A Systematic Review, in **International Conference on Global Software Development (ICGSE)**. 2010, IEEE Computer Society: Princeton, NJ, USA. p. 177-186. ISSN/ISBN: 978-0-7695-4122-8/10

IC-8. [399] **Monasor, M.J.**, A. Vizcaino, and M. Piattini, An Architecture for Creating Simulators for Training Global Software Development, in **International Conference on Global Software Development (ICGSE)**. 2011: Helsinki, Finland. p. 90 - 94. ISSN/ISBN: 978-1-4577-1839-7

IC-9. [400] **Monasor, M.J.**, A. Vizcaíno, and M. Piattini, A Framework for Training Skills for Global Software Development, in **International Conference on Global Software Development (ICGSE)**. 2010: Princeton, NJ, USA. p. 355-356. ISSN/ISBN: 978-0-7695-4122-8/10

IC-10. [401] **Monasor, M.J.**, A. Vizcaíno, and M. Piattini. An educational environment for learning Global Software Development. in 10th IEEE International Conference on Advanced Learning Technologies (ICALT). 2010. Sousse, Tunisia. p. 99 - 101. ISSN/ISBN: 978-1-4244-7144-7

IC-11. [402] **Monasor, M.J.**, A. Vizcaíno, and M. Piattini. A training tool for Global Software Development. in 9th International Conference on Information Technology Based Higher Education and Training (ITHET). 2010. Cappadocia, Turkey. p. 9-16. ISSN/ISBN: 978-1-4244-4811-1

IC-12. [403] **Monasor, M.J.**, et al., Towards a Global Software Development Community Web: Identifying Patterns and Scenarios, in PARIS Workshop, **International Conference on Global Software Development (ICGSE)**. 2013: Bari, Italy. p. 41-46. ISSN/ISBN: 978-0-7695-5055-8

International Book Chapters

CHP-1. [404] **Monasor, M.J.**, A. Vizcaino, and M. Piattini, Preparing Students and Engineers for Global Software Development: An Architecture Based on Simulation, in Encyclopedia of Software Engineering. 2012. p. 117-186.

CHP-2. [405] **Monasor, M.J.**, M. Piattini, and A. Vizcaíno, A Systematic Review of Distributed Software Development: Problems and Solutions, in Handbook of Research on Software Engineering and Productivity Technologies: Implications of Globalization. 2010, IGI Global. p. 209-225. ISSN/ISBN: 9781605667317

CHP-3. [406] **Monasor, M.J.**, A. Vizcaíno, and M. Piattini, A Tool for Training Students and Engineers in Global Software Development Practices, in Collaboration and Technology, G. Kolfschoten, T. Herrmann, and S. Lukosch, Editors. 2010, Springer Berlin Heidelberg. p. 169-184. ISSN/ISBN: 978-3-642-15713-4

CHP-4. [407] **Monasor, M.J.**, and M. Piattini, Problems and Solutions in Distributed Software Development: A Systematic Review, in Software Engineering Approaches for Offshore and Outsourced Development, K. Berkling, et al., Editors. 2009, Springer Berlin Heidelberg. p. 107-125. ISSN/ISBN: 978-3-642-01855-8

Latin American Conferences

LAC-1. [408] **Monasor, M.J.**, A. Vizcaíno, and M. Piattini, Docencia en Desarrollo Global de Software: Una revisión Sistemática, in XVII Jornadas de Enseñanza Universitaria de Informática. 2011: Seville, Spain. p. 221-230. ISSN/ISBN: 978-84-615-8119-1

LAC-2. [409] **Monasor, M.J.**, A. Vizcaino, and M. Piattini, VENTURE: Hacia un entorno para el entrenamiento del Desarrollo Global de Software, in XVI Jornadas de Ingeniería del Software y Bases de Datos (JISBD). 2011: A Coruña (Spain). p. 1065-1070. ISSN/ISBN: 978-84-9749-486-1

Technical Reports

TR-1. [410] **Monasor, M.J.**, A. Vizcaíno, M. Piattini, J. Noll, and S. Beecham: Evaluation of a simulation-based platform for training Global Software Development. Lero-TR-2013-02. Lero, The Irish Software Engineering Research Centre. University of Limerick, Limerick (2013)

Table 65. Relationship between the publications and research issues of the PhD thesis

Reference	Global Software Development	Global Software Development Education	VENTURE	GSD Community
JCR-1			✓	
IJ-1	✓			
IJ-2	✓			
IC-1			✓	
IC-2			✓	
IC-3			✓	
IC-4			✓	
IC-5			✓	
IC-6	✓			
IC-7		✓		
IC-8			✓	
IC-9			✓	
IC-10			✓	
IC-11			✓	
IC-12				✓
CHP-1			✓	
CHP-2	✓			
CHP-3	✓			
CHP-4			✓	
LAC-1		✓		
LAC-2			✓	
TR-1			✓	

8.5.4 Citations

This section details the citations of the author's work as detailed in Table 66, which presents the citations reported by Google Scholars on 21/01/2014. Citations from the author, his supervisors and collaborators have been excluded, resulting in a total of 110 citations.

Table 66. Citations of author's work

IJ-1. [232] Monasor, M.J., M. Piattini, and A. Vizcaíno, Challenges and Improvements in Distributed Software Development: A Systematic Review. Advances in Software Engineering, 2009: p. 1-16.	Citations: 79

1. Omoronyia, I., J. Ferguson, M. Roper, and M. Wood, A review of awareness in distributed collaborative software engineering. Software: Practice and Experience, 2012. 40(12): p. 1107-1133.
2. Steinmacher, I., A.P. Chaves, and M.A. Gerosa, Awareness support in global software development: a systematic review based on the 3C collaboration model, in Proceedings of the 16th international conference on Collaboration and technology. 2010, Springer-Verlag: Maastricht, The Netherlands.
3. Fernández-Luna, J.M., J.F. Huete, R. Pérez-Vázquez, and J.C. Rodríguez-Cano, CIRLab: A groupware framework for collaborative information retrieval research. Information Processing & Management, 2009. 46(6): p. 749-761.
4. Mishra, D. and A. Mishra, A review of non-technical issues in global software development. International Journal of Computer Applications in Technology, 2011. 40(3): p. 216-224.
5. Bannerman, P.L., E. Hossain, and R. Jeffery. Scrum practice mitigation of global software development coordination challenges: A distinctive advantage? in System Science (HICSS), 45th Hawaii International Conference on. 2012: IEEE.
6. Hossain, E., P.L. Bannerman, and D.R. Jeffery, Scrum practices in global software development: A research framework, in Product-focused software process improvement. 2011, Springer. p. 88-102.
7. Keshlaf, A.A. and S. Riddle, Risk Management for Web and Distributed Software Development Projects, Fifth International Conference on Internet Monitoring and Protection. 2010, IEEE Computer Society. p. 22-28.
8. Mishra, D. and A. Mishra, Distributed Information System Development: Review of Some Management Issues, in On the Move to Meaningful Internet Systems: OTM Workshops, Springer-Verlag, Editor. 2009: Berlin.
9. Fauzi, S.S.M., P.L. Bannerman, and M. Staples. Software Configuration Management in Global Software Development: A Systematic Map. in 17th Asia Pacific Software Engineering Conference. 2010. Sydney, Australia.
10. Vivian, R.L., E.H.M. Huzita, G.C.L. Leal, and A.P.C. Steinmacher, Context-awareness on software artifacts in distributed software development: a systematic review, in Proceedings of the 17th international conference on Collaboration and technology. 2011, Springer-Verlag: Paraty, Brazil.
11. Chaves, A.P., I. Steinmacher, L. Leal, E.H. Huzita, and A.B. Biasão, OntoDiSENv1: an ontology to support global software development. CLEI Electronic Journal, 2011. 14(2): p. 2-2.
12. Ciccozzi, F. and I. Crnković. Performing a project in a Distributed Software Development Course: Lessons Learned. in International Conference on Global Software Development (ICGSE). 2010. Princeton, NJ, USA.
13. Hossain, E., P.L. Bannerman, and R. Jeffery, Towards an understanding of tailoring scrum in global software development: a multi-case study, in Proceedings of the 2011 International Conference on on Software and Systems Process. 2011, ACM: Waikiki, Honolulu, HI, USA. p. 110-119.
14. Tell, P. and M.A. Babar, Activity theory applied to global software engineering: Theoretical foundations and implications for tool builders, in Global Software Engineering (ICGSE), IEEE Seventh International Conference on. 2012, IEEE. p. 21-30.
15. Tell, P. and M.A. Babar, Requirements for an infrastructure to support Activity-Based Computing in Global Software Development, in International Conference on Global Software Engineering Workshops (ICGSE). 2011: Helsinki, Findland. p. 62-69.

16. Almeida, L.H., P.R. Pinheiro, and A.B. Albuquerque, Applying multi-criteria decision analysis to global software development with scrum project planning, in Proceedings of the 6th international conference on Rough sets and knowledge technology. 2011, Springer-Verlag: Banff, Canada.
17. Colomo-Palacios, R., P. Soto-Acosta, G.-P.F. J., and Á. García-Crespo, A Study of the Impact of Global Software Development in Packaged Software Release Planning. Journal of Universal Computer Science, 2012. 18(19): p. 2646-2668.
18. Almeida, L.H., A.B. Albuquerque, and P. Rogério Pinheiro, A Multi-criteria Model for Planning and Fine-tuning Distributed Scrum Projects, in Global Software Engineering (ICGSE), 6th IEEE International Conference on. 2011, IEEE. p. 75-83.
19. Khan, M. and M. Tariq, The Context of Global Software Development: Challenges, Best Practices and Benefits. Information Management & Business Review, 2011. 3(4): p. 193-197.
20. Leal, G.C.L., C.A.d. Silva, E.H.M. Huzita, T.F.C. Tait, and P.R. Maringá, Recomendaç oes para a Gestao do Desenvolvimento de Software com Equipes Distribuıdas. 2010.
21. Pesola, J.-P., H. Tanner, J. Eskeli, P. Parviainen, and D. Bendas, Integrating Early V&V Support to a GSE Tool Integration Platform, in Global Software Engineering Workshop (ICGSEW), Sixth IEEE International Conference on. 2011, IEEE. p. 95-101.
22. Dullemond, K., B. van Gameren, and R. van Solingen, Collaboration should become a first-class citizen in support environments for software engineers, in Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 8th International Conference on. 2012, IEEE. p. 398-405.
23. Lai, H., R. Peng, D. Sun, F. Shao, Y. Liu, and Y. Ni, A systematic review of Re-specific wikis for distributed requirements engineering. Wuhan University, Wuhan, China, Tech. Rep, 2012. 2012530.
24. Steinmacher, I., A.P. Chaves, and M.A. Gerosa, Awareness Support in Distributed Software Development: A Systematic Review and Mapping of the Literature. Comput. Supported Coop. Work, 2013. 22(2-3): p. 113-158.
25. Tell, P. and M. Ali Babar, Supporting activity based computing paradigm in global software development, 26th IEEE/ACM International Conference on Automated Software Engineering. 2011, IEEE Computer Society. p. 508-511.
26. Kroll, J., E.R. Hess, J.L.N. Audy, and R. Prikladnicki, Researching into Follow-the-Sun Software Development: Challenges and Opportunities, in International Conference on Global Software Engineering (ICGSE). 2011: Helsinki, Findland. p. 60-65.
27. Bendix, L., J. Magnusson, and C. Pendleton, Configuration Management Stories from the Distributed Software Development Trenches, in Global Software Engineering (ICGSE), IEEE Seventh International Conference on. 2012, IEEE. p. 51-55.
28. Lai, H., R. Peng, D. Sun, F. Shao, and Y. Liu, A Survey of RE-specific Wikis for Distributed Requirements Engineering, in Semantics, Knowledge and Grids (SKG), Eighth International Conference on. 2012. p. 47-55.
29. Keshlaf, A. and S. Riddle, Web and Distributed Software Development Risks Management: WeDRisk Approach. International Journal on Advances in Software, 2011. 3(3 and 4): p. 447-460.
30. Tihinen, M., P.i. Parviainen, R. Kommeren, and J. Rotherham, Metrics in Distributed Product Development, in ICSEA, The Sixth International Conference on Software Engineering Advances. 2011. p. 275-280.
31. Silva, F.Q., R. Prikladnicki, A.C.C. França, C.V. Monteiro, C. Costa, and R. Rocha, An evidence-based model of distributed software development project management: results from a systematic mapping study. Journal of Software: Evolution and Process, 2012. 24(6): p. 625-642.
32. Marques, A.B., R. Rodrigues, and T. Conte. Systematic literature reviews in distributed software development: A tertiary study. in Global Software Engineering (ICGSE), IEEE Seventh International Conference on. 2012: IEEE. p. 134-143.
33. Pehmöller, A., F.S. Capgemini, and S. Wagner, Patterns for testing in global software development, in 13th International Conference on Quality Engineering in Software Technology. 2010.

34. Mishra, D. and A. Mishra, Development: Evaluating the Past to Envision the Future. Journal of Global Information Technology Management, 2011. 14(4): p. 48-69.
35. Bendix, L., J. Magnusson, and C. Pendleton, Configuration Management Support for Distributed Software Development. 2012.
36. Orsoletta, R.A.D. Simulated co-location in distributed software development: An experience report. in Global Software Engineering Workshops (ICGSEW), IEEE Seventh International Conference on. 2012: IEEE.
37. Colomo-Palacios, R., P. Soto-Acosta, A. Mishra, and Ã. García-Crespo, Software quality management improvement through mentoring: an exploratory study from GSD projects, in On the Move to Meaningful Internet Systems: OTM Workshops. 2011, Springer. p. 190-199.
38. Qahtani, A., A. Gravell, and G. Wills, Toward a framework for software development across organisational boundaries, in PhD Symposium in 13th International Conference on Agile Software Development (XP2012). 2012: Malmo, Sweden.
39. Pehmöller, A., F. Salger, and S. Wagner, Testing in Global Software Development - A Pattern Approach, in 5th World Congress for Software Quality. 2011: Shanghai, China.
40. Junior, I.H.d.F., C.M.d.O. Rodrigues, R.R.d. Azevedo, H.P.d. Moura, E.R.G. Dantas, and F. Freitas, Best Practices to Enhance the Communication in Distributed Software Development Environments, in International Conference on Research and Practical Issues of Enterprise Information Systems (CONFENIS). 2010.
41. Pierce, K., C. Ingram, B. Bos, and A. Ribeiro, Experience in Managing Requirements Between Distributed Parties in a Research Project Context, in Global Software Engineering (ICGSE), IEEE 8th International Conference on. 2013, IEEE. p. 124-128.
42. Nieuwenhuis, R., Virtual Extreme Programming Workbench: a support tool for practitioners of extreme programming in a distributed environment, in 28th Annual Pacific Northwest Software Quality Conference. 2009: Portland, Oregon, USA.
43. Hande, A., S. Suralkar, and P. Chawan, Distributed Software Project Development. International Journal of Engineering Research and Applications (IJERA) 2012. 2(3): p. 998-1003.
44. Qahtani, A., G. Wills, and A. Gravell, Toward a Framework for Localisation of Product Software across Organisational Boundaries, in Electronics and Computer Science. 2012, University of Southampton. p. 26.
45. Ruano-Mayoral, M., R. Colomo-Palacios, J.M. Fernández-González, and Á. García-Crespo. Towards a framework for work package allocation for GSD. in On the Move to Meaningful Internet Systems: OTM Workshops. 2011: Springer. p. 200-207.
46. Xia, H., M. Dawande, and V. Mookerjee, Optimal Coordination in Distributed Software Development, in Conference on Information Systems and Technologies, CIST. 2013: Minneapolis, Minnesota.
47. Izquierdo Cortázar, D., Global and Geographically Distributed Work Teams: Understanding the Bug Fixing Process and Bug-prone Activity Patterns. 2012, Universidad Rey Juan Carlos: Madrid.
48. Silva, J.T.d., M.A.l. Gerosa, I. Wiese, R. Ré, and I. Steinmacher, An Extensible Service for Experts Recommendation on Distributed Software Development Projects, in Global Software Engineering Workshops (ICGSEW), IEEE Seventh International Conference on. 2012, IEEE. p. 18-21.
49. Tihinen, M., P. Parviainen, R. Kommeren, and J. Rotherham, Metrics and Measurements in Global Software Development. International Journal On Advances in Software, 2012. 5(3 and 4): p. 278-292.
50. Bala, R., V. Krishnan, and W. Zhu, Distributed Development and Product Line Decisions. Production and Operations Management, 2013.
51. Leal, G.C.L., A.P. Chaves, E.H.M. Huzita, and M.E. Delamaro, An Integrated Approach of Software Development and Test Processes to Distributed Teams. J. UCS, 2012. 18(19): p. 2686-2705.
52. Parviainen, P. and M. Tihinen, Knowledge-related challenges and solutions in GSD. Expert Systems, The Journal of Knowledge Engineering, 2011.
53. Marzanah, A., A. Salfarina, A. Abdul Azim, and A. Rusli, Barriers sorrounding knowledge transfer in non-collocated software architecture development. IJASCSE 2012. 1(3).
54. Narayanan, V. and R. DeFillippi, The Influence of Strategic Context on Project Management Systems: A

Senior Management Perspective. Project Governance: Getting Investments Right, 2012: p. 1-45.
55. Mishra, D., A. Mishra, R. Colomo-Palacios, and C. Casado-Lumbreras, Global Software Development and Quality Management: A Systematic Review, in On the Move to Meaningful Internet Systems: OTM Workshops. 2013, Springer. p. 302-311.
56. Mishra, P., N. Roy, S. Jain, and J. Fernandez, A Risk Framework for New Country Compliance for Global Software Companies, in Global Software Engineering Workshops (ICGSEW), IEEE Seventh International Conference on. 2012, IEEE. p. 1-5.
57. Pamulapati, S. and D. Gaddipati, Follow-the-Sun Software Development: Controlled Experiment. 2011, Blekinge Institute of Technology.
58. Keshlaf, A. and S. Riddle, An Empirical Study of Web and Distributed Software Risks from Three Perspectives: Project, Process and Product, in ICIMP, The Sixth International Conference on Internet Monitoring and Protection. 2011. p. 38-44.
59. Sillaber, C. and R. Breu. The Impact of Knowledge Sharing Platforms in Distributed Requirements Engineering Scenarios: A Systematic Review. in The 8th International Conference on Knowledge Management in Organizations. 2014: Springer.
60. Hossain, M.E., Scrum Practice Mitigation of Coordination Challenges in Global Software Development Projects: An Empirical Study. 2011, The University of New South Wales.
61. ul Haq, S., M.N.A. Khan, and M. Tariq, Information Management and Business Review. Context, 2011. 3(4): p. 193-197.
62. Huzita, E.H.M., G.C.L. Leal, R. Balancieri, T. Tait, E. Cardoza, R.R.d.M. Penteado, and R.L. Vivian. Knowledge and Contextual Information Management in Global Software Development: Challenges and Perspectives. in Global Software Engineering Workshops (ICGSEW), IEEE Seventh International Conference on. 2012: IEEE. p. 43-48.
63. Nawaz, S., Knowledge capture systems in software maintenance projects. International Journal of Research in Commerce, IT & Management, 2013. 3(7).
64. Izquierdo-Cortazar, D., Global and Geographically Distributed Work Teams: Understanding the Bug Fixing Process and Potentially Bug-prone Activity Patterns, in Software Maintenance and Reengineering (CSMR), 16th European Conference on. 2012, IEEE. p. 505-508.
65. Tell, P., M.A. Babar, and J. Grundy, A Preliminary User Evaluation of an Infrastructure to Support Activity-Based Computing in Global Software Development (ABC4GSD), in Global Software Engineering (ICGSE), IEEE 8th International Conference on. 2013, IEEE. p. 100-109.
66. Costa, C. and L. Murta, Version Control in Distributed Software Development: a Systematic Mapping Study, in International Conference on Global Software Development (ICGSE). 2013: Bari, Italy. p. 90-99.
67. Löfstedt, T. and D. Jonsson, Att koordinera distribuerade IT-projekt med stöd av informations-och kommunikationsteknik: En fallstudie på IFS World Operations AB. 2012, Linköping.
68. dos Santos, A.C.C., C.C. Borges, D.E. Carneiro, and F.Q. da Silva. Estudo baseado em Evidências sobre Dificuldades, Fatores e Ferramentas no Gerenciamento da Comunicação em Projetos de Desenvolvimento Distribuído de Software. in Proceedings of 7th Experimental Software Engineering Latin American Workshop (ESELAW). 2010. p. 20.
69. Giraldo Velásquez, F.D., Á.M. Jiménez Rojas, S. Ochoa, and S. Zapata, Colaborato latinoamericano para la investigación en ingeniería de software experimental, soportado por técnicas CSCW-L implementadas sobre redes académicas de alta velocidad. e-colabora "Revista de ciencia, educación, innovación y cultura apoyadas por redes de tecnología avanzada", 2012. 2(3): p. 92-110.
70. Soares, P.H., Uma estratégia para tratar os aspectos sócio-culturais no desenvolvimento distribuído de software, in Centro de tecnología, departamento de informática. 2011, Universidade esdual de Maringá.
71. Misra, S., R. Colomo-Palacios, T. Pusatli, and P. Soto-Acosta, A discussion on the role of people in global software development. Tehnički vjesnik, 2013. 20(3): p. 525-531.
72. Galicia, R. and J. Yazmín, Diseño de un proceso para la administración de configuración de software. 2013, Centro Universitario Querétaro.

73. Ruano Mayoral, M., Marco para la asignación de paquetes de trabajo en entornos de desarrollo global de software. 2012, Universidad Carlos III de Madrid.
74. Degerman, G. and L. Engzell, Introducing Open Source ways of working. 2011, KTH/Maskinkonstruktion (Inst.).
75. Degerman, G. and L. Engzell, Introduktion av Open Source som arbetssätt. 2011, Uppsala University
76. Tamy Ishii, F., G.C. L Leal, E.V. Cardoza Galdamez, R. Balancieri, T.F. Calvi Tait, and E.H. Moriya Huzita. Knowledge management in distributed software development: a systematic review. in XVIII Congreso Argentino de Ciencias de la Computación. 2013. p. 857-866.
77. Qahtani, A.M., G.B. Wills, and A.M. Gravell, A Framework of Challenges and Key Factors for Applying Agile Methods for the Development and Customisation of Software Products in Distributed Projects. International Journal of Digital Society, 2013. 1(1): p. 766-773.
78. Verner, J.M., O.P. Brereton, B.A. Kitchenham, M. Turner, and M. Niazi, Risks and risk mitigation in global software development: A tertiary study. Information and Software Technology, 2014. 56(1): p. 54-78.
79. Lehtinen, T.O., R. Virtanen, J.O. Viljanen, M.V. Mäntylä, and C. Lassenius, A Tool Supporting Root Cause Analysis for Synchronous Retrospectives in Distributed Software Teams. Information and Software Technology, 2014. In press.

IC-7. [373] **Monasor, M.J.**, A. Vizcaíno, M. Piattini, and I. Caballero, Preparing students and engineers for Global Software Development: A Systematic Review, in International Conference on Global Software Development (ICGSE). 2010, IEEE Computer Society: Princeton, NJ, USA. p. 177-186.	Citations: 10

1. Deitersy, C., C. Herrmannz, R. Hildebrandtz, E. Knauss, M. Kuhrmannx, A. Rauschy, B. Rumpez, and K. Schneider, GloSE-Lab: Teaching Global Software Engineering, in International Conference on Global Software Engineering (ICGSE). 2011: Helsinki, Findland. p. 156-160.
2. Fortaleza, L.L., T. Conte, S. Marczak, and R. Prikladnicki. Towards a GSE international teaching network: Mapping Global Software Engineering courses. in Collaborative Teaching of Globally Distributed Software Development Workshop (CTGDSD). 2012. p. 1-5.
3. Gotel, O., C. Scharff, and V. Kulkarni, Mixing continents, competences and roles: five years of lessons for software engineering education. Software, IET, 2012. 6(3): p. 199-213.
4. Marques, A.B., R. Rodrigues, and T. Conte. Systematic literature reviews in distributed software development: A tertiary study. in Global Software Engineering (ICGSE), IEEE Seventh International Conference on. 2012: IEEE.
5. Paasivaara, M., C. Lassenius, D. Damian, P. Räty, and A. Schröter. Teaching students global software engineering skills using distributed scrum. International Conference on Software Engineering. 2013: IEEE Press. p. 1128-1137.
6. Raety, P., B. Behm, K.-K. Dikert, M. Paasivaara, C. Lassenius, and D. Damian, Communication Practices in a Distributed Scrum Project, in COINs Conference. 2013: Chile.
7. Cao, L., H. Zhu, and G. Su, Global Software Development Project, in Americas Conference on Information Systems, AMCIS. 2012: Seattle, Washington, USA.
8. Pamulapati, S. and D. Gaddipati, Follow-the-Sun Software Development: Controlled Experiment. 2011, Blekinge Institute of Technology.
9. Rahman, N.A. and S. Sahibuddin, A Systematic Mapping Study on Requirements Elicitation Techniques in Collaborative Application. Lecture Notes on Software Engineering, 2013. 1(1).
10. Ribaud, V. and P. Saliou. Introducing Problem-Based Learning in a Joint Masters Degree: Offshoring Information Technologies. in International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CISSE). 2012.

CHP-4. [407] **Monasor, M.J**., and M. Piattini, Problems and Solutions in Distributed Software Development: A Systematic Review, in Software Engineering Approaches for Offshore and Outsourced Development, K. Berkling, et al., Editors. 2009, Springer Berlin Heidelberg. p. 107-125.	Citations: 9

1. Acharya, M.N. and N. Aslam, Coordination in Global Software Development-Challenges, associated threats, and mitigating practices. 2012, Master Thesis, Software Engineering, Blekinge Institute of Technology, Karlskrona, Sweden.
2. Iqbal, A. and S.S. Abbas, Communication Risks and Best practices in Global Software Development. 2011.
3. Gomes, V. and S. Marczak. Problems? We All Know We Have Them. Do We Have Solutions Too? A Literature Review on Problems and Their Solutions in Global Software Development. in Global Software Engineering (ICGSE), IEEE Seventh International Conference on. 2012: IEEE.
4. Khan, H.H., M.N.r. Mahrin, and S. Chuprat. Risk Generating Situations of Requirement Engineering in Global Software Development. in The Second International Conference on Informatics Engineering & Information Science (ICIEIS). 2013: The Society of Digital Information and Wireless Communication.
5. Boskovic, M., D. Gasevic, G. Grossmann, and M. Stumptner, Service Family Integration in the Cloud: A Feature and Process Model Approach. 2011.
6. Gomes, V.M., Um estudo empírico sobre o impacto da confiança no desempenho de projetos distribuídos de desenvolvimento de software. 2013, Pontifical Catholic University of Rio Grande do Sul.
7. Ruano Mayoral, M., Marco para la asignación de paquetes de trabajo en entornos de desarrollo global de software. 2012, Universidad Carlos III de Madrid.
8. Khan, H.H., M. Naz'ri bin Mahrin, and S. bt Chuprat, Factors Generating Risks during Requirement Engineering Process in Global Software Development Environment. International Journal of Digital Information and Wireless Communications (IJDIWC), 2014. 4(1): p. 63-78.
9. Verner, J.M., O.P. Brereton, B.A. Kitchenham, M. Turner, and M. Niazi, Risks and risk mitigation in global software development: A tertiary study. Information and Software Technology, 2014. 56(1): p. 54-78.

IJ-2. [392] **Monasor, M.J.**, A. Vizcaíno, and M. Piattini, Improving Distributed Software Development in Small and Medium Enterprises. The Open Software Engineering Journal, 2009: p. 26-37.	Citations: 3

1. Shah, Y.H., M. Raza, and S. UlHaq, Communication Issues in GSD. International Journal of Advanced Sciences and Technology, 2012. 40: p. 69-76.
2. Rangevik, A., Software Development across Time Zones: A Study of Globally Distributed Software Development in Small Enterprises. 2013.
3. Vivian, R.L., E.H.M. Huzita, and G.C.L. Leal. Supporting distributed software development through context awareness on software artifacts: the DiSEN-CollaborAR approach. in Proceedings of the 28th Annual ACM Symposium on Applied Computing. 2013: ACM.

JCR-1. [391] **Monasor, M.J.**, A. Vizcaíno, and M. Piattini, Cultural and linguistic problems in GSD: a simulator to train engineers in these issues. Journal of Software Maintenance and Evolution: Research and Practice (Special Issue on Global Software Engineering), 2011. 24(6): p. 707-717. Impact factor JCR (2011): 1.273	Citations: 2

1. Cos, J.A., R. Toval, A. Toval, J.L. Fernández-Aleman, J.M. Carrillo-de-Gea, and J. Nicolas. Internationalization requirements for e-learning audit purposes. in Global Engineering Education Conference (EDUCON), IEEE. 2012. p. 1-6.
2. Misra, S., R. Colomo-Palacios, T. Pusatli, and P. Soto-Acosta, A discussion on the role of people in global software development. Tehnički vjesnik, 2013. 20(3): p. 525-531.

IC-2. [394] **Monasor, M.J.**, A. Vizcaíno, and M. Piattini, Providing Training in GSD by Using a Virtual Environment, in International Conference on Product-Focused Software Development and Process	Citations: 1

Improvement (PROFES). 2012: Madrid, Spain. p. 203-217.	

1. Niazi, M., S. Mahmood, M. Alshayeb, M.R. Riaz, K. Faisal, and N. Cerpa, Challenges of project management in Global Software Development: Initial results. in Science and Information Conference (SAI), 2013: IEEE.

IC-10. [401] **Monasor, M.J.**, A. Vizcaíno, and M. Piattini. An educational environment for learning Global Software Development. in 10th IEEE International Conference on Advanced Learning Technologies (ICALT). 2010. Sousse, Tunisia. p. 99 - 101.	Citations: 1

1. Pereira, C.E., S. Paladini, and F.M. Schaf. Control and Automation Engineering Education: combining physical, remote and virtual labs. in Systems, Signals and Devices (SSD), 9th International Multi-Conference on. 2012: IEEE.

IC-11. [402] **Monasor, M.J.**, A. Vizcaíno, and M. Piattini. A training tool for Global Software Development. in 9th International Conference on Information Technology Based Higher Education and Training (ITHET). 2010. Cappadocia, Turkey. p. 9-16.	Citations: 1

1. Acharya, M.N. and N. Aslam, Coordination in Global Software Development-Challenges, associated threats, and mitigating practices. 2012, Master Thesis, Software Engineering, Blekinge Institute of Technology, Karlskrona, Sweden.

CHP-3. [406] **Monasor, M.J.**, A. Vizcaíno, and M. Piattini, A Tool for Training Students and Engineers in Global Software Development Practices, in Collaboration and Technology, G. Kolfschoten, T. Herrmann, and S. Lukosch, Editors. 2010, Springer Berlin Heidelberg. p. 169-184.	Citations: 1

1. Edwards, H.K., K. Furumo, E. Abu-Shanab, and K.M. Nor. Communication as a Predictor of Success in Projects involving Virtual Teams. in Business and Information (BAI). 2011. Bangkok, Thailand.

IC-9. [400] **Monasor, M.J.**, A. Vizcaíno, and M. Piattini, A Framework for Training Skills for Global Software Development, in International Conference on Global Software Development (ICGSE). 2010: Princeton, NJ, USA. p. 355-356.	Citations: 1

1. Ogata, S. and S. Matsuura, Training of requirements analysis modeling with UML-based prototype generation tool, in Proceedings of the 5th India Software Engineering Conference. 2012, ACM: Kanpur, India.

IC-12. [403] **Monasor, M.J.**, et al., Towards a Global Software Development Community Web: Identifying Patterns and Scenarios, in PARIS Workshop, International Conference on Global Software Development (ICGSE). 2013: Bari, Italy. p. 41-46.	Citations: 1

1. Zanetti, M.S., A complex systems approach to software engineering. 2013, Eidgenössische Technische Hochschule ETH Zürich: Zurich.

CHP-2. [405] **Monasor, M.J.**, M. Piattini, and A. Vizcaíno, A Systematic Review of Distributed Software Development: Problems and Solutions, in Handbook of Research on Software Engineering and Productivity Technologies: Implications of Globalization. 2010, IGI Global. p. 209-225.	Citations: 1

1. Verner, J.M., O.P. Brereton, B.A. Kitchenham, M. Turner, and M. Niazi, Risks and risk mitigation in global software development: A tertiary study. Information and Software Technology, 2014. 56(1): p. 54-78.

Bibliography

1. Herbsleb, J.D. and D. Moitra, *Guest Editors' Introduction: Global Software Development.* IEEE Softw., 2001. **18**(2): p. 16-20.
2. Damian, D., S. Marczak, M. Dascalu, M. Heiss, and A. Liche, *Using Real-Time Conferencing Tools to Leverage the Collaborative Potential of Global Teams: An Experience Report from Siemens IT Solutions and Services*, in *Fourth IEEE International Conference on Global Software Engineering (ICGSE)*. 2009, IEEE Computer Society: Limerick, Ireland. p. 239-243.
3. Hsieh, Y., *Culture and Shared Understanding in Distributed Requirements Engineering*, in *IEEE International Conference on Global Software Engineering (ICGSE)*. 2006, IEEE Computer Society: Florianópolis, Brazil. p. 101-108.
4. Lutz, B., *Linguistic Challenges in Global Software Development: Lessons Learned in an International SW Development Division*, in *Fourth IEEE International Conference on Global Software Engineering (ICGSE)*. 2009, IEEE Computer Society: Limerick, Ireland. p. 249-253.
5. Seat, E. and S.M. Lord, *Enabling effective engineering teams: a program for teaching interaction skills*, in *Frontiers in Education Conference (FIE). 28th Annual*. 1998. p. 246-251.
6. Petkovic, D., G. Thompson, and R. Todtenhoefer, *Teaching practical software engineering and global software engineering: evaluation and comparison.* SIGCSE Bull., 2006. **38**(3): p. 294-298.
7. Wainfan, L., *Challenges in Virtual Collaboration: Videoconferencing Audioconferencing and Computer--Mediated Communications*. 2005: RAND Corporation. 128.
8. Shah, H., N.J. Nersessian, M.J. Harrold, and W. Newstetter, *Studying the influence of culture in global software engineering: thinking in terms of cultural models*, in *Proceedings of the 4th international conference on Intercultural Collaboration*. 2012, ACM: Bengaluru, India. p. 77-86.
9. Richardson, I., S. Moore, D. Paulish, V. Casey, and D. Zage, *Globalizing Software Development in the Local Classroom*, in *Proceedings of the 20th Conference on Software Engineering Education & Training*. 2007, IEEE Computer Society. p. 64-71.
10. Lago, P., H. Muccini, L. Beus-Dukic, I. Crnkovic, S. Punnekkat, and H.V. Vliet, *Towards a European Master Programme on Global Software Engineering*, in *Proceedings of the 20th Conference on Software Engineering Education & Training*. 2007, IEEE Computer Society: Dublin, Ireland. p. 184-194.
11. Petkovic, D., G.D. Thompson, and R. Todtenhoefer, *Assessment and comparison of local and global SW engineering practices in a classroom setting*, in *Proceedings of the 13th annual conference on Innovation and technology in computer science education*. 2008, ACM: Madrid, Spain. p. 78-82.
12. Damian, D., A. Hadwin, and B. Al-Ani. *Instructional design and assessment strategies for teaching global software development: a framework*. in *International Conference on Software Engineering (ICSE)*. 2006: ACM Press New York, NY, USA. p. 685-690.
13. Gotel, O., V. Kulkarni, C. Scharff, and L. Neak, *Working Across Borders: Overcoming Culturally-Based Technology Challenges in Student Global Software Development*, in *21st Conference on Software Engineering Education and Training*. 2008, IEEE Computer Society. p. 33-40.
14. Bellur, U., *An Academic Perspective on Globalization in the Software Industry*, in *Proceedings of the 30th Annual International Computer Software and Applications Conference*. 2006, IEEE Computer Society. p. 53-54.
15. Ghezzi, C. and D. Mandrioli, *The challenges of software engineering education*, in *Proceedings of the 27th international conference on Software engineering*. 2005, ACM: St. Louis, MO, USA. p. 637-638.
16. Prikladnicki, R. and L. Pilatti, *Improving Contextual Skills in Global Software Engineering: A Corporate Training Experience*, in *IEEE International Conference on Global Software Engineering (ICGSE)*. 2008, IEEE Computer Society: Bangalore, India. p. 239-243.

17. Burnell, L.J., J.W. Priest, and J.R. Durrett, *Teaching Distributed Multidisciplinary Software Development.* IEEE Softw., 2002. **19**(5): p. 86-93.
18. Meneely, A. and L. Williams, *On preparing students for distributed software development with a synchronous, collaborative development platform*, in *Proceedings of the 40th ACM technical symposium on Computer science education*. 2009, ACM: Chattanooga, TN, USA. p. 529-533.
19. Setamanit, S.-o., W. Wakeland, and D. Raffo, *Using simulation to evaluate global software development task allocation strategies.* Software Process: Improvement and Practice, 2007: p. 491-503.
20. Sarker, S. and S. Sahay, *Implications of space and time for distributed work: an interpretive study of US-Norwegian systems development teams.* Eur. J. Inf. Syst., 2004. **13**(1): p. 3-20.
21. Faulk, S., M. Young, D. Weiss, and L. Yu, *Collaborative teaching of globally distributed software development: community building workshop (CTGDSD)*, in *Proceedings of the 33rd International Conference on Software Engineering*. 2011, ACM: Waikiki, Honolulu, HI, USA. p. 1208-1209.
22. Martins, L.L., L.L. Gilson, and M.T. Maynard, *Virtual Teams: What Do We Know and Where Do We Go From Here?* Journal of Management, 2004. **30**(6): p. 805-835.
23. Davison, R., *Offshoring information technology: Sourcing and outsourcing to a global workforce.* Information Technology for Development, 2007. **13**(1): p. 101-102.
24. Prikladnicki, R., D. Damian, and J.L.N. Audy. *Patterns of Evolution in the Practice of Distributed Software Development: Quantitative Results from a Systematic Review*. in *12th International Conference on Evaluation and Assessment in Software Engineering (EASE)* 2008. University of Bari, Italy. p. 100-109.
25. Prikladnicki, R., J.L.N. Audy, and J.R. Evaristo, *Distributed Software Development: Toward an Understanding of the Relationship Between Project Team, Users and Customers.* ICEIS, 2003: p. 417-423.
26. McConnell, S., *Rapid Development: Taming Wild Software Schedules*. 1996, Redmond, Wa: Microsoft Press. 660.
27. Grinter, R.E., J.D. Herbsleb, and D.E. Perry, *The geography of coordination: dealing with distance in R&D work*, in *Proceedings of the international ACM SIGGROUP conference on Supporting group work*. 1999, ACM: Phoenix, Arizona, United States. p. 306-315.
28. Oshri, I., J. Kotlarsky, and L.P. Willcocks, *Global software development: Exploring socialization and face-to-face meetings in distributed strategic projects.* The Journal of Strategic Information Systems, 2007. **16**(1): p. 25-49.
29. Werner, K., D. Rombach, and R. Feldmann, *Outsourcing in India.* IEEE Software, 2001. **18**(2): p. 78-86.
30. Lacity, M.C., S.A. Khan, and L.P. Willcocks, *A review of the IT outsourcing literature: Insights for practice.* The Journal of Strategic Information Systems, 2009. **18**(3): p. 130-146.
31. Carmel, E. and R. Agarwal, *Tactical Approaches for Alleviating Distance in Global Software Development.* IEEE Software, 2001. **18**(2): p. 22-29.
32. Damian, D., J. Chisan, P. Allen, and B. Corrie, *Awareness meets requirements management: awareness needs in global software development*, in *Workshop on Global Software Development, International Conference on Software Engineering (ICSE)*. 2003: Portland, Oregon. p. 7-11.
33. Herbsleb, J.D. and D. Moitra, *Global software development.* IEEE Software, 2001. **18**(2): p. 16-20.
34. Christof Ebert , P.D.N., *Surviving Global Software Development.* IEEE Software, 2001. **18**(2): p. 62-69.
35. Carmel, E., *Global Software Teams: Collaborating Across Borders and Time Zones*. 1999, Upper Saddle River, NJ, USA: Prentice Hall.

36. Herbsleb, J.D. and R.E. Grinter, *Splitting the organization and integrating the code: Conway's law revisited*, in *Proceedings of the 21st international conference on Software engineering (ICSE)*. 1999, ACM: Los Angeles, California, United States. p. 85-95.
37. Šmite, D., C. Wohlin, Z. Galviņa, and R. Prikladnicki, *An empirically based terminology and taxonomy for global software engineering.* Empirical Software Engineering, 2012: p. 1-49.
38. Layman, L., L. Williams, D. Damian, and H. Bures, *Essential communication practices for Extreme Programming in a global software development team.* Information and Software Technology, 2006. **48**(9): p. 781-794.
39. Ågerfalk, P., B. Fitzgerald, H. Holmström, B. Lings, B. Lundell, and E. Conchúir. *A framework for considering opportunities and threats in distributed software development* in *International Workshop on Distributed Software Development*. 2005. Paris, France p. 47-61.
40. Herbsleb, J.D., D.J. Paulish, and M. Bass, *Global software development at siemens: experience from nine projects*, in *Proceedings of the 27th international conference on Software engineering*. 2005, ACM: St. Louis, MO, USA. p. 524-533.
41. Battin, R.D., R. Crocker, J. Kreidler, and K. Subramanian, *Leveraging Resources in Global Software Development.* IEEE Softw., 2001. **18**(2): p. 70-77.
42. Krishna, S., S. Sahay, and G. Walsham, *Managing cross-cultural issues in global software outsourcing.* Commun. ACM, 2004. **47**(4): p. 62-66.
43. Damian, D., F. Lanubile, and H. Oppenheimer, *Addressing the Challenges of Software Industry Globalization: The Workshop on Global Software Development. .* ICSE, 2003: p. 793-794.
44. Herbsleb, J.D. *Global Software Engineering: The Future of Socio-technical Coordination*. in *International Conference on Software Engineering: Future of Software Engineering (FOSE'07)*. 2007. Minneapolis, MN, USA: IEEE Computer Society. p. 188-198.
45. Karolak, D.W., *Global Software Development: Managing Virtual Teams and Environments*. 1999: IEEE Computer Society Press. 158.
46. Kitchenham, B. and S. Charters, *Guidelines for performing Systematic Literature Reviews in Software Engineering*, in *Technical Report EBSE-2007-001*. 2007, Keele University and Durham University Joint Report.
47. Pino, F.J., F. García, and M. Piattini, *Software Process Improvement in Small and Medium Software Enterprises: A Systematic Review*, in *Software Quality Journal*, S. Netherlands, Editor. 2007. p. 237-261.
48. Marvin, V.Z. and R.W. Dolores, *Experimental Models for Validating Technology*. 1998: Los Alamitos, CA, USA p. 23-31.
49. *ISO/IEC 12207:2002*, ISO, Editor. 2002, AMENDMENT 1: Information technology - Software life cycle processes. International Organization for Standardization: Geneva, Switzerland.
50. Lloyd, W., M.B. Rosson, and J. Arthur. *Effectiveness of Elicitation Techniques in Distributed Requirements Engineering*. in *10th Anniversary IEEE Joint International Conference on Requirements Engineering (RE)*. 2002. Essen, Germany. p. 311-318.
51. Fernández, A., B. Garzaldeen, I. Grützner, and J. Münch, *Guided support for collaborative modeling, enactment and simulation of software development processes.* Software Process: Improvement and Practice, 2004. **9**(2): p. 95-106.
52. Biehl, J.T., M. Czerwinski, G. Smith, and G.G. Robertson. *FASTDash: a visual dashboard for fostering awareness in software teams*. in *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2007. San Jose, California, USA: ACM. p. 1313-1322.
53. Ma, J., J. Li, W. Chen, R. Conradi, J. Ji, and C. Liu, *A state-of-the-practice study on communication and coordination between chinese software suppliers and their global outsourcers.* Software Process: Improvement and Practice, 2008. **13**(3): p. 233-247.
54. Berenbach, B., *Impact of organizational structure on distributed requirements engineering processes: lessons learned*, in *Proceedings of the 2006 international workshop on Global software development for the practitioner*. 2006, ACM: Shanghai, China. p. 15-19.

55. Madachy, R.J., *Cost modeling of distributed team processes for global development and Software-Intensive Systems of Systems.* Software Process: Improvement and Practice, 2008. **13**(1): p. 51-61.
56. Bruegge, B., A.H. Dutoit, and T. Wolf, *Sysiphus: Enabling informal collaboration in global software development*, in *International Conference on In Global Software Engineering, (ICGSE)*. 2006. p. 139-148.
57. Moe, N.B. and D. Šmite, *Understanding a lack of trust in Global Software Teams: a multiple-case study.* Softw. Process, 2008. **13**(3): p. 217-231.
58. Schümmer, T. and J.M. Haake, *Supporting distributed software development by modes of collaboration*, in *Proceedings of the seventh conference on European Conference on Computer Supported Cooperative Work*. 2001, Kluwer Academic Publishers: Bonn, Germany. p. 79-98.
59. Mohan, K. and B. Ramesh, *Traceability-based knowledge integration in group decision and negotiation activities.* Decision Support Systems, 2007. **43**(3): p. 968-989.
60. Carey, J.M., *Creating global software: A conspectus and review.* Interacting with Computers, 1998. **9**(4): p. 449-465.
61. Moll, J.v., J. Jacobs, R. Kusters, and J. Trienekens, *Defect detection oriented lifecycle modeling in complex product development.* Information and Software Technology, 2004. **46**(10): p. 665-675.
62. Casey, V. and I. Richardson, *Project Management within Virtual Software Teams*, in *International Conference on Global Software Engineering (ICGSE)*. 2006, IEEE Computer Society: Florianopolis, Brazil. p. 33-42.
63. Munkvold, B.E. and I. Zigurs, *Process and technology challenges in swift-starting virtual teams.* Information & Management, 2007. **44**(3): p. 287-299.
64. Clerc, V., *Towards architectural knowledge management practices for global software development*, in *Proceedings of the 3rd international workshop on Sharing and reusing architectural knowledge*. 2008, ACM: Leipzig, Germany.
65. Nagappan, N., B. Murphy, and V. Basili, *The influence of organizational structure on software quality: an empirical case study*, in *Proceedings of the 30th international conference on Software engineering*. 2008, ACM: Leipzig, Germany. p. 521-530.
66. Crowston, K., Q. Li, K. Wei, U.Y. Eseryel, and J. Howison, *Self-organization of teams for free/libre open source software development.* Information and Software Technology, 2007. **49**(6): p. 564-575.
67. Narayanaswamy, K. and N.M. Goldman, *A flexible framework for cooperative distributed software development.* Journal of Systems and Software, 1991. **16**(2): p. 97-105.
68. Lucia, A.D., F. Fasano, G. Scanniello, and G. Tortora, *Enhancing collaborative synchronous UML modelling with fine-grained versioning of software artefacts.* Journal of Visual Languages & Computing, 2007. **18**(5): p. 492-503.
69. Araujo, R.M.d. and M.R.S. Borges, *The role of collaborative support to promote participation and commitment in software development teams.* Software Process: Improvement and Practice, 2007. **12**(3): p. 229-246.
70. Dossick, S.E. and G.E. Kaiser. *CHIME: a metadata-based distributed software development environment*. in *Proceedings of the 7th European software engineering conference held jointly with the 7th ACM SIGSOFT international symposium on Foundations of software engineering*. 1999. Toulouse, France: Springer-Verlag. p. 464-475.
71. Ocker, R.J. and J. Fjermestad, *Communication differences in virtual design teams: findings from a multi-method analysis of high and low performing experimental teams.* SIGMIS Database, 2008. **39**(1): p. 51-67.
72. Al-Ani, B., E. Trainer, R. Ripley, A. Sarma, A.v.d. Hoek, and D. Redmiles, *Continuous coordination within the context of cooperative and human aspects of software engineering*, in *International workshop on Cooperative and human aspects of software engineering (CHASE)*. 2008, ACM: Leipzig, Germany. p. 1-4.

73. Ovaska, P., M. Rossi, and P. Marttiin, *Architecture as a coordination tool in multi-site software development.* Software Process: Improvement and Practice, 2003. **8**(4): p. 233-247.
74. Akmanligil, M. and P.C. Palvia, *Strategies for global information systems development.* Information & Management, 2004. **42**(1): p. 45-59.
75. Paasivaara, M. and C. Lassenius, *Collaboration practices in global inter-organizational software development projects.* Software Process: Improvement and Practice, 2003. **8**(4): p. 183-199.
76. Froehlich, J. and P. Dourish. *Unifying Artifacts and Activities in a Visual Tool for Distributed Software Development Teams*. in *Proceedings of the 26th International Conference on Software Engineering*. 2004: IEEE Computer Society. p. 387-396.
77. Pilatti, L., J.L.N. Audy, and R. Prikladnicki, *Software configuration management over a global software development environment: lessons learned from a case study*, in *International workshop on Global software development for the practitioner*. 2006, ACM: Shanghai, China. p. 45-50.
78. Powell, A., G. Piccoli, and B. Ives, *Virtual teams: a review of current literature and directions for future research.* SIGMIS Database, 2004. **35**(1): p. 6-36.
79. Gomes, P.J. and N.R. Joglekar, *Linking modularity with problem solving and coordination efforts.* Managerial and Decision Economics, 2008. **5**: p. 443-457.
80. Gorton, I. and S. Motwani, *Issues in co-operative software engineering using globally distributed teams.* Information and Software Technology, 1996. **38**(10): p. 647-655.
81. Prikladnicki, R., J.L.N. Audy, and R. Evaristo, *Global software development in practice lessons learned.* Software Process: Improvement and Practice, 2003. **8**(4): p. 267-281.
82. Gousios, G., E. Kalliamvakou, and D. Spinellis, *Measuring developer contribution from software repository data*, in *International working conference on Mining software repositories*. 2008, ACM: Leipzig, Germany. p. 129-132.
83. Prikladnicki, R., J.L.N. Audy, and R. Evaristo. *A Reference Model for Global Software Development: Findings from a Case Study*. in *International Conference on Global Software Engineering, (ICGSE)* 2006. p. 18-28.
84. Halverson, C.A., J.B. Ellis, C. Danis, and W.A. Kellogg. *Designing task visualizations to support the coordination of work in software development*. in *20th anniversary conference on Computer supported cooperative work (CSCW)*. 2006. Banff, Alberta, Canada: ACM. p. 39-48.
85. Ramasubbu, N. and R.K. Balan. *Globally distributed software development project performance: an empirical analysis*. in *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*. 2007. Dubrovnik, Croatia: ACM. p. 125-134.
86. Hanks, B., *Empirical evaluation of distributed pair programming.* International Journal of Human-Computer Studies, 2008. **66**(7): p. 530–544.
87. Sakthivel, S., *Virtual workgroups in offshore systems development.* Information and Software Technology, 2005. **47**(5): p. 305-318.
88. Heistracher, T., T. Kurz, G. Marcon, and C. Masuch. *Collaborative Software Engineering with a Digital Ecosystem*. in *International Conference on Global Software Engineering, (ICGSE)* 2006. p. 119-126.
89. Sangwan, R.S. and J. Ros, *Architecture leadership and management in globally distributed software development*, in *Proceedings of the first international workshop on Leadership and management in software architecture (LMSA)*. 2008, ACM: Leipzig, Germany. p. 17-22.
90. Herbsleb, J.D., A. Mockus, T.A. Finholt, and R.E. Grinter. *Distance, dependencies, and delay in a global collaboration*. in *ACM conference on Computer supported cooperative work*. 2000. Philadelphia, Pennsylvania, United States: ACM. p. 319-328.
91. Babar, M.A., B. Kitchenham, L. Zhu, I. Gorton, and R. Jeffery, *An empirical study of groupware support for distributed software architecture evaluation process.* Journal of Systems and Software, 2006. **79**(7): p. 912-925.

92. Herbsleb, J.D., A. Mockus, T.A. Finholt, and R.E. Grinter, *An empirical study of global software development: distance and speed*, in *Proceedings of the 23rd International Conference on Software Engineering*. 2001, IEEE Computer Society: Toronto, Ontario, Canada. p. 81-90.
93. Sarkar, S., R. Sindhgatta, and K. Pooloth. *A collaborative platform for application knowledge management in software maintenance projects*. in *Proceedings of the 1st Bangalore annual Compute conference*. 2008. Bangalore, India: ACM. p. 1-7.
94. Ali-Babar, M., *The application of knowledge-sharing workspace paradigm for software architecture processes*, in *Proceedings of the 3rd international workshop on Sharing and reusing architectural knowledge*. 2008, ACM: Leipzig, Germany. p. 45-48.
95. Sarma, A., Z. Noroozi, and A.v.d. Hoek. *Palantír: raising awareness among configuration management workspaces*. in *Proceedings of the 25th International Conference on Software Engineering*. 2003. Portland, Oregon: IEEE Computer Society. p. 444 - 454.
96. Holmes, R. and R.J. Walker, *Promoting developer-specific awareness*, in *International workshop on Cooperative and human aspects of software engineering (CHASE)*. 2008, ACM: Leipzig, Germany. p. 61-64.
97. Shami, N.S., N. Bos, Z. Wright, S. Hoch, K.Y. Kuan, J. Olson, and G. Olson, *An experimental simulation of multi-site software development*, in *Conference of the Centre for Advanced Studies on Collaborative research*. 2004, IBM Press: Markham, Ontario, Canada. p. 255-266.
98. Holmstrom, H., E.O. Conchuir, P.J. Agerfalk, and B. Fitzgerald, *Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance*, in *International Conference on Global Software Engineering (ICGSE)*. 2006, IEEE Computer Society. p. 3-11.
99. Sengupta, B., S. Chandra, and V. Sinha, *A research agenda for distributed software development*, in *International Conference on Software Engineering (ICSE)*. 2006, ACM: Shanghai, China. p. 731-740.
100. Biehl, J.T., W.T. Baker, B.P. Bailey, D.S. Tan, K.M. Inkpen, and M. Czerwinski. *Impromptu: a new interaction framework for supporting collaboration in multiple display environments and its field evaluation for co-located software development*. in *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*. 2008. Florence, Italy: ACM. p. 939-948.
101. Smite, D., *Global software development projects in one of the biggest companies in Latvia: is geographical distribution a problem?* Software Process: Improvement and Practice, 2006. **11**(1): p. 61-76.
102. Kotlarsky, J., P.C.v. Fenema, and L.P. Willcocks, *Developing a knowledge-based perspective on coordination: The case of global software projects.* Information & Management, 2008. **45**(2): p. 96-108.
103. Souza, C.R.d., S. Quirk, E. Trainer, and D.F. Redmiles. *Supporting collaborative software development through the visualization of socio-technical dependencies*. in *International ACM conference on Supporting group work*. 2007. Sanibel Island, Florida, USA: ACM. p. 147-156.
104. Kuni, R. and N. Bhushan, *IT Application Assessment Model for Global Software Development*, in *Proceedings of the IEEE international conference on Global Software Engineering*. 2006, IEEE Computer Society.
105. Spanjers, H., M.t. Huurne, B. Graaf, M. Lormans, D. Bendas, and R.v. Solingen, *Tool Support for Distributed Software Engineering*, in *Proceedings of the IEEE international conference on Global Software Engineering*. 2006, IEEE Computer Society.
106. Wongthongtham, P., E. Chang, and T. Dillon. *Ontology-based multi-agent system to multi-site software development*. in *Workshop on Quantitative techniques for software agile process*. 2004. Newport Beach, California: ACM. p. 66-75.
107. Storey, M.-A.D., D. Čubranić, and D.M. German. *On the use of visualization to support awareness of human activities in software development: a survey and a framework*. in *ACM symposium on Software visualization*. 2005. St. Louis, Missouri: ACM. p. 193-202.

108. Suzuki, J. and Y. Yamamoto, *SoftDock: A Distributed Collaborative Platform for Model-Based Software Development*, in *Proceedings of the 10th International Workshop on Database & Expert Systems Applications*. 1999, IEEE Computer Society. p. 672 - 676.
109. Lee, G., W. DeLone, and J.A. Espinosa, *Ambidextrous coping strategies in globally distributed software development projects.* Commun. ACM, 2006. **49**(10): p. 35-40.
110. Baentsch, M., G. Molter, and P. Sturm, *WebMake: Integrating distributed software development in a structure-enhanced Web.* Computer Networks and ISDN Systems, 1995. **27**(6): p. 789-800.
111. Lindqvist, E., B. Lundell, and B. Lings. *Distributed development in an intra-national, intra-organisational context: an experience report*. in *International workshop on Global software development for the practitioner*. 2006. Shanghai, China: ACM. p. 80-86.
112. Tamura, Y., S. Yamada, and M. Kimura, *A reliability assessment tool for distributed software development environment based on Java and J/Link.* European Journal of Operational Research, 2006. **175**(1): p. 435-445.
113. Aranda, G.N., A. Vizcaino, A. Cechich, M. Piattini, and J.J. Castro-Schez. *Cognitive-Based Rules as a Means to Select Suitable Groupware Tools*. in *International Conference on Cognitive Informatics, (ICCI)* 2006. p. 418-423.
114. Taxen, L., *An integration centric approach for the coordination of distributed software development projects.* Information and Software Technology, 2006. **48**(9): p. 767-780.
115. Liu, S., Y. Zheng, H. Shen, S. Xia, and C. Sun. *Real-Time Collaborative Software Modeling Using UML with Rational Software Architect*. in *International Conference on Collaborative Computing: Networking, Applications and Worksharing* 2006.
116. Thissen, M.R., J.M. Page, M.C. Bharathi, and T.L. Austin. *Communication tools for distributed software development teams*. in *ACM SIGMIS CPR conference on Computer personnel doctoral consortium and research conference: The global information technology workforce*. 2007. St. Louis, Missouri, USA: ACM. p. 28-35.
117. Wongthongtham, P., E. Chang, T. Dillon, and I. Sommerville, *Ontology-based multi-site software development methodology and tools.* Journal of Systems Architecture, 2006. **52**(11): p. 640-653.
118. Tiako, P.F., *Collaborative approach for modeling and performing mobile software process components*, in *International conference on Collaborative technologies and systems*. 2005, IEEE Computer Society: Saint Louis, Missouri, USA. p. 40 - 47.
119. Xiao, W., C. Chi, and M. Yang. *On-line collaborative software development via wiki*. in *Proceedings of the 2007 international symposium on Wikis*. 2007. Montreal, Quebec, Canada: ACM. p. 177-183.
120. Vale, S. and S. Hammoudi, *Towards context independence in distributed context-aware applications by the model driven approach*, in *Proceedings of the 3rd international workshop on Services integration in pervasive environments*. 2008, ACM: Sorrento, Italy. p. 31-36.
121. Zhuge, H., *Knowledge flow management for distributed team software development.* Knowledge-Based Systems, 2002. **15**(8): p. 465-471.
122. Lanubile, F., T. Mallardo, and F. Calefato, *Tool support for geographically dispersed inspection teams.* Software Process: Improvement and Practice, 2003. **8**(4): p. 217-231.
123. Ramesh, B., L. Cao, K. Mohan, and P. Xu, *Can distributed software development be agile?* Commun. ACM, 2006. **49**(10): p. 41-46.
124. Siakas, K.V. and B. Balstrup, *Software outsourcing quality achieved by global virtual collaboration.* Software Process: Improvement and Practice, 2006. **11**(3): p. 319-328.
125. Sangwan, R., M. Bass, N. Mullick, D.J. Paulish, and J. Kazmeier, *Global Software Development Handbook (Auerbach Series on Applied Software Engineering Series)*. 2006, Boston, MA, USA: Auerbach Publications.

126. Hossain, E., M. Babar, and H. Paik, *Using Scrum in Global Software Development: A Systematic Literature Review*, in *International Conference on Global Software Engineering (ICGSE)*. 2009, IEEE Computer Society: Limerick, Ireland. p. 175-184.
127. Noll, J., S. Beecham, and I. Richardson, *Global software development and collaboration: barriers and solutions.* ACM Inroads, 2010. **1**(3): p. 66-78.
128. Smite, D. and C. Wohlin, *A Whisper of Evidence in Global Software Engineering.* Software, IEEE, 2011. **28**(4): p. 15-18.
129. Alsudairi, M. and Y.K. Dwivedi, *A multi-disciplinary profile of IS/IT outsourcing research.* Journal of Enterprise Information Management, 2004. **23**(2): p. 215-258.
130. Kroll, J., J.L.N. Audy, and R. Prikladnicki. *Mapping the Evolution of Research on Global Software Engineering - A Systematic Literature Review*. in *ICEIS*. 2011. p. 260-265.
131. Verner, J.M., O.P. Brereton, B.A. Kitchenham, M. Turner, and M. Niazi. *Systematic literature reviews in global software development: A tertiary study*. in *Evaluation & Assessment in Software Engineering (EASE), 16th International Conference on*. 2012. p. 2-11.
132. Smite, D., C. Wohlin, R. Feldt, and T. Gorschek. *Reporting Empirical Research in Global Software Engineering: A Classification Scheme*. in *International Conference on Global Software Engineering, (ICGSE)*. 2008. p. 173-181.
133. Treude, C., M.-a. Storey, and J. Weber, *Empirical studies on collaboration in software development: A systematic Literature Review*, in *Technical Report DCS-331-IR*. 2009, Department of Computer Science, University of Victoria.
134. Humayun, M., C. Gang, and I. Masood, *An empirical study on investigating the role of KMS in promoting trust within GSD teams*, in *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering (EASE)*. 2013, ACM: Porto de Galinhas, Brazil. p. 207-211.
135. Borges, A., S. Soares, S. Meira, H. Tomaz, R. Rocha, and C. Costa, *Ontologies supporting the distributed software development: a systematic mapping study*, in *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*. 2013, ACM: Porto de Galinhas, Brazil.
136. Ali, N., S. Beecham, and I. Mistrik, *Architectural Knowledge Management in Global Software Development: A Review*, in *International Conference on Global Software Engineering (ICGSE)*. 2010, IEEE Computer Society. p. 347 - 352.
137. Almeida, L.H., P.R. Pinheiro, and A.B. Albuquerque, *Applying multi-criteria decision analysis to global software development with scrum project planning*, in *Proceedings of the 6th international conference on Rough sets and knowledge technology (RSKT)*. 2011, Springer-Verlag: Banff, Canada. p. 311-320.
138. Hossain, E., P.L. Bannerman, and R. Jeffery, *Towards an understanding of tailoring scrum in global software development: a multi-case study*, in *International Conference on on Software and Systems Process*. 2011, ACM: Waikiki, Honolulu, HI, USA. p. 110-119.
139. Jalali, S. and C. Wohlin, *Global software engineering and agile practices: a systematic review.* Journal of Software: Evolution and Process, 2011. **24**(6): p. 643-659.
140. Šteinberga, L. and D. Šmite, *Towards Understanding of Software Engineer Motivation in Globally Distributed Projects*, in *International Conference on Global Software Engineering Workshop (ICGSEW)*. 2011, IEEE Computer Society. p. 117 - 119.
141. Fernando, B.A.J., T. Hall, and A. Fitzpatrick, *The impact of media selection on stakeholder communication in agile global software development: a preliminary industrial case study*, in *Proceedings of the 49th SIGMIS annual conference on Computer personnel research*. 2011, ACM: San Antonio, Texas, USA. p. 131-139.
142. Ramesh, B., K. Mohan, and L. Cao, *Ambidexterity in Agile Distributed Development: An Empirical Investigation.* Info. Sys. Research, 2012. **23**(2): p. 323-339.
143. Persson, J.S. and L. Mathiassen, *A Process for Managing Risks in Distributed Teams.* IEEE Softw., 2010. **27**(1): p. 20-29.

144. Prikladnicki, R. and J.L.N. Audy, *Process models in the practice of distributed software development: A systematic review of the literature.* Inf. Softw. Technol., 2010. **52**(8): p. 779-791.
145. Rocha, R.G.C., C. Costa, C. Rodrigues, R.R.d. Azevedo, I.H. Junior, S. Meira, and R. Prikladnicki, *Collaboration Models in Distributed Software Development: a Systematic Review.* CLEI Electronic Journal, 2011. **14**(2).
146. Costa, C., C. Cunha, R. Rocha, A.C.C. França, F.Q.B.d. Silva, and R. Prikladnicki, *Models and tools for managing distributed software development: a systematic literature review*, in *Proceedings of the 14th international conference on Evaluation and Assessment in Software Engineering*. 2010, British Computer Society: UK. p. 73-76.
147. Cataldo, M. and J.D. Herbsleb, *Factors leading to integration failures in global feature-oriented development: an empirical analysis*, in *Proceedings of the 33rd International Conference on Software Engineering*. 2011, ACM: Waikiki, Honolulu, HI, USA. p. 161-170.
148. Portillo-Rodríguez, J., A. Vizcaíno, M. Piattini, and S. Beecham, *Tools used in Global Software Engineering: A systematic mapping review.* Information and Software Technology, 2012. **54**(7): p. 663-685.
149. Chauhan, M.A. and M.A. Babar, *Cloud infrastructure for providing tools as a service: quality attributes and potential solutions*, in *Proceedings of the WICSA/ECSA Companion Volume*. 2012, ACM: Helsinki, Finland. p. 5-13.
150. Guzmán, J.G., J.S. Ramos, A.d.A. Seco, and A. Sanz-Esteban, *Success Factors for the Management of Global Virtual Teams for Software Development.* International Journal of Human Capital and Information Technology Professionals, 2011. **2**(2): p. 48-59.
151. Ebling, T., J.L.N. Audy, and R. Prikladnicki, *A Systematic Literature Review of Requirements Engineering in Distributed Software Development Environments*, in *ICEIS*. 2009. p. 363-366.
152. Bjarnason, E., *Distances between requirements engineering and later software development activities: a systematic map*, in *Proceedings of the 19th international conference on Requirements Engineering: Foundation for Software Quality*. 2013, Springer-Verlag: Essen, Germany. p. 292-307.
153. Lopez, A., J. Nicolas, and A. Toval. *Risks and Safeguards for the Requirements Engineering Process in Global Software Development*. in *International Conference on Global Software Engineering (ICGSE)*. 2009. p. 394-399.
154. Fauzi, S.S.M., P.L. Bannerman, and M. Staples. *Software Configuration Management in Global Software Development: A Systematic Map*. in *17th Asia Pacific Software Engineering Conference*. 2010. Sydney, Australia. p. 404-413.
155. Costa, C. and L. Murta, *Version Control in Distributed Software Development: a Systematic Mapping Study*, in *International Conference on Global Software Development (ICGSE)*. 2013: Bari, Italy. p. 90-99.
156. Khan, S.U., M. Niazi, and R. Ahmad, *Critical Success Factors for Offshore Software Development Outsourcing Vendors: A Systematic Literature Review*, in *International Conference on Global Software Engineering (ICGSE)*. 2009, IEEE Computer Society: Limerick, Ireland. p. 207-216.
157. Khan, S.U., M. Niazi, and R. Ahmad, *Barriers in the selection of offshore software development outsourcing vendors: An exploratory study using a systematic literature review.* Inf. Softw. Technol., 2011. **53**(7): p. 693-706.
158. Anh, N.D. *Dispersion, coordination and performance in global software teams: A systematic review*. in *ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. 2012. Lund, Sweden Sweden. p. 129-138.
159. Nurdiani, I., R. Jabangwe, D. Šmite, and D. Damian, *Risk Identification and Risk Mitigation Instruments for Global Software Development: Systematic Review and Survey Results*, in *Sixth International Conference on Global Software Engineering Workshop (ICGSEW)*. 2011, IEEE Computer Society. p. 36 - 41.

160. Steinmacher, I., A.P. Chaves, and M.A. Gerosa, *Awareness support in global software development: a systematic review based on the 3C collaboration model*, in *Proceedings of the 16th international conference on Collaboration and technology*. 2010, Springer-Verlag: Maastricht, The Netherlands. p. 185-201.
161. Vivian, R.L., E.H.M. Huzita, G.C.L. Leal, and A.P.C. Steinmacher, *Context-awareness on software artifacts in distributed software development: a systematic review*, in *Proceedings of the 17th international conference on Collaboration and technology*. 2011, Springer-Verlag: Paraty, Brazil. p. 30-44.
162. Steinmacher, I., A.P. Chaves, and M.A. Gerosa, *Awareness Support in Distributed Software Development: A Systematic Review and Mapping of the Literature.* Comput. Supported Coop. Work, 2013. **22**(2-3): p. 113-158.
163. Kroll, J., S.I. Hashmi, I. Richardson, and J.L.N. Audy, *A Systematic Literature Review of Best Practices and Challenges in Follow-the-Sun Software Development*, in *PARIS Workshop, International Conference on Global Software Development (ICGSE)*. 2013: Bari, Italy. p. 18-23.
164. Budgen, D., S. Drummond, P. Brereton, and N. Holland. *What scope is there for adopting evidence-informed teaching in SE?* in *Software Engineering (ICSE), 2012 34th International Conference on*. 2012. p. 1205-1214.
165. Su, H., S. Jodis, and H. Zhang, *Providing an integrated software development environment for undergraduate software engineering courses.* J. Comput. Small Coll., 2007. **23**(2): p. 143-149.
166. Murphy, C., D. Phung, and G. Kaiser, *A distance learning approach to teaching eXtreme programming.* SIGCSE Bull., 2008. **40**(3): p. 199-203.
167. Silva, F.Q.B.d., C. Costa, A.C. C., and R. Prikladinicki. *Challenges and Solutions in Distributed Software Development Project Management: a Systematic Literature Review*. in *International Conference on Global Software Development (ICGSE)*. 2010. Princeton, NJ, USA. p. 87-96.
168. Liang, X., X. Ma, Q. Yang, Y. Zhuo, B. Xu, and A. Ma. *A Virtual Human Resource Organization Model in Dual-Shore Collaborative Software Development*. in *4th International Conference on Wireless Communications, Networking and Mobile Computing, (WiCOM)*. 2008. Dalian, Liaoning (China). p. 1-5.
169. Sinha, V.S., B. Sengupta, and S. Ghosal, *An Adaptive Tool Integration Framework to Enable Coordination in Distributed Software Development*, in *International Conference on Global Software Engineering (ICGSE)*. 2007, IEEE Computer Society. p. 151-155.
170. Lee-Kelley, L. and T. Sankey, *Global virtual teams for value creation and project success: A case study.* International Journal of Project Management, 2008. **26**(1): p. 51-62.
171. Ramasubbu, N., M.S. Krishnan, and P. Kompalli, *Leveraging Global Resources: A Process Maturity Framework for Managing Distributed Development.* IEEE Softw., 2005. **22**(3): p. 80-86.
172. Gerd, H. *TAPER: A generic framework for establishing an offshore development center*. in *International Conference on Global Software Engineering (ICGSE)*. 2007. Munich, Germany. p. 162-172.
173. Kayworth, T. and D. Leidner, *The global virtual manager: a prescription for success.* European Management Journal, 2000. **18**: p. 183-194.
174. Casey, V. and I. Richardson, *The Impact of Fear on the Operation of Virtual Teams*, in *International Conference on Global Software Engineering (ICGSE)*. 2008, IEEE Computer Society: Bangalore, India. p. 163-172.
175. Deshpande, S. and I. Richardson. *Management at the Outsourcing Destination - Global Software Development in India*. in *International Conference on Global Software Engineering (ICGSE)*. 2009. Limerick, Ireland. p. 217-225.
176. Leszak, M. and M. Meier, *Successful Global Development of a Large-scale Embedded Telecommunications Product*, in *International Conference on Global Software Engineering (ICGSE)*. 2007, IEEE Computer Society: Munich, Germany. p. 23-32.

177. Petkovic, D., R. Todtenhoefer, and G. Thompson. *Teaching Practical Software Engineering and Global Software Engineering: Case Study and Recommendations*. in *Frontiers in Education Conference, 36th Annual*. 2006. San Diego, CA, USA. p. 19-24.
178. Romero, M., A. Vizcaino, and M. Piattini. *A Simulator for Education and Training in Global Requirements Engineering: A Work in Progress*. in *International Conference on Advanced Learning Technologies (ICALT)*. 2008. Santander, Spain. p. 123-125.
179. Romero, M., A. Vizcaíno, and M. Piattini, *Using Virtual Agents for the Teaching of Requirements Elicitation in GSD*, in *Proceedings of the 8th international conference on Intelligent Virtual Agents*. 2008, Springer-Verlag: Tokyo, Japan. p. 539-540.
180. Gotel, O., V. Kulkarni, M. Say, and C. Scharff. *Quality-Driven Competition: Uniting Undergraduates, Graduates and Professionals on Global Software Development Projects*. in *21st IEEE-CS Conference on Software Engineering Education and Training Workshop, 2008. CSEETW '08.* 2008. Charleston, South Carolina, USA. p. 19-21.
181. Gotel, O., V. Kulkarni, L. Neak, C. Scharff, and S. Seng, *Introducing Global Supply Chains into Software Engineering Education*, in *Software Engineering Approaches for Offshore and Outsourced Development*, S. Berlin, Editor. 2007: Heidelberg. p. 44-58.
182. Gotel, O., V. Kulkarni, M. Say, C. Scharff, and T. Sunetnanta, *A Global and Competition-Based Model for Fostering Technical and Soft Skills in Software Engineering Education*, in *22nd Conference on Software Engineering Education and Training*. 2009, IEEE Computer Society. p. 271-278.
183. Gotel, O., V. Kulkarni, C. Scharff, and L. Neak, *Students as Partners and Students as Mentors: An Educational Model for Quality Assurance in Global Software Development*, S. Berlin, Editor. 2009: Heidelberg. p. 90-106.
184. Setamanit, S.-o., W. Wakeland, and D. Raffo, *Exploring the Impact of Task Allocation Strategies for Global Software Development Using Simulation*, in *Software Process Change*, S. Berlin, Editor. 2006: Heidelberg. p. 274-285.
185. Ocker, R., D. Kracaw, S.R. Hiltz, M.B. Rosson, and L. Plotnick, *Enhancing Learning Experiences in Partially Distributed Teams: Training Students to Work Effectively Across Distances*, in *Proceedings of the 42nd Hawaii International Conference on System Sciences*. 2009, IEEE Computer Society. p. 1-10.
186. Ocker, R., M.B. Rosson, D. Kracaw, and S.R. Hiltz, *Training Students to Work Effectively in Partially Distributed Teams.* Trans. Comput. Educ., 2009. **9**(1): p. 1-24.
187. Bmegge, B., A.H. Dutoit, R. Kobylinski, and G. Teubner, *Transatlantic project courses in a university environment*, in *Proceedings of the Seventh Asia-Pacific Software Engineering Conference*. 2000, IEEE Computer Society. p. 30-37.
188. Favela, J. and F. Peña-Mora, *An Experience in Collaborative Software Engineering Education.* IEEE Softw., 2001. **18**(2): p. 47-53.
189. Honig, W.L. and T. Prasad, *A classroom outsourcing experience for software engineering learning*, in *Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education*. 2007, ACM: Dundee, Scotland. p. 181-185.
190. Braun, A., A.H. Dutoit, A.G. Harrer, and B. Brüge, *iBistro: A Learning Environment for Knowledge Construction in Distributed Software Engineering Courses*, in *Proceedings of the Ninth Asia-Pacific Software Engineering Conference*. 2002, IEEE Computer Society: Gold Coast, Australia. p. 197.
191. Adya, M., D. Nath, A. Malik, and V. Sridhar, *Bringing global sourcing into the classroom: experiential learning via software development project*, in *ACM SIGMIS CPR Conference on Computer personnel research: The global information technology workforce*. 2007, ACM: St. Louis, Missouri, USA. p. 20-27.
192. Bouillon, P., J. Krinke, and S. Lukosch, *Software Engineering Projects in Distant Teaching*, in *Proceedings of the 18th Conference on Software Engineering Education & Training*. 2005, IEEE Computer Society. p. 147-154.

193. Xiaoqing, L. *Collaborative global software development and education*. in *29th Annual International Computer Software and Applications Conference (COMPSAC)*. 2005. Edinburgh, Scotland. p. 371.
194. Lago, P., H. Muccini, and M.A. Babar, *Developing a Course on Designing Software in Globally Distributed Teams*, in *International Conference on Global Software Engineering (ICGSE)*. 2008, IEEE Computer Society: Bangalore, India. p. 249-253.
195. Setamanit, S.-o. and D. Raffo, *Identifying Key Success Factors for Globally Distributed Software Development Project Using Simulation: A Case Study*, in *Making Globally Distributed Software Development a Success Story*, S. Berlin, Editor. 2008: Heidelberg. p. 320-332.
196. Schümmer, T., S. Lukosch, and J.M. Haake, *Teaching distributed software development with the project method*, in *Conference on Computer support for collaborative learning: learning 2005: the next 10 years!* 2005, International Society of the Learning Sciences: Taipei, Taiwan. p. 577-586.
197. Richardson, I., A.E. Milewski, N. Mullick, and P. Keil, *Distributed development: an education perspective on the global studio project*, in *Proceedings of the 28th international conference on Software engineering*. 2006, ACM: Shanghai, China. p. 679-684.
198. Ahamed, S.I., *Model for Global Software Engineering Project Life Cycle and How to Use it in Classroom for Preparing Our Students for the Globalization*, in *Annual International Computer Software and Applications Conference*. 2006, IEEE Computer Society: Chicago, Illinois, USA. p. 59-60.
199. Toyoda, S., M. Miura, and S. Kunifuji, *A Case Study on Project-Management Training-Support Tools for Japanese/Chinese/Indian Offshore Development Engineers*, in *Knowledge-Based Intelligent Information and Engineering Systems*, S. Berlin, Editor. 2009: Heidelberg. p. 1222-1229.
200. Swigger, K., R. Brazile, B. Harrington, X. Peng, and F. Alpaslan. *Teaching Students How to Work in Global Software Development Environments*. in *International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*. 2006. Atlanta, GA, USA. p. 1-7.
201. Mead, N.R., A. Drommi, D. Shoemaker, and J. Ingalsbe. *A Study of the Impact on Students Understanding Cross Cultural Differences in Software Engineering Work*. in *33rd Annual IEEE International Computer Software and Applications Conference*. 2009. Seattle, Washington, USA. p. 644-645.
202. Rusu, A., A. Rusu, R. Docimo, C. Santiago, and M. Paglione, *Academia-academia-industry collaborations on software engineering projects using local-remote teams*, in *Proceedings of the 40th ACM technical symposium on Computer science education*. 2009, ACM: Chattanooga, TN, USA. p. 301-305.
203. Bondi, A.B. and J.P. Ros. *Experience with Training a Remotely Located Performance Test Team in a Quasi-agile Global Environment*. in *International Conference on Global Software Engineering (ICGSE)*. 2009. Princeton, NJ, USA. p. 254-261.
204. Berkling, K., M. Geisser, T. Hildenbrand, and F. Rothlauf, *Offshore Software Development: Transferring Research Findings into the Classroom*, in *Software Engineering Approaches for Offshore and Outsourced Development*, S. Berlin, Editor. 2007: Heidelberg. p. 1-18.
205. Hackett, M., *Building Effective Global Software Test Teams through Training*, in *IEEE International Conference on Global Software Engineering (ICGSE)*. 2007, IEEE Computer Society: Munich, Germany. p. 293-294.
206. Lutz, B., *Training for Global Software Development in an International "Learning Network"*, in *International Conference on Global Software Engineering (ICGSE)*. 2007, IEEE Computer Society: Munich, Germany. p. 140-150.
207. Swigger, K., F.N. Aplaslan, V. Lopez, R. Brazile, G. Dafoulas, and F.C. Serce, *Structural factors that affect global software development learning team performance*, in *Proceedings of the*

special interest group on management information system's 47th annual conference on Computer personnel research. 2009, ACM: Limerick, Ireland. p. 187-196.

208. Swigger, K., F.C. Serce, F.N. Alpaslan, R. Brazile, G. Dafoulas, and V. Lopez, *A Comparison of Team Performance Measures for Global Software Development Student Teams*, in *International Conference on Global Software Engineering (ICGSE)*. 2009, IEEE Computer Society: Limerick, Ireland. p. 267-274.

209. Gotel, O., V. Kulkarni, C. Scharff, and L. Neak, *Integration Starts on Day One in Global Software Development Projects*, in *International Conference on Global Software Engineering (ICGSE)*. 2008, IEEE Computer Society: Bangalore, India. p. 244-248.

210. Haake, J.M., T. Schummer, A. Haake, M. Bourimi, and B. Landgraf. *Supporting flexible collaborative distance learning in the CURE platform*. in *Hawaii International Conference on System Sciences (HICSS)*. 2004. Hawaii, USA. p. 1-10.

211. Woods, D.R., *Problem Based Learning: How to Get the Most from PBL*. 1994, Hamilton, Ontario, USA: McMaster University.

212. Cajander, Å., T. Clear, M. Daniels, J. Edlund, P. Hamrin, C. Laxer, and M. Persson. *Students analyzing their collaboration in an International Open Ended Group Project*. in *39th ASEE/IEEE Frontiers in Education Conference (FIE)*. 2009. San Antonio, Texas, USA: IEEE. p. 50-55.

213. Corder, D. and A. U, *Integrating Second Life to enhance global intercultural collaboration projects.* ACM Inroads, 2010. **1**(3): p. 43-50.

214. Clear, T. and M. Daniels. *2D & 3D Introductory Processes in Virtual Groups*. in *33rd ASEE/IEEE Frontiers in Education Conference.* 2003. Boulder, Colorado: IEEE. p. S1F1-S1F6.

215. Daniels, M., *The contribution of open ended group projects to international student collaborations.* ACM Inroads, 2010. **1**(3): p. 79-84.

216. Daniels, M., Å. Cajander, A. Pears, and T. Clear, *Engineering Education Research in Practice: Evolving Use of Open Ended Group Projects as a Pedagogical Strategy for Developing Skills in Global Collaboration (Special issue on Applications of Engineering Education Research).* International Journal of Engineering Education 2010. **26**(4): p. 795-806.

217. Cajander, Å., T. Clear, and M. Daniels. *Introducing an External Mentor in an International Open Ended Group Project*. in *39th ASEE/IEEE Frontiers in Education Conference.* 2009. San Antonio, Texas IEEE. p. T1A1-T1A6.

218. Daniels, M., M. Petre, V. Almstrum, L. Asplund, C. Bjorkmann, C. Erickson, B. Klein, M. Last, and A. Berglund. *RUNESTONE, an International Student Collaboration Project*. in *IEEE Frontiers in Education Conference (FIE)*. 1998. Tempe, Arizona: IEEE. p. 727 - 732.

219. Daniels, M., A. Berglund, and M. Petre, *Reflections on International Projects in Undergraduate CS Education.* Computer Science Education, 1999. **9**(3): p. 256-267.

220. Daniels, M. and S. Fincher, *Evaluating a joint international project in disjunct courses*, in *Proceedings of the SEFI CDWG Conference: What have they learned, SEFI Document no. 23,*. 1999. p. 139-144.

221. Braun, A., A.H. Dutoit, A.G. Harrer, and B. Brüge, *iBistro: A Learning Environment for Knowledge Construction in Distributed Software Engineering Courses*, in *Proceedings of the Ninth Asia-Pacific Software Engineering Conference*. 2002, IEEE Computer Society. p. 197-203.

222. Leigh, E. and L. Spindler, *Simulations and Games as Chaordic Learning Contexts.* Simul. Gaming, 2004. **35**(1): p. 53-69.

223. Navarro, E.O. and A.v.d. Hoek, *SimSE: an educational simulation game for teaching the Software engineering process.* SIGCSE Bull., 2004. **36**(3): p. 233-233.

224. Abran, A., P. Bourque, R. Dupuis, J.W. Moore, and L.L. Tripp, *Guide to the Software Engineering Body of Knowledge - SWEBOK*, in *IEEE Computer Society*, J.W.M. Alain Abran, Editor. 2004: Los Alamitos, California.

225. Atlee, J.M., J. Richard J. LeBlanc, T.C. Lethbridge, A. Sobel, and J.B. Thompson, *Software engineering 2004: ACM/IEEE-CS guidelines for undergraduate programs in software*

engineering, in *Proceedings of the 27th international conference on Software engineering*. 2005, ACM: St. Louis, MO, USA. p. 623-624.

226. *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*. 2008, Newtown Square, Pennsylvania, USA: Project Management Institute (PMI). 459.
227. Guzmán, J.G., J.S. Ramos, A.A. Seco, and A.S. Esteban, *How to get mature global virtual teams: a framework to improve team process management in distributed software teams.* Software Quality Control, 2010. **18**(4): p. 409-435.
228. Hawthorne, M.J. and D.E. Perry, *Software engineering education in the era of outsourcing, distributed development, and open source software: challenges and opportunities*, in *Proceedings of the 27th international conference on Software engineering*. 2005, ACM: St. Louis, MO, USA. p. 643-644.
229. Herbsleb, J.D. and A. Mockus, *An Empirical Study of Speed and Communication in Globally Distributed Software Development.* IEEE Trans. Softw. Eng., 2003. **29**(6): p. 481-494.
230. Strang, K.D., *Examining effective and ineffective transformational project leadership.* Team Performance Management, 2005. **11**(3/4): p. 68–103.
231. Niinimaki, T., A. Piri, and C. Lassenius, *Factors Affecting Audio and Text-Based Communication Media Choice in Global Software Development Projects*, in *International Conference on Global Software Engineering (ICGSE)*. 2009, IEEE Computer Society: Limerick, Ireland. p. 153-162.
232. Monasor, M.J., M. Piattini, and A. Vizcaíno, *Challenges and Improvements in Distributed Software Development: A Systematic Review.* Advances in Software Engineering, 2009: p. 1-16.
233. Nunamaker, J.F., B.A. Reinig, and R.O. Briggs, *Principles for effective virtual teamwork.* Commun. ACM, 2009. **52**(4): p. 113-117.
234. Casey, V. and I. Richardson. *Virtual software teams: Overcoming the obstacles*. in *3rd World Congress for Software Quality*. 2005. Munich, Germany.
235. Niederman, F. and F.B. Tan, *Managing global IT teams: considering cultural dynamics.* Commun. ACM, 2011. **54**(4): p. 24-27.
236. Casey, V., *Developing trust in virtual software development teams.* J. Theor. Appl. Electron. Commer. Res., 2010. **5**(2): p. 41-58.
237. Eskeli, J. and J. Maurolagoitia, *Global Software Development: Current Challenges and Solutions*, in *6th International Conference on Software and Data Technologies*. 2011: Seville, Spain. p. 29-34.
238. Jodie Eden-Jones, a., RMIT University (2004) *Work effectively with culturally diverse clients and co-workers*.
239. Clear, T., *Replicating an 'Onshore' Capstone Computing Project in a 'Farshore' Setting – an Experience Report*, in *International Conference on Global Software Engineering (ICGSE)*. 2011: Helsinki, Findland. p. 161-165.
240. Sampaio, J. and A. Moniz, *Assessing Human And Technological Dimensions In Virtual Team's Operational Competences*. 2007, University Library of Munich, Germany.
241. Strang, K.D., *Leadership substitutes and personality impact on time and quality in virtual new product development projects.* Project Management Journal, 2010. **42**(1): p. 73–90.
242. Parvathanathan, K., A. Chakrabarti, P.P. Patil, S. Sen, N. Sharma, and Y. Johng, *Global Development and Delivery in Practice: Experiences of the IBM Rational India Lab*. 2007: IBM Press.
243. Koehn, P.H. and J.N. Rosenau, *Transnational competence in an emergent epoch.* International Studies Perspectives, 2002. **3**(2): p. 105-127.
244. Nguyen, P.T., M.A. Babar, and J.M. Verner, *Critical factors in establishing and maintaining trust in software outsourcing relationships*, in *Proceedings of the 28th international conference on Software engineering*. 2006, ACM: Shanghai, China. p. 624-627.
245. Al-Ani, B., H. Wilensky, D. Redmiles, and E. Simmons, *An Understanding of the Role of Trust in Knowledge Seeking and Acceptance Practices in Distributed Development Teams*, in

International Conference on Global Software Engineering (ICGSE). 2011: Helsinki, Findland. p. 25-34.
246. Tsai, M.-T. and Y.-C. Huang, *Exploratory learning and new product performance: The moderating role of cognitive skills and environmental uncertainty.* The Journal of High Technology Management Research, 2008. **19**(2): p. 83-93.
247. Aronson, Z.H., R.R. Reilly, and G.S. Lynn, *The impact of leader personality on new product development teamwork and performance: The moderating role of uncertainty.* Journal of Engineering and Technology Management, 2006. **23**(3): p. 221-247.
248. Wong-MingJi, D.J., *Leadership Competencies for Managing Global Virtual Teams*, in *Encyclopedia of Multimedia Technology and Networking*. 2005, IGI Global. p. 519-525.
249. Shachaf, P., *Cultural diversity and information and communication technology impacts on global virtual teams: An exploratory study.* Inf. Manage., 2008. **45**(2): p. 131-142.
250. Anawati, D. and A. Craig, *Behavioral Adaptation Within Cross-Cultural Virtual Teams.* IEEE Transactions of professional communicacion, 2006. **49**(1): p. 44-56.
251. Endrass, B., E. André, L. Huang, and J. Gratch, *A data-driven approach to model culture-specific communication management styles for virtual agents*, in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*. 2010, International Foundation for Autonomous Agents and Multiagent Systems: Toronto, Canada. p. 99-108.
252. Hofstede, G., *Cultural constraints in management theories.* Academy of Management Executive, 1993. **7**(1): p. 81-94.
253. Im, S. and C. Nakata, *Crafting an environment to foster integration in new product teams.* International Journal of Research in Marketing, 2008. **25**(3): p. 164-172.
254. Trompenaars, A. and C. Hampden-Turner, *Riding the waves of culture: understanding cultural diversity in global business*. 1998: McGraw Hill.
255. Hall, E.T., *Beyond Culture*. 1976: Anchor Press. 320.
256. Hofstede, G. and G.J. Hofstede, *Cultures and organizations: software of the mind*. 2nd ed, ed. McGraw-Hill. 2005, New York, NY, USA.
257. House, R.J., P.J. Hanges, M. Javidan, P. Dorfman, and V. Gupta, *Culture, Leadership, and Organizations: The GLOBE Study of 62 Societies*. 2004, Thousand Oaks, California, USA: Sage Publications. 848.
258. Endrass, B., M. Rehm, and E. André, *Culture-specific communication management for virtual agents*, in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*. 2009, International Foundation for Autonomous Agents and Multiagent Systems: Budapest, Hungary. p. 281-287.
259. Clear, T. and D. Kassabova, *A Course in Collaborative Computing: Collaborative Learning and Research with a Global Perspective*, in *Proceedings of the 39th ACM Technical Symposium on Computer Science Education*, M. Guzdial and S. Fitzgerald, Editors. 2008, ACM: Portland, Oregon. p. 63-67.
260. Hitchcock, L., V.H. Quan, and T.C. Danh, *Intercultural competence in practice: reflections on establishing crosscultural collaborative education programmes.* ACM Inroads, 2010. **1**(3): p. 85-93.
261. John Stouby, P., *A Process for Managing Risks in Distributed Teams*, M. Lars, Editor. 2010. p. 20-29.
262. Levina, N. and A.A. Kane, *Onshore immigrant managers as boundary spanners on offshored software development projects: partners or bosses?*, in *International workshop on Intercultural collaboration*. 2009, ACM: Palo Alto, California, USA. p. 61-70.
263. Sahay, S. and G. Walsham, *Social Structure and Managerial Agency in India.* Organization Studies, 1997. **18**(3): p. 415-444.
264. Ramingwong, S. and A.S.M. Sajeev, *A Multidimensional Model for Mum Effect in Offshore Outsourcing*, in *2nd IFIP/IEEE International Symposium on Theoretical Aspects of Software Engineering*. 2008, IEEE Computer Society. p. 237-240.

265. Sussman, S.W. and L. Sproull, *Straight Talk: Delivering Bad News Through Electronic Communication.* Info. Sys. Research, 1999. **10**(2): p. 150-166.
266. Robey, D., *Blowing the whistle on troubled software projects.* Commun. ACM, 2001. **44**(4): p. 87-93.
267. Churchill, E.F. and D. Snowdon, *Collaborative virtual environments: An introductory review of issues and systems.* Virtual Reality, 1998. **3**(1): p. 3-15.
268. Hamrin, P. and M. Persson, *Exploring the Notion of Space in Virtual Collaborations : Finding Prerequisites for Success in Virtual Teams*. 2010, Uppsala University.
269. Hasler, B.S., T. Buecheler, and R. Pfeifer, *Collaborative Work in 3D Virtual Environments: A Research Agenda and Operational Framework*, in *Proceedings of the 3d International Conference on Online Communities and Social Computing: Held as Part of HCI International*. 2009, Springer-Verlag: San Diego, CA. p. 23-32.
270. Atkins, C. and M. Caukill, *Serious fun and serious learning: the Challenge of Second Life*, in *Learning and teaching in the virtual world of Second Life*. 2009, Tapir Academic Press: Trondheim, Norway. p. 79-89.
271. Szilas, N., J. Barles, and M. Kavakli, *An implementation of real-time 3D interactive drama.* Comput. Entertain., 2007. **5**(1): p. 5.
272. Drappa, A. and J. Ludewig, *Simulation in software engineering training*, in *Proceedings of the 22nd international conference on Software engineering*. 2000, ACM: Limerick, Ireland. p. 199-208.
273. Iwai, K., M. Akiyoshi, M. Samejima, and H. Morihisa, *A Situation-Dependent Scenario Generation Framework for Project Management Skill-up Simulator*, in *6th International Conference on Software and Data Technologies*. 2011: Seville, Spain. p. 408-412.
274. Jan, D., D. Herrera, B. Martinovski, D. Novick, and D. Traum, *A Computational Model of Culture-Specific Conversational Behavior*, in *Proceedings of the 7th international conference on Intelligent Virtual Agents*. 2007, Springer-Verlag: Paris, France. p. 45 - 56.
275. Last, M.Z., M. Daniels, M.L. Hause, and M.R. Woodroffe. *Learning from students: continuous improvement in international collaboration*. in *Innovation and Technology in Computer Science Education*. 2002. Aarhus, Denmark: ACM. p. 136-140.
276. Berglund, A., *Learning computer systems in a distributed project course: The what, why, how and where*, in *Acta Universitatis Upsaliensis*, D.o.t.F.o.S. Technology, Editor. 2005, Uppsala University: Uppsala. p. 233.
277. Hause, M., M. Petre, and M. Woodroffe. *Performance in International Computer Science Collaboration Between Distributed Student Teams*. in *33rd ASEE/IEEE Frontiers in Education Conference*. 2003. Boulder, Colorado: IEEE. p. S1F13- S1F18.
278. Clear, T. *Virtual groups, Collaborative Trials and Tribulations*. in *ED-MEDIA 2003 World Conference On Educational Multimedia, Hypermedia & Telecommunications*. 2003. Honolulu, Hawaii: AACE. p. 627 - 630.
279. Clear, T., *Exploring the notion of 'cultural fit' in global virtual collaborations.* ACM Inroads, 2010. **1**(3): p. 58-65.
280. Clear, T., *Developing and Implementing a Groupware Application to Support International Collaborative Learning*, in *Management Science and Information Systems*. 2000, Auckland University: Auckland. p. 1-331.
281. Clear, T. *Global Virtual Teams: Moderating Behaviour in 3D Collaborative Virtual Environments*. in *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (ELEARN)*. 2004. Washington, D.C.: AACE. p. 2321 - 2326.
282. Clear, T. and G. Foot. *Avatars in Cyberspace - A Java 3D Application to Support Performance of Virtual Groups*. in *Innovation and Technology in Computer Science Education*. 2002. Aarhus, Denmark: ACM. p. 222.

283. Sagae, A., W.L. Johnson, and R. Row, *Serious game environments for language and culture education*, in *Proceedings of the NAACL HLT Demonstration Session*. 2010, Association for Computational Linguistics: Los Angeles, California. p. 29-32.
284. Mandl-Striegnitz, P., *How to successfully use software project simulation for educating software project managers*, in *Proceedings of the Frontiers in Education Conference*. 2001, IEEE Computer Society. p. T2D-19-24.
285. Kirschner, P.A., J. Sweller, and R.E. Clark, *Why minimal guidance during instruction does not work: an analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching.* Educational Psychologist, 2006. **41**(2): p. 75-86.
286. Core, M., D. Traum, H.C. Lane, W. Swartout, J. Gratch, M.V. Lent, and S. Marsella, *Teaching Negotiation Skills through Practice and Reflection with Virtual Humans.* Simulation, 2006. **82**(11): p. 685-701.
287. Cassell, J., T. Bickmore, M. Billinghurst, L. Campbell, K. Chang, H. Vilhjálmsson, and H. Yan. *Embodiment in conversational interfaces: Rea*. in *Proceedings of the SIGCHI conference on Human factors in computing systems*. 1999: ACM. p. 520-527.
288. Bevacqua, E. and C. Pelachaud, *Expressive audio-visual speech: Research Articles.* Comput. Animat. Virtual Worlds, 2004. **15**(3-4): p. 297-304.
289. Iacobelli, F. and J. Cassell, *Ethnic Identity and Engagement in Embodied Conversational Agents*, in *Intelligent Virtual Agents*, C. Pelachaud, et al., Editors. 2007, Springer Berlin Heidelberg. p. 57-63.
290. Kavakli, M., T. Rudra, and M. Li, *An Embodied Conversational Agent for Counselling Aborigines*, in *6th International Conference on Software and Data Technologies*. 2011: Seville, Spain. p. 371-376.
291. Pelachaud, C., *Studies on gesture expressivity for a virtual agent.* Speech Commun., 2009. **51**(7): p. 630-639.
292. Wohlin, C., M. Höst, and K. Henningsson, *Empirical Research Methods in Software Engineering*, in *Empirical Methods and Studies in Software Engineering*. 2003, Springer Berlin Heidelberg. p. 7-23.
293. Harrison, R., N. Badoo, E. Barry, S. Biffl, A. Parra, B. Winter, and J. Wuest, *Directions and Methodologies for Empirical Software EngineeringResearch.* Empirical Softw. Engg., 1999. **4**(4): p. 405-410.
294. Johnson, R.B., A.J. Onwuegbuzie, and L.A. Turner, *Toward a Definition of Mixed Methods Research.* Journal of Mixed Methods Research, 2007. **1**(2): p. 112-133.
295. Denzin, N.K. and Y.S. Lincoln, *The SAGE Handbook of Qualitative Research*. 2005: Sage Publications.
296. Potter, W.J., *An Analysis of Thinking and Research about Qualitative Methods*. 1996: L. Erlbaum Associates.
297. Shaw, I., *Qualitative Evaluation*. 1999: SAGE Publications.
298. Beecham, S., *A Requirements-based Software Process Maturity Model*. 2003, University of Hertfordshire.
299. Seaman, C.B., *Qualitative Methods in Empirical Studies of Software Engineering.* IEEE Trans. Softw. Eng., 1999. **25**(4): p. 557-572.
300. Walker, R.J., L.C. Briand, D. Notkin, C.B. Seaman, and W.F. Tichy, *Panel: empirical validation: what, why, when, and how*, in *Proceedings of the 25th International Conference on Software Engineering*. 2003, IEEE Computer Society: Portland, Oregon. p. 721-722.
301. Drehmer, D.E. and S.M. Dekleva, *A note on the evolution of software engineering practices.* Journal of Systems and Software, 2001. **57**(1): p. 1-7.
302. Miller, R.L., J. Maltby, D.A. Fullerton, and C. Acton, *SPSS for Social Scientists*. 2002: Palgrave Global Publishing. 352.

303. Thota, N., A. Berglund, and T. Clear, *Illustration of paradigm pluralism in computing education research*, in *Fourteenth Australasian Computing Education Conference*. 2012, Australian Computer Society, Inc.: Melbourne, Australia. p. 103-112.
304. Orlikowski, W.J. and J.J. Baroudi, *Studying Information Technology in Organizations: Research Approaches and Assumptions.* Information Systems Research, 1991. **2**(1): p. 1-28.
305. Clark, V.L.P., *The Adoption and Practice of Mixed Methods: U.S. Trends in Federally Funded Health-Related Research.* Qualitative Inquiry, 2010. **16**(6): p. 428-440.
306. Kitchenham, B.A., S.L. Pfleeger, L.M. Pickard, P.W. Jones, D.C. Hoaglin, K.E. Emam, and J. Rosenberg, *Preliminary guidelines for empirical research in software engineering.* IEEE Trans. Softw. Eng., 2002. **28**(8): p. 721-734.
307. Lethbridge, T.C., *What Knowledge Is Important to a Software Professional?* Computer, 2000. **33**(5): p. 44-50.
308. Lohr, S.L., *Sampling: Design and Analysis: Design And Analysis*. 2010: Brooks/Cole.
309. Hakim, C., *Research Design: Strategies and Choices in the Design of Social Research*. 1987: Allen & Unwin.
310. Hetzel, W.C., *The sorry state of software practice measurement and evaluation.* Journal of Systems and Software, 1995. **31**(2): p. 171-179.
311. Krosnick, J.A., *Survey research.* Annual review of psychology, 1999. **50**(1): p. 537-567.
312. Dybå, T., *An Instrument for Measuring the Key Factors of Successin Software Process Improvement.* Empirical Softw. Eng., 2000. **5**(4): p. 357-390.
313. Greenhalgh, T. and R. Taylor, *Papers that go beyond numbers (qualitative research)*. BMJ 315: 740-743. 1997.
314. Creswell, J.W., *Research design: qualitative & quantitative approaches*. 1994: Sage Publications.
315. Tesch, R., *Qualitative Research: Analysis Types and Software Tools*. 1990: Falmer Press.
316. Braun, V. and V. Clarke, *Using thematic analysis in psychology.* Qualitative Research in Psychology, 2006. **3**(2): p. 77-101.
317. Boyatzis, R.E., *Transforming Qualitative Information: Thematic Analysis and Code Development*. 1998: SAGE Publications.
318. Guest, G., K.M. MacQueen, and E.E. Namey, *Applied Thematic Analysis*. 2011: SAGE Publications.
319. Saldana, J., *The Coding Manual for Qualitative Researchers*. 2012: SAGE Publications.
320. Kenney, J.F. and E.S. Keeping, *Mathematics of statistics*. 1947: Van Nostrand.
321. Goodman, S.N., *Toward Evidence-Based Medical Statistics. 1: The P Value Fallacy.* Annals of Internal Medicine, 1999. **130**(12): p. 995-1004.
322. Patel, J.K. and C.B. Read, *Handbook of the Normal Distribution, Second Edition*. 1996: Taylor & Francis.
323. Shapiro, S.S. and M.B. Wilk, *An analysis of variance test for normality (complete samples).* Biometrika, 1965. **52**(3-4): p. 591-611.
324. Gravetter, F.J. and L.A.B. Forzano, *Research Methods for the Behavioral Sciences (Gravetter)*. 2010: Wadsworth.
325. Tukey, J.W., *Exploratory Data Analysis*. 1977: Addison-Wesley Publishing Company.
326. Sprent, P. and N.C. Smeeton, *Applied Nonparametric Statistical Methods*. 2007: Chapman & Hall/CRC.
327. Mann, H.B. and D.R. Whitney, *On a Test of Whether One of Two Random Variables is Stochastically Larger Than the Other*. 1947: Institute of Mathematical Statistics.
328. Siegel, S. and N.J. Castellan, *Nonparametric statistics for the behavioral sciences*. 1988: McGraw-Hill.
329. Kanji, G.K., *100 Statistical Tests*. 1999: SAGE Publications.
330. Trochim, W.M.K. and J.P. Donnelly, *The Research methods knowledge base*. 2008: Atomic Dog/Cengage Learning.

331. Seehusen, S., C. Lecon, and C. Kaben. *Specification of learning trails in virtual courses*. in *Frontiers in Education Conference (FIE)*. 2000. Kansas city, Missouri, USA. p. S3D11-S3D15.
332. Leidner, D.E. and S.L. Jarvenpaa, *The use of information technology to enhance management school education: a theoretical view.* MIS Q., 1995. **19**(3): p. 265-291.
333. Olina, Z. and H. Sullivan, *Student self-evaluation, teacher evaluation, and learner performance.* Educational Technology Research and Development, 2004. **52**(3): p. 5-22.
334. O'Neill, G., S. Huntley-Moore, and P. Race, *Case Studies of Good Practices in Assessment of Student Learning in Higher Education*. 2007: AISHE. 162.
335. Alexander, C., *A pattern language: towns, buildings, construction*. Vol. 2. 1977: Oxford University Press, USA.
336. Coplien, J.O. and N.B. Harrison, *Organizational Patterns of Agile Software Development*. 2005: Prentice Hall.
337. Hargreaves, E., D. Damian, F. Lanubile, and J. Chisan, *Global software development: building a research community.* SIGSOFT Softw. Eng. Notes, 2004. **29**(5): p. 1-5.
338. Hoyle, R.H., M.J. Harris, and C.M. Judd, *Research Methods and Social Relations*. 8 ed. 2009: Wadsworth Publishing.
339. Beecham, S., N. Carroll, and J. Noll, *A Decision Support System for Global Team Management: Expert Evaluation*, in *REMIDI - International Workshop on Tool Support Development and Management in Distributed Software Projects co-located with 7th IEEE International Conference on Global Software Engineering (ICGSE)*. 2012, IEEE Computer Society: Porto Alegre, Brazil. p. 12 - 17.
340. Holzinger, A., *Usability engineering methods for software developers.* Commun. ACM, 2005. **48**(1): p. 71-74.
341. Hwang, W. and G. Salvendy, *Number of people required for usability evaluation: the 10±2 rule.* Commun. ACM. **53**(5): p. 130-133.
342. Nielsen, J., *Enhancing the explanatory power of usability heuristics*, in *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*. 1994, ACM: Boston, Massachusetts, United States. p. 152-158.
343. Jones, M.J., *Just-in-time training.* Advances in Developing Human Resources, 2001. **3**(4): p. 480-487.
344. Wallace, R.S., *The Anatomy of A.L.I.C.E*, in *Parsing the Turing Test*, S. Netherlands, Editor. 2008. p. 181-210.
345. Clear, T. *International Collaborative Learning - The Facilitation Process*. in *World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA)*. 1999. Seattle, Washington: AACE. p. 1759-1764.
346. Baldwin, J. and D. Damian, *Tool Usage within a Globally Distributed Software Development Course and Implications for Teaching* in *3rd International Workshop on Collaborative Teaching of Globally Distributed Software Development (CTGDSD)*. 2013: San Francisco, CA, USA. p. 15-19.
347. Craig, S., *Cross-Cultural Dialogues: Seventy-Four Brief Encounters with Cultural Difference*. 1994: Intercultural Press, Incorporated.
348. Cheng, W., *Intercultural Conversation*. 2003: John Benjamins Publishing Company.
349. Warren, T.L. *National cultures in international communication*. in *International Professional Communication Conference (IPCC)*. 1998. p. 305-312.
350. Garcia, O. and R. Otheguy, *English Across Cultures: Cultures Across English : A Reader in Cross Cultural Communication*. Contributions to the Sociology of Language [CSL] Series, ed. Otheguy. 1989, Berlin, New-York: Mouton de Gruyter. 492.
351. Axtell, R.E., *Do's and Taboos of Using English Around the World*. 1995, Canada: Wiley. 224.
352. Grzega, J., *Reflection on Concepts of English for Europe: British English, American English, Euro-English, Global English.* Journal for EuroLinguistiX 2, 2005: p. 44-64.

353. Sargent, R.G. *Verification, validation and accreditation of simulation models*. in *Simulation Conference, 2000. Proceedings. Winter*. 2000. p. 50-59 vol.1.
354. Carson, J.S., *Convincing users of model's validity is challenging aspect of modeller's job.* Industrial engineering, 1986. **18**(6): p. 74 - 85.
355. Robinson, S., *Simulation model verification and validation: increasing the users' confidence*, in *Proceedings of the 29th conference on Winter simulation*. 1997, IEEE Computer Society: Atlanta, Georgia, USA.
356. Davis, P.K., *Generalizing Concepts and Methods of Verification, Validation, and Accreditation (VV&A) for Military Simulations*. R-4249-ACQ. 1992: RAND Corporation.
357. Gass, S.I., *Decision-Aiding Models: Validation, Assessment, and Related Issues for Policy Analysis.* Operations Research, 1983. **31**(4): p. 603-631.
358. Mandinach, E.B. (2005) *The Development of Effective Evaluation Methods for E-Learning: A Concept Paper and Action Plan*. Teachers College Record **107**, 1814-1835.
359. Mikropoulos, T.A. and A. Natsis, *Educational virtual environments: A ten-year review of empirical research (1999-2009).* Computers & Education, 2011. **56**(3): p. 769-780.
360. Wilson, K. and J. Fowler, *Assessing the impact of learning environments on students' approaches to learning: comparing conventional and action learning designs.* Assessment & Evaluation in Higher Education, 2005. **30**(1): p. 87-101.
361. Akar, E., E. Öztürk, B. Tunçer, and M. Wiethoff, *Evaluation of a collaborative virtual learning environment.* Education + Training, 2004. **46**(6/7): p. 343 - 352.
362. Vizcaíno, A., *Enhancing collaborative learning using a simulated student agent*. 2001, University of Castilla-La Mancha: Ciudad Real. p. 271.
363. Charness, G., U. Gneezy, and M.A. Kuhn, *Experimental methods: Between-subject and within-subject design.* Journal of Economic Behavior & Organization, 2012. **81**(1): p. 1-8.
364. Creswell, J.W. and V.L.P. Clark, *Designing and Conducting Mixed Methods Research*. 2007: SAGE Publications.
365. Sauro, J. and J.R. Lewis, *Quantifying the User Experience: Practical Statistics for User Research*. 2012: Elsevier Science.
366. ISO, *ISO/IEC 25022 - Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Measurement of quality in use*. 2013.
367. Nations, U., *Training Manual on Disability Statistics*. 2008: World Health Organization.
368. *Cape, P.*, *Questionnaire Length, Fatigue Effects and Response Quality Revisited*, in *ARF Re:think*. 2010, Survey Sampling International.
369. Marsden, P.V. and J.D. Wright, *Handbook of Survey Research*. 2010: Emerald Group Publishing.
370. McCarney, R., J. Warner, S. Iliffe, R. van Haselen, M. Griffin, and P. Fisher, *The Hawthorne Effect: a randomised, controlled trial.* BMC Medical Research Methodology, 2007. **7**(1): p. 30.
371. Kirkpatrick, D.L., *Evaluating Training Programs: The Four Levels*. 1998: Berrett-Koehler.
372. Fransson, A., *On Qualitative Differences in Learning: IV--Effects of Intrinsic Motivation and Extrinsic Test Anxiety on Processes and Outcome.* British Journal of Educational Psychology, 1977. **47**(3): p. 244-257.
373. Monasor, M.J., A. Vizcaíno, M. Piattini, and I. Caballero, *Preparing students and engineers for Global Software Development: A Systematic Review*, in *International Conference on Global Software Development (ICGSE)*. 2010, IEEE Computer Society: Princeton, NJ, USA. p. 177-186.
374. Clear, T., *Supporting the Work of Global Virtual Teams: The Role of Technology-Use Mediation*, in *Computing and Mathematical Sciences* 2008, Auckland University of Technology: Auckland. p. 1-473.
375. Bansler, J.r.P. and E. Havn, *Sensemaking in Technology-Use Mediation: Adapting Groupware Technology in Organizations.* Computer Supported Cooperative Work (CSCW), 2006. **15**(1): p. 55-91.
376. Michiel van, G., *Getting Real in the Classroom*, V. Doug, Editor. 2007. p. 106-107.

377. Lacity, M.C. and J.W. Rottman, *The Impact of Outsourcing on Client Project Managers.* Computer, 2008. **41**(1): p. 100-102.
378. Carmel, E. and P. Abbott, *Why 'nearshore' means that distance matters.* Commun. ACM, 2007. **50**(10): p. 40-46.
379. Elliott, M.S. and W. Scacchi, *Free software developers as an occupational community: resolving conflicts and fostering collaboration*, in *International ACM SIGGROUP conference on Supporting group work*. 2003, ACM: Sanibel Island, Florida, USA. p. 21 - 30.
380. Meneely, A. and L. Williams, *Secure open source collaboration: an empirical study of linus' law*, in *Proceedings of the 16th ACM conference on Computer and communications security*. 2009, ACM: Chicago, Illinois, USA.
381. Dustdar, S., *Caramba - A Process-Aware Collaboration System Supporting Ad hoc and Collaborative Processes in Virtual Teams.* Distrib. Parallel Databases, 2004. **15**(1): p. 45-66.
382. Redmiles, D., H. Wilensky, K. Kosaka, and R.d. Paula, *What ideal end users teach us about collaborative software*, in *International ACM SIGGROUP conference on Supporting group work*. 2005, ACM: Sanibel Island, Florida, USA. p. 260 - 263.
383. Thorpe, S., *The Use of Story in Building Online Group Relationships*, in *Handbook of Research on Computer Mediated Communication*. 2008, IGI Global. p. 551-569.
384. Coppola, N.W., S.R. Hiltz, and N. Rotter. *Building trust in virtual teams*. in *International Professional Communication Conference (IPCC)*. 2001. p. 353-366.
385. Baocong, J., W. Hongyun, A. Sufang, and F. Haiguang. *Research on distance collaborative activities for teacher education based on online video and cloud computing environment*. in *International Conference on Computer Science & Education (ICCSE)*. 2011. p. 180-185.
386. Yager, S.E., *Using information technology in a virtual work world: characteristics of collaborative workers*, in *ACM SIGCPR conference on Computer personnel research*. 1999, ACM: New Orleans, Louisiana, USA. p. 73-78.
387. Poole, M.J., *Developing online communities of practice in preservice teacher education*, in *Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community*. 2002, International Society of the Learning Sciences: Boulder, Colorado. p. 579-580.
388. Mansour, O., M. Abusalah, and L. Askenäs, *Wiki-based community collaboration in organizations*, in *Proceedings of the 5th International Conference on Communities and Technologies*, ACM: Brisbane, Australia. p. 79-87.
389. Bramson, R.M., *Coping with Difficult People*. 1988: Dell.
390. Välimäki, A., S. Vesiluoma, and K. Koskimies, *Scenario-Based Assessment of Process Pattern Languages*, in *Product-Focused Software Process Improvement*. 2009, Springer Berlin Heidelberg. p. 246-260.
391. Monasor, M.J., A. Vizcaíno, and M. Piattini, *Cultural and linguistic problems in GSD: a simulator to train engineers in these issues.* Journal of Software Maintenance and Evolution: Research and Practice (Special Issue on Global Software Engineering), 2011. **24**(6): p. 707-717.
392. Monasor, M.J., A. Vizcaíno, and M. Piattini, *Improving Distributed Software Development in Small and Medium Enterprises.* The Open Software Engineering Journal, 2009: p. 26-37.
393. Monasor, M.J., A. Vizcaíno, M. Piattini, J. Noll, and S. Beecham, *Simulating Global Software Development processes for use in Education: A Feasibility Study*, in *20th European Conference, EuroSPI*, F. McCaffery, R.V. O´Connor, and R. Messnarz, Editors. 2013, Springer: Dundalk, Ireland. p. 36-47.
394. Monasor, M.J., A. Vizcaíno, and M. Piattini, *Providing Training in GSD by Using a Virtual Environment*, in *International Conference on Product-Focused Software Development and Process Improvement (PROFES)*. 2012: Madrid, Spain. p. 203-217.
395. Monasor, M.J., A. Vizcaíno, and M. Piattini, *VENTURE: Towards a framework for simulating GSD in educational environments*, in *5th International Conference on Research Challenges in Information Science (RCIS)*. 2011: Gosier, Guadeloupe, France. p. 469 - 478.

396. Monasor, M.J., A. Vizcaino, and M. Piattini, *A Framework for Defining Simulators with Which to Train Global Software Development*, in *International Conference on Software and Data Technologies (ICSOFT)*. 2011: Seville, Spain. p. 261-264.
397. Monasor, M.J., A. Vizcaíno, and M. Piattini. *Training Global Software Development skills through a simulated environment*. in *5th International Conference on Software and Data Technologies (ICSOFT)*. 2010. Athens, Greece. p. 271-274.
398. Garrido, P.J., A. Vizcaíno, J. Andrada, M.J. Monasor, and M. Piattini, *A Tool for Decisions Management in Distributed Software Projects*, in *6th REMIDI Workshop in the International Conference on Global Software Engineering Workshops (ICGSE)*. 2012: Porto Alegre (Brazil). p. 22-27.
399. Monasor, M.J., A. Vizcaino, and M. Piattini, *An Architecture for Creating Simulators for Training Global Software Development*, in *International Conference on Global Software Development (ICGSE)*. 2011: Helsinki, Finland. p. 90 - 94.
400. Monasor, M.J., A. Vizcaíno, and M. Piattini, *A Framework for Training Skills for Global Software Development*, in *International Conference on Global Software Development (ICGSE)*. 2010: Princeton, NJ, USA. p. 355-356.
401. Monasor, M.J., A. Vizcaíno, and M. Piattini. *An educational environment for learning Global Software Development*. in *10th IEEE International Conference on Advanced Learning Technologies (ICALT)*. 2010. Sousse, Tunisia. p. 99 - 101.
402. Monasor, M.J., A. Vizcaíno, and M. Piattini. *A training tool for Global Software Development*. in *9th International Conference on Information Technology Based Higher Education and Training (ITHET)*. 2010. Cappadocia, Turkey. p. 9-16.
403. Monasor, M.J., A. Vizcaíno, M. Piattini, J. Noll, and S. Beecham, *Towards a Global Software Development Community Web: Identifying Patterns and Scenarios*, in *PARIS Workshop, International Conference on Global Software Development (ICGSE)*. 2013: Bari, Italy. p. 41-46.
404. Monasor, M.J., A. Vizcaino, and M. Piattini, *Preparing Students and Engineers for Global Software Development: An Architecture Based on Simulation*, in *Encyclopedia of Software Engineering*. 2012. p. 117-186.
405. Monasor, M.J., M. Piattini, and A. Vizcaíno, *A Systematic Review of Distributed Software Development: Problems and Solutions*, in *Handbook of Research on Software Engineering and Productivity Technologies: Implications of Globalization*. 2010, IGI Global. p. 209-225.
406. Monasor, M., A. Vizcaíno, and M. Piattini, *A Tool for Training Students and Engineers in Global Software Development Practices*, in *Collaboration and Technology*, G. Kolfschoten, T. Herrmann, and S. Lukosch, Editors. 2010, Springer Berlin Heidelberg. p. 169-184.
407. Jiménez, M. and M. Piattini, *Problems and Solutions in Distributed Software Development: A Systematic Review*, in *Software Engineering Approaches for Offshore and Outsourced Development*, K. Berkling, et al., Editors. 2009, Springer Berlin Heidelberg. p. 107-125.
408. Monasor, M.J., A. Vizcaíno, and M. Piattini, *Docencia en Desarrollo Global de Software: Una revisión Sistemática*, in *XVII Jornadas de Enseñanza Universitaria de Informática*. 2011: Seville, Spain. p. 221-230.
409. Monasor, M.J., A. Vizcaino, and M. Piattini, *VENTURE: Hacia un entorno para el entrenamiento del Desarrollo Global de Software*, in *XVI Jornadas de Ingeniería del Software y Bases de Datos (JISBD)*. 2011: A Coruña (Spain). p. 1065-1070.
410. Monasor, M.J., A. Vizcaíno, M. Piattini, J. Noll, and S. Beecham, *Evaluation of a simulation-based platform for training Global Software Development*, in *Lero-TR-2013-02*. 2013, Lero, The Irish Software Engineering Research Centre. University of Limerick: Limerick.
411. Heo, M., *A learning and assessment tool for web-based distributed education*, in *Proceedings of the 4th conference on Information technology curriculum*. 2003, ACM: Lafayette, Indiana, USA. p. 151 - 154.
412. Huizinga, D.M., *Identifying topics for instructional improvement through on-line tracking of programming assignments*, in *Proceedings of the 6th annual conference on Innovation and*

technology in computer science education. 2001, ACM: Canterbury, United Kingdom. p. 129-132.

413. Saikkonen, R., L. Malmi, and A. Korhonen, *Fully automatic assessment of programming exercises*, in *Proceedings of the 6th annual conference on Innovation and technology in computer science education*. 2001, ACM: Canterbury, United Kingdom. p. 133-136.
414. Bowyer, J. and J. Hughes, *Assessing undergraduate experience of continuous integration and test-driven development*, in *International Conference on Software Engineering (ICSE)*. 2006, ACM: Shanghai, China. p. 691-694.
415. Black, P., C. Harrison, C. Lee, B. Marshall, and D. Wiliam, *Assessment for Learning: Putting it into Practice*. 2003: Open University Press. 152.
416. Sung, Y.-T., K.-E. Chang, S.-K. Chiou, and H.-T. Hou, *The design and application of a web-based self- and peer-assessment system.* Comput. Educ., 2005. **45**(2): p. 187-202.
417. Thompson, E., A. Luxton-Reilly, J.L. Whalley, M. Hu, and P. Robbins, *Bloom's taxonomy for CS assessment*, in *Proceedings of the tenth conference on Australasian computing education - Volume 78*. 2008, Australian Computer Society, Inc.: Wollongong, NSW, Australia. p. 155-161.
418. Anderson, L.W., D.R. Krathwohl, and B.S. Bloom, *A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives*. 2001: Longman.
419. Bloom, B.S., *Taxonomy of Educational Objectives Book 1: Cognitive Domain*. 2 ed. 1984: Addison-Wesley.
420. Alaoutinen, S. and K. Smolander, *Student self-assessment in a programming course using bloom's revised taxonomy*, in *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education*. 2010, ACM: Bilkent, Ankara, Turkey. p. 155-159.
421. Urquiza-Fuentes, J. and J.Á. Velázquez-Iturbide, *Comparing the effectiveness of different educational uses of program animations*, in *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education*. 2012, ACM: Haifa, Israel. p. 174-179.
422. Davis, S.L. and S.J. Finney, *Examining the psychometric properties of the Cross-Cultural Adaptability Inventory.* Educational and Psychological Measurement, 2006. **66**: p. 318-330.
423. Rosinski, P., *Coaching Across Cultures: New Tools for Leveraging National, Corporate, and Professional Differences*. 2003: Nicholas Brealey Publishing. 296.
424. Van Der Zee, K.I. and U. Brinkmann, *Construct Validity Evidence for the Intercultural Readiness Check against the Multicultural Personality Questionnaire.* International Journal of Selection and Assessment, 2004. **12**(3): p. 285-290.
425. Karlgren, K. and R. Ramberg, *The Use of Design Patterns in Overcoming Misunderstandings in Collaborative Interaction Design.* CoDesign - International Journal of CoCreation in Design and the Arts, 2012. **8**(4): p. 231-246.
426. Gamma, E., R. Helm, R. Johnson, and J. Vlissides, *Design Patterns CD: Elements of Reusable Object-Oriented Software* Vol. 1. 1998: Addison-Wesley Professional.
427. Sanchez-Segura, M.-I., F. Medina-Dominguez, A. de Amescua, and A. Mora-Soto, *Improving the efficiency of use of software engineering practices using product patterns.* Information Sciences, 2010. **180**(14): p. 2721-2742.
428. Bergin, J., J. Eckstein, M.L. Manns, H. Sharp, J. Chandler, K. Marquardt, E. Wallingford, M. Sipos, and M. Völter, *Pedagogical Patterns: Advice for Educators*. 2012: Software Tools.
429. Jorrín-Abellán, I.M., I. Ruiz-Requies, D. Hernández Leo, E.D. Villasclaras-Fernández, Y. Dimitriadis, B. Rubia-Avi, and J.I. Asensio-Pérez, *Collage, a Collaborative Learning Design Editor Based on Patterns.* Educational Technology & Society, 2006. **9**(1): p. 58-71.
430. Laplante, P.A. and C.J. Neill, *AntiPatterns: Identification, Refactoring, and Management*. 2006: Auerbach Publications.

431. MacGregor, E., Y. Hsieh, and P. Kruchten, *Cultural patterns in software process mishaps: incidents in global projects*, in *Workshop on Human and social factors of software engineering*. 2005, ACM: St. Louis, Missouri. p. 1-5.
432. Paasivaara, M., C. Lassenius, and P. Pyysiainen, *Communication Patterns and Practices in Software Development Networks*. 2003, Helsinki, Finland: Helsinki University of Technology. p. 783-798.
433. Välimäki, A., J. Kääriäinen, and K. Koskimies, *Global Software Development Patterns for Project Management*, in *Software Process Improvement*. 2009, Springer Berlin Heidelberg. p. 137-148.
434. Pehmöller, A., F.S. Capgemini, and S. Wagner, *Patterns for testing in global software development*, in *13th International Conference on Quality Engineering in Software Technology*. 2010.
435. Salger, F., J. Englert, and G. Engels. *Towards Specification Patterns for Global Software Development Projects - Experiences from the Industry*. in *Quality of Information and Communications Technology (QUATIC), 2010 Seventh International Conference on the*. p. 73-78.

Appendix I: Literature review on assessment methods

The effectiveness of educational training environments has been assessed in various ways [411], [412], [413]. Minimizing the instructors' workload is a common requirement in these systems, although not all aspects of assessment can be automated. As an example, a semi-automatic web-based assessment tool has been presented by Heo [411] with the aim of helping instructors comment directly on students' submissions, provide students with access to this information, and measure their learning through quizzes.

In a comparable study, Bowyer and Hughes [414] describe an assessment with students using a set of process measures that examine their participation, ability to perform testing activities and their reflection on the process. The authors gathered these measures by examining several sources: logs from a development environment, records of e-portfolios and the version control history of a version control system.

Peer- and self-assessment is often used to track those learning processes in which instructors delegate the responsibility of assessment to their students [415]. This type of assessment also plays a role in the students' motivation by giving them goals to achieve and a means to follow their progress. Some researchers have observed that students' results improve when they evaluate their own work [333]. Web-based tools have been developed in various fields with the aim of providing peer and self-assessment [416].

Bloom's Taxonomy has been widely used for assessment purposes in varying disciplines, as the scale is shown to be relatively easy to adapt to specific topics [417]. The systematic guidelines provided make the assessment process reliable, by avoiding subjectivity. The Revised Bloom's Taxonomy [418] provides a more in-depth assessment by describing, measuring, and classifying learning objectives. It contains six levels of thinking skills: remembering, understanding, applying, analyzing, evaluating, and creating. The mastery of basic levels is required before progressing to higher levels of training. These levels are ranked on a scale of increasing complexity, in which higher levels integrate with previous levels.

Assessment in Software Engineering education

Most of the current research in Software Engineering education is focused on the original Bloom's Taxonomy [419]. As an example, Alaoutinen and Smolander [420] present a self-assessment tool intended for a programming course based on a survey questionnaire that uses Bloom's Taxonomy as the basis for its scale. Learning outcomes are used to set goals for students, thus providing teachers with an objective basis on which to measure the level of knowledge gained by students.

Urquiza and Velázquez [421] also applied the traditional taxonomy in the field of program animations. At the beginning of the course, the students were informed about the evaluation. Their knowledge acquisition was then measured by means of three knowledge tests of different levels of complexity (low, medium and high). The questions considered were mapped onto Bloom's knowledge, understanding, application, analysis and synthesis categories.

The main difficulty with these approaches is assessing the students' improvement, as the scores are not linked to previous knowledge and skills. Some methods deal with this problem by assessing the students several times during the course [420]. The research in this field therefore shows the need to assess students before and after training in order to measure their progress.

Assessment in GSD training

Despite the lack of studies on assessment in the GSD field, Damian et al. [12] report on an experience of a GSD course conducted in collaboration with three universities. The course emphasized requirements management activities through synchronous computer-mediated relationships, and incorporated the following parameters in order to assess GSD skills [12]:

- International teamwork:
  - Project goals: by examining project outcomes.
  - Learning cultural diversity skills: students recognized cultural differences by means of discussions.
  - Shared understanding of software features: by means of a questionnaire.
  - Sense of classroom community: students responded to a questionnaire measuring their feelings of connectedness.
- Distributed project management:
  - Students' perception as regards the effectiveness of coordination strategies and negotiation activities.
  - Number of issues resolved during project meetings.
  - Students' participation in class.
  - Logs of students' communication events.
  - Reflection on teamwork by means of reports.
- Computer mediated communications:
  - Use of computer-mediated communication tools: students were asked to rate the effectiveness of the tool.
- Ambiguity/uncertainty:
  - Tracking the number of ambiguous issues resolved.

Damian et al. [12] applied these metrics (in which each category was weighted according to its importance), to produce a final score.

Assessment in multi-cultural training

In the related field of multicultural training, there are tools that are oriented towards providing multinational corporations and expatriates with training. Cross Cultural Adaptability Inventory (CCAI) [422] is an online tool that addresses a student's ability to adapt to any culture. By means of a 50 question test, it provides self-assessment on cross-cultural effectiveness, covering four areas: emotional resilience, personal autonomy, flexibility/openness and perceptual acuity. A feedback report plots the scores on a graph, thus enabling users to discover their weak and strong points. A guide helps participants understand the implications of their scores and suggests actions that can help them strengthen their skills.

The Cultural Orientations Framework (COF) [423] is a framework that is used to assess and compare cultures based on the following cultural dimensions: sense of power and responsibility, time management approaches, definitions of identity and purpose, organizational arrangements, notions of territory and boundaries, communication patterns, and modes of thinking. By means of a questionnaire, COF determines the individual's cultural profile, whose validity is checked in further steps of the procedure. The Intercultural Readiness Check (IRC) [424] also makes use of a similar concept.

Assessment of team performance

Finally, the performance of virtual teams has also been assessed in GSD fields that are not directly related to education. Petkovic et al. [11] propose practical Software Engineering teamwork assessment methods based on comparison measurements. They measure and analyze: *quality of final deliverables, expended effort, level of collaborative activity, teamwork problems*. This work particularly focuses on the quality of deliverables, and the focus and measurements could also be applied to assessments in educational environments.

Appendix II: Literature review on GSD patterns

Sources of Global Software Development (GSD) information, such as academic literature, often focus on high-level issues rather than on specific problems. Researchers tend to generalize problems and solutions; however, practitioners and instructors frequently need to identify real low-level scenarios and patterns in an effort to study specific problems and their solutions.

During the process of developing simulations it was found that simulating accurate GSD scenarios requires gaining access to real problems that could be reproduced. What is more, the kinds of scenarios that are required, based on cultural, linguistic and procedural problems in GSD, are perhaps too specific to be reported in literature. Gaining access to the type of low-level, realistic problems and scenarios needed for a simulation-based training platform therefore implies a challenge.

Moreover, this problem is not restricted to the educational field. Researchers could also benefit from having access to real GSD scenarios, along with information originating from previous experiences. Companies could also identify and classify common problems, so as to apply accepted solutions in order to improve their processes and conduct mitigating actions [35].

When identifying problems and solutions it is helpful to use patterns [425]. One way in which to create a shared understanding of the problems that may occur in the different contexts of GSD is by extracting common patterns that include strategies on how to tackle these problems. For example, the Design Patterns contribution from the Gang of Four [426] has become very important in the field of Software Design education. Another relevant work was conducted by Coplien and Harrison [336] who present the concept of Organizational Patterns, considering the structure of organizations and providing solutions to frequent software development problems. Sanchez-Segura et al. [427] present the idea of product patterns with the intention of improving the efficiency of software engineering practices. Product patterns are interconnected through their entries and exits and gather the experience of experts in the development of the software in order to obtain a specific software product.

"**Pedagogical Patterns**" aim to "capture the essence of the practice in a compact form that can be easily communicated to those who need the knowledge" [428]. These patterns can be used to apply common practices and techniques that regulate the flow of collaborative learning activities. Based on these patterns, teachers can build up their own range of learning modules from a repository of collected experiences that can be reused [429]. **Anti-patterns** are also applied, to define practices that produce more negative consequences than beneficial results [430].

The relevant literature deals with different kinds of patterns and models in several fields of application related to GSD, some of which are summarized in Table 67. As an example, MacGregor et al. [431] present the following:

- Yes (but no) Pattern: Individuals in a certain culture tend to always respond "Yes", causing a problem of miscommunication by making the interlocutor assume false facts.
- We'll-take-you-literally (Anti) Pattern: Different cultures may have different perceptions of a development practice, process, or artifact, placing different levels of importance on them.
- We're-one-single-team (Anti) pattern: The concept of team varies in different cultures, entailing different rules and norms. For example, the use of formal and informal communication can vary in different cultures depending on different factors.
- The-customer-is-king (Anti) Pattern: Different cultures have a different concept of hierarchy, which influences the way in which developers and clients communicate. This is appreciated by the use of direct/indirect speech, and verbal structures to show respect.
- Proxy Pattern: Some individuals who have sufficient experience in interacting with two cultures are placed in a position in which they can, in a sense, translate between cultures.

Table 67 summarizes some of the GSD patterns reported by literature.

Table 67. Patterns in GSD

Cultural patterns
Yes (but no) pattern, proxy pattern, we'll-take-you-literally (anti) pattern, we're-one-single-team (anti) pattern, the-customer-is-king (anti) pattern [431]. Unproductive productivity, hesitant to always say yes, owning rather than modularizing [8].
Communication and interaction patterns
Four main types are identified by Paasivaara et al. [432]: 1) problem solving, 2) informing, monitoring and feedback, 3) relationship building, and 4) decision making and coordination.
Patterns for project management
GSD strategy, fuzzy front end, communicate early, divide and conquer with iterations, key roles in sites, communication tools, common repositories and tools, work allocation, architectural work allocation, phase-based work allocation, feature-based work allocation, use common processes, iteration planning, multi-level daily meetings, iteration review, organize knowledge transfer, manage competence and notice cultural differences [433].
Testing patterns
Test cases as memorandum of understanding of knowledge transfer, light-weight pre-acceptance test, communication on eye level, use tool for bug tracking, moving on-/off-business analyst, complementing testing attitudes, align understanding of general testing approach, continuous integration test, mirrored team manager, extension of the day, traceable test cases, tester's sparring partner, complement test skills, synchronized test environments, central test environment, evaluation of constraints for test data [434].
Requirements engineering patterns
Define 'use case', onboard business analyst during requirements engineering, ship test cases, use work packages and handover checkpoints, use bidirectional cross references and map business terms to entity attributes [435].

Appendix III: Questionnaire used in Expert Evaluation 1

BQ1. Do you think this training method can be effective in the accurate training of specific cultural and linguistic differences in GSD?

BQ2. Do you think it would be feasible to train students/members of your university/company by applying this environment?

BQ3. What problems did you experience while using the Chat Simulator?

BQ4. Can you suggest any point for improvement or new features that you would like to be implemented in the simulator?

BQ5. Upon considering the training scenario in which you have participated, which other training scenarios do you think it would be interesting to design for the training of specific GSD problems or skills?

BQ6. Can you suggest any point for improvement or new features that you would like to be implemented in the Designer?

BQ7. Please, indicate your opinion as regards the following points: 1 (strongly disagree), 5 (strongly agree):

- The Chat Simulator is easy to use
- The Chat Simulator can be used to train people to recognize linguistic differences in GSD
- The Chat Simulator can be used to train people to recognize cultural differences in GSD
- The Chat Simulator is effective in correcting students' mistakes (e.g. use of ambiguous language or inappropriate behavior)
- The information in the final report provides students with useful guidance – i.e., they can, given this information, reflect on how they can improve their linguistic and cultural communication.
- The Designer can highlight typical cultural and linguistic mistakes in GSD
- I enjoyed using the Chat Simulator
- Using a Chat Simulator is a good way to train individuals (as a concept)

BQ8. Nationality
BQ9. How many years have you worked in GSD?
BQ10. Position: ☐ Practitioner ☐ Researcher ☐ Teacher of these subjects___________☐ Other ___________
BQ11. May we contact you again? ☐ Yes ☐ No. Email:________________________
BQ12. May we include your name in the list of experts of this study? ☐ Yes ☐ No

For practitioners only

BQ13. How many years have you worked in the Software Industry?
BQ14. What is your current role?
BQ15. Size of company?
BQ16. Size of IT dept?
BQ17. No. of countries involved in GSD, and country where head office is based?

Appendix IV: Pilot Evaluation feedback questionnaire

PE1. Did you have any problems when executing the training scenario? Please, explain the problems in detail.
PE2. What is your perception of the duration of the evaluation process?
PE3. Can you think of any suggestions that might improve the evaluation process?
PE4. Did you understand all the questions in the questionnaire? Please explain the problems in those questions.
PE5. Are there any ambiguous questions?
PE6. Do you think any important questions are missing?
PE7. Do you think we should remove any questions?
PE8. Do you have any further suggestions to improve the evaluation?

Appendix V: Baseline questionnaire

BASELINE QUESTIONS

Personal details:

BQ18. Nationality, BQ2. Culture, BQ3. Age, BQ4. Gender
BQ5. List the languages you speak along with level of expertise (low, medium, high, native)
BQ6. Qualifications (Secondary school, Graduate, Master's degree, Doctorate)
BQ7. What is your current position?
BQ8. Months of experience in GSD.
BQ9. Are you a practitioner

Only for practitioners

BQ10. Size of current company? (number of employees)
BQ11. Size of IT dept? (number of employees)
BQ12. No. of countries usually involved in Global Software Development
BQ13. Roles that you have played in Global Software Development projects
BQ14. No. of Global Software Development projects on which you have worked (involving different countries).
BQ15. No. of distributed projects on which you have worked (involving different sites in the same country).

Please indicate the extent to which you agree with the following statement (1 strongly disagree, 5 strongly agree):

BQ16. I do NOT like using e-learning systems.

Appendix VI: Pre training questionnaire

SCENARIO A

Please indicate the extent to which you agree with the following statements:

PRAQ1. I am familiar with the concept of a false-friend.

Not at all familiar – Slightly familiar – Somewhat familiar – Moderately familiar – Extremely familiar

PRAQ2. In the case of a positive answer to the previous question, indicate how many Spanish false-friends you know.

None - One- 2/3- 4/5 - more than 5

PRAQ3. I know how to start an ice-breaking conversation with a Mexican.

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PRAQ4. I know which communication style it is most appropriate to use with a Mexican (direct or indirect).

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PRAQ5. I know the concept of the "Mum Effect"

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PRAQ6. I am aware of Mexicans' willingness to work extra hours.

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PRAQ7. I am aware of Mexicans' respect for the chain of command.

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PRAQ8. I am aware that Mexicans tend to value family more than work.

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PRAQ9. ***Some cultures tend to always agree, others tend to be more critical and argue, providing reasons.*** **I am aware of which of these styles is most common for Mexicans.**

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

SCENARIO B

Please indicate the extent to which you agree with the following statements:

PRBQ1. I am familiar with the kinds of mistakes that Hindi speakers make when speaking English.

Not at all familiar – Slightly familiar – Somewhat familiar – Moderately familiar – Extremely familiar

PRBQ2. I know the concept of "Follow-the-Sun".

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PRBQ3. I know the concept of "Unproductive Productivity".

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PRBQ4. I am aware of how the chain of command works in an Indian culture.

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PRBQ5. *Some cultures tend to always agree, others tend to be more critical and argue, providing reasons.* I know which of these styles is most common for Indians.

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PRBQ6. I am aware of how Indians can show concern when they do not agree with you.

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PRBQ7. I am aware of Indians' willingness to work extra hours/overtime.

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PRBQ8. I am aware of how open the Indian culture is as regards asking questions and seeking advice.

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PRBQ9. I am aware of Indians' punctuality when attending meetings or arriving at their workplace.
Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PRBQ10. I am aware of how people from an Indian culture feel about processes, rules and documentation.

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

Appendix VII: Post training questionnaire

SCENARIO A
Please indicate the extent to which you agree with the following statements:

PSAQ1. I am familiar with the concept of a false-friend.

Not at all familiar – Slightly familiar – Somewhat familiar – Moderately familiar – Extremely familiar

PSAQ2. In the case of a positive answer to the previous question, indicate how many Spanish false-friends you know.

None - One- 2/3- 4/5 - more than 5

PSAQ3. I know how to start an ice-breaking conversation with a Mexican.

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PSAQ4. I know which communication style it is more appropriate to use with a Mexican (direct or indirect).

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PSAQ5. I know the concept of the "Mum Effect"

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PSAQ6. I am aware of Mexicans' willingness to work extra hours.

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PSAQ7. I am aware of Mexicans' respect for the chain of command.

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PSAQ8. I am aware that Mexicans tend to value family more than work.

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PSAQ9. *Some cultures tend to always agree, others tend to be more critical and argue, providing reasons.* I am aware of which of these styles is more common for Mexicans.

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

SCENARIO B

Please indicate the extent to which you agree with the following statements:

PSBQ1. I am familiar with the kind of mistakes that Hindi speakers make when speaking English.

Not at all familiar – Slightly familiar – Somewhat familiar – Moderately familiar – Extremely familiar

PSBQ2. I know the concept of "Follow-the-Sun".

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PSBQ3. I know the concept of "Unproductive Productivity".

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PSBQ4. I am aware of how the chain of command works in an Indian culture.

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PSBQ5. *Some cultures tend to always agree, others tend to be more critical and argue, providing reasons.* I know which of these styles is more common for Indians.

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PSBQ6. I am aware of how Indians can show concern when they do not agree with you.

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PSBQ7. I am aware of Indians' willingness to work extra hours/overtime.

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PSBQ8. I am aware of how open the Indian culture is as regards asking questions and seeking advice.

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PSBQ9. I am aware of Indians' punctuality when attending meetings or arriving at their workplace.
Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

PSBQ10. I am aware of how people from an Indian culture feel about processes, rules and documentation.

Not at all aware – Slightly aware – Somewhat aware – Moderately aware – Extremely aware

Generic questions (to apply in both scenarios)
Please indicate the extent to which you agree with the following statements: (1 strongly disagree, 5 strongly agree):

PSQ1. The length of the simulation was too long.
PSQ2. The story in the scenario was clear, concise and complete
PSQ3. The objective of the scenario was clear.
PSQ4. The training scenario was enjoyable.
PSQ5. The training scenario was engaging.
PSQ6. The training scenario was difficult to understand.
PSQ7. The training scenario was representative of a Global Software Development environment.
PSQ8. The training scenario was effective in helping me learn Global Software Development concepts.
PSQ9. The training scenario helped me to improve the skills required in Global Software Development.
PSQ10. It was always clear how to interact with the Virtual Agents.
PSQ11. Overall, I am satisfied with the scenario outcomes.

PSQ12. Please add any additional comments or suggestions.

Appendix VIII: Opinion questionnaire

Cultural and linguistic skills:

Answer the following questions (1 strongly disagree, 5 strongly agree):

- OQ1. I have increased my knowledge of the Indian culture.
- OQ2. I have increased my knowledge of the Mexican culture.
- OQ3. I now have a better concept of how to adapt my style of speech to fit the context of the conversation.
- OQ4. I have gained linguistic understanding with which to interact with people from other cultures.
- OQ5. I have NOT increased my cultural knowledge.
- OQ6. I would feel more comfortable interacting with people from Mexico.
- OQ7. I would feel more comfortable interacting with people from India.
- OQ8. Taking part on this course has increased my interest in other cultures.

Teamwork and general GSD skills:

Answer the following questions (1 strongly disagree, 5 strongly agree):

- OQ9. I am more confident when confronting conflicts, misunderstandings and problems in Global Software Development.
- OQ10. I have more knowledge as regards how to gain the confidence and trust of the person I am interacting with.
- OQ11. I have learnt to detect missing information and misinterpretations of the person I am interacting with.
- OQ12. I think this experience has increased my ability to create professional relationships with people who think differently from me.
- OQ13. I have learnt new ways in which to manage conflicts and disagreements.

OQ14. I think this experience will make me more effective in collaborating and working within a multidisciplinary and multicultural team.
OQ15. I do NOT now feel more able to identify typical problems of Global Software Development.

General perception

Answer the following questions (1 strongly disagree, 5 strongly agree):

OQ16. I have acquired/reinforced new knowledge as regards Global Software Development concepts
OQ17. Overall satisfaction with the learning outcomes.

Answer the following questions (1 strongly disagree, 5 strongly agree):

OQ18. I enjoyed using VENTURE
OQ19. This e-learning platform has the potential to be used by students/members of my university/company.
OQ20. I prefer participating on a training course with VENTURE than in traditional classes.
OQ21. Teaching Global Software Development is useful
OQ22. Using this tool has changed my perception of the usefulness of e-learning systems

e-Learning Environment

Please, indicate your opinion on the e-Learning web as regards the following points: 1 (strongly disagree), 5 (strongly agree):

OQ23. The e-learning platform is easy to navigate and use.
OQ24. The materials and explanations are simple and well organized.
OQ25. I always knew which activities I had to work on.
OQ26. I like the way this course is delivered in an interactive manner.
OQ27. The final reports are clear and help me to understand my problems and my score.
OQ28. I became confused when using the e-Learning platform.

Chat Simulator

Please, indicate your opinion of the Chat Simulator as regards the following points: 1 (strongly disagree), 5 (strongly agree):

OQ29. The Chat Simulator can reproduce Global Software Development scenarios in an accurate manner.
OQ30. Simulated conversations provide awareness of the problems that can appear in Global Software Development.
OQ31. I felt involved in the scenario.
OQ32. The Virtual Guide can highlight and provide useful feedback on Global Software Development.
OQ33. I did NOT enjoy using the Chat Simulator.
OQ34. I learnt more when I failed.
OQ35. The Chat Simulator is easy to use.
OQ36. I was able to use the simulation without getting lost.

Assessment Method

Please, indicate your opinion of the Assessment Method as regards the following points: 1 (strongly disagree), 5 (strongly agree):

OQ37. I was aware of the learning outcomes at the beginning of the course.
OQ38. The assessment information is easy to understand and interpret.
OQ39. The feedback provided facilitated learning.

OQ40. The reports helped me to understand the topics trained.
OQ41. The scores I got seem fair.
OQ42. The assessment motivated me to improve my performance.

Answer the following question:

OQ43. What changes would you propose in order to improve your experience?

Quality in use

USEFULNESS MEASURE

Please, indicate your opinion as regards the following points: 1 (strongly disagree), 5 (strongly agree):

OQ44. The system was simple to use
OQ45. It was easy to learn to use this system
OQ46. I was able to complete the tasks and scenarios quickly using this system
OQ47. The system is stable and I did not receive error messages
OQ48. The description of the scenarios is clear
OQ49. The organization of information on the system screens was clear
OQ50. The information was effective in helping me complete the tasks and scenarios
OQ51. The interface of this system was pleasant
OQ52. I liked using the interface of this system
OQ53. This system has all the functions and capabilities I expect it to have
OQ54. Overall, I am satisfied with this system

TRUST, PLEASURE AND COMFORT MEASURES

Please, indicate your opinion as regards the following points: 1 (strongly disagree), 5 (strongly agree):

OQ55. I feel comfortable using this system
OQ56. I feel confident when using this system
OQ57. This system performs the tasks that it was intended to
OQ58. The information in this training is valuable
OQ59. I feel uncomfortable when using this system

OQ60. Please indicate any further comments or suggestions regarding this topic.

Appendix IX: Normality tests

This appendix presents the Shapiro-Wilk tests used to check the normality of the answers to the questionnaires applied in the Field Study. The Shapiro-Wilk method has been applied because it is appropriate for small sample sizes (< 50 samples).

Scenarios difficulty

Table 68 presents the results from the Shapiro-Wilk Test for the scores obtained in both scenarios. It will be noted from these results that for Scenario A with a p-value < 0.05, the null hypothesis that the sample is normally distributed is rejected, whereas Scenario B is normal (p-value>0.05) according to this test. Figure 60 and Figure 61 show the histograms for both scenarios.

Table 68. Difficulty of scenarios (test of normality)

	Scenario	Shapiro-Wilk		
		Statistic	df	Sig.
Score	Scenario A	.936	34	.047
	Scenario B	.980	34	.778

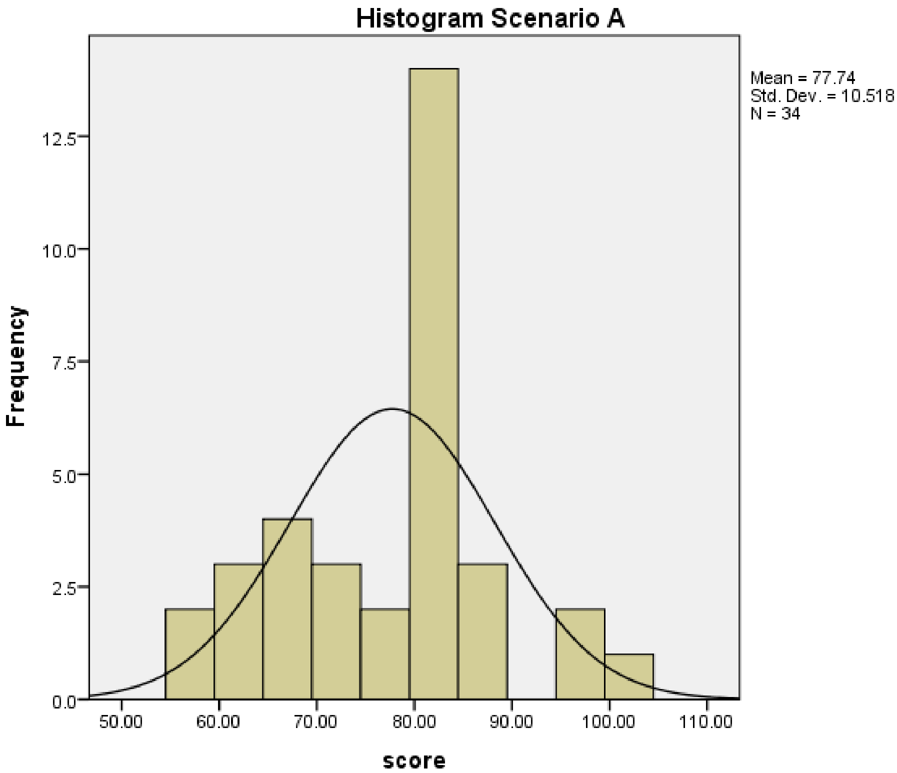


Figure 60. Histogram score Scenario A

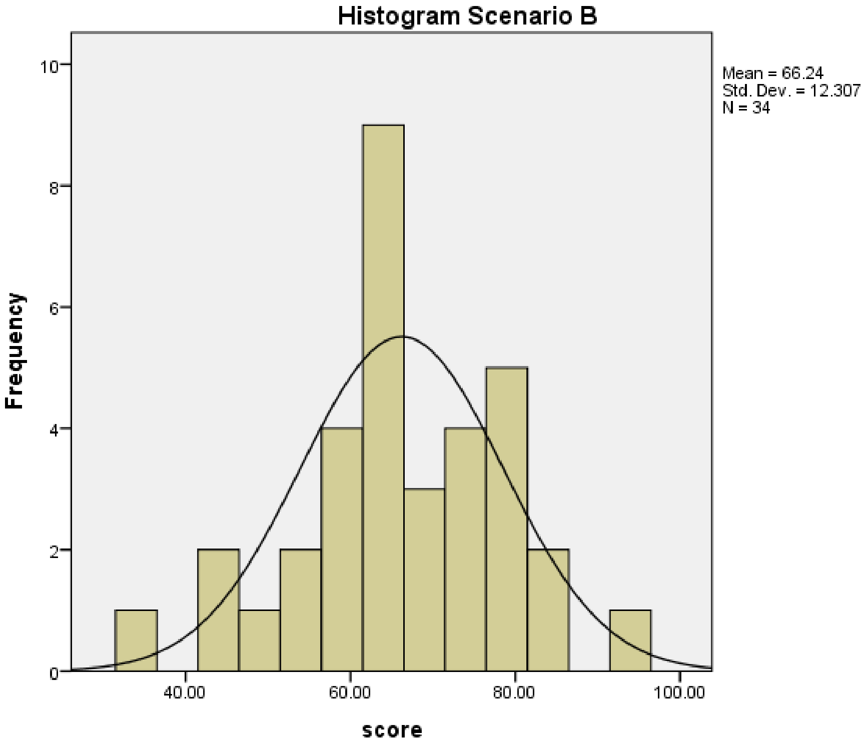


Figure 61. Histogram score Scenario B

Use of direct/indirect style rule

The results of the Shapiro-Wilk test used to check whether the “Use of direct/indirect style” rule was fired following a normal distribution are shown in Table 69, depicting that in both scenarios the p-value was lower than 0.05, so the null hypothesis that the sample is normally distributed is rejected and subsequent tests will be carried out by using non-parametric tests.

Table 69. Use of direct/indirect style (test of normality)

Rule	Scenario	Shapiro-Wilk		
		Statistic	df	Sig.
Use of direct/indirect style	Scenario A	.819	34	.000
	Scenario B	.797	34	.000

The histograms shown in Figure 62 and Figure 63 for Scenario A and Scenario B respectively, confirm this information.

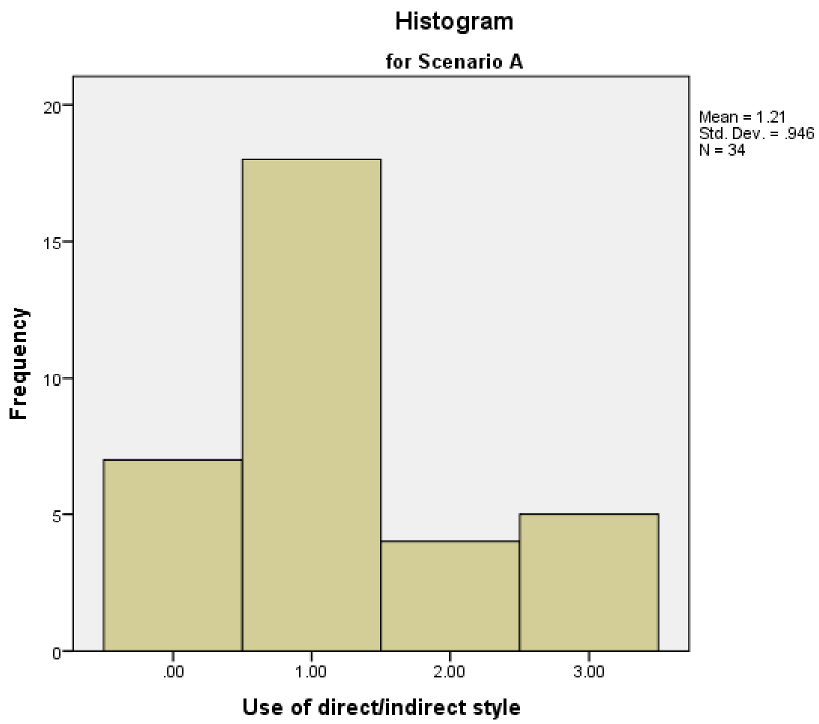


Figure 62. Histogram rule Scenario A

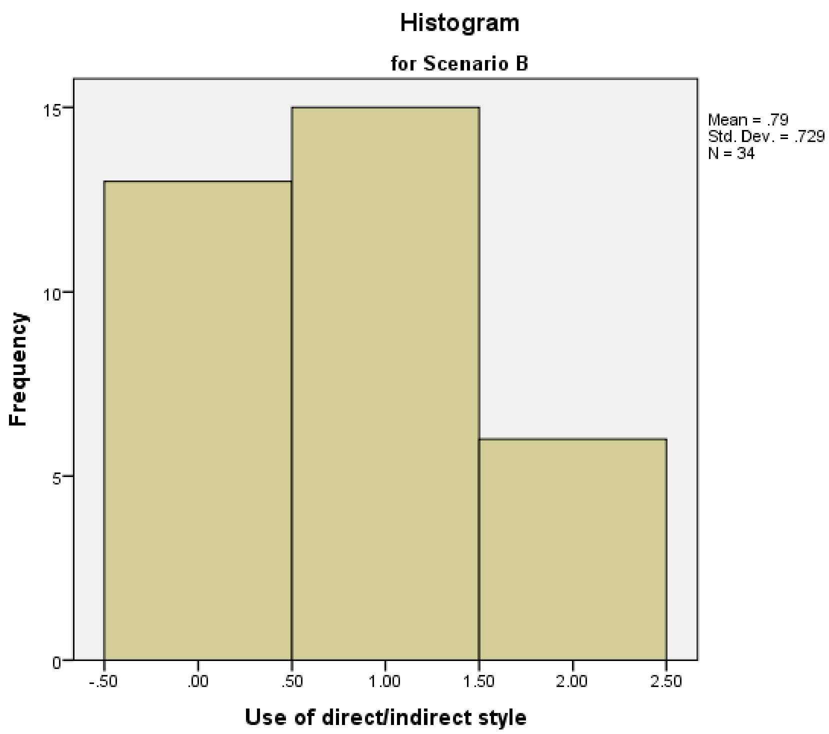


Figure 63. Histogram rule Scenario B

Pre and post questionnaires Scenario A

With regard to the pre questionnaire for Scenario A, the answers do not follow any normal distribution since, according to Table 70, the p-value for the tests is always lower than 0.001.

The answers to the post questionnaire for the same scenario is similarly non normally distributed since, as shown in Table 71, all the p-values are under 0.05.

Table 70. Pre questionnaire Scenario A (test of normality)

	Shapiro-Wilk		
	Statistic	df	Sig.
I am familiar with the concept of a false-friend	.796	32	.000
How many Spanish false-friends do you know?	.771	32	.000
Some cultures tend to always agree, others tend to be more critical and argue, providing reasons. I am aware of which of these styles is more common for Mexicans	.782	32	.000
I am aware that Mexicans tend to value family more than work	.815	32	.000
I am aware of Mexicans' respect for the chain of command	.746	32	.000
I am aware of Mexicans' the willingness to work extra hours	.759	32	.000
I know the concept of the "Mum Effect"	.530	32	.000
I know which communication style it is more appropriate to use with a Mexican (direct or indirect).	.698	32	.000
I know how to start an icebreaking conversation with a Mexican	.798	32	.000

Table 71. Post questionnaire Scenario A (test of normality)

	Shapiro-Wilk		
	Statistic	df	Sig.
I am familiar with the concept of a false-friend	.883	33	.002
How many Spanish false-friends do you know?	.868	33	.001
Some cultures tend to always agree, others tend to be more critical and argue, providing reasons. I am aware of which of these styles is more common for Mexicans	.867	33	.001
I am aware that Mexicans tend to value family more than work	.573	33	.000
I am aware of Mexicans' respect for the chain of command	.813	33	.000
I am aware of Mexicans' willingness to work extra hours	.783	33	.000
I know the concept of the "Mum Effect"	.796	33	.000
I know which communication style it is more appropriate to use with a Mexican (direct or indirect).	.881	33	.002
I know how to start an icebreaking conversation with a Mexican	.850	33	.000

Pre and post questionnaires Scenario B

With regard to Scenario B, the answers for both the pre questionnaire and the post questionnaire do not follow any normal distribution since, according to Table 72 and Table 73, respectively, the p-value for the tests is always lower than 0.05.

Table 72. Pre questionnaire Scenario B (test of normality)

	Shapiro-Wilk		
	Statistic	df	Sig.
I am familiar with the kinds of mistakes that Hindi speakers make when speaking English	.702	34	.000
I am aware of how people from an Indian culture feel about processes, rules and documentation	.728	34	.000
I am aware of Indians' punctuality when attending meetings or arriving at their workplace	.811	34	.000
I am aware of how open the Indian culture is as regards asking questions and seeking advice	.763	34	.000
I am aware of Indians' willingness to work extra hours/overtime	.838	34	.000
I am aware of how Indians can show concern when they do not agree with you	.795	34	.000
Some cultures tend to always agree, others tend to be more critical and argue, providing reasons. I know which of these styles is more common for Indians	.830	34	.000
I am aware of how the chain of command works in an Indian culture	.740	34	.000
I know the concept of "Unproductive Productivity"	.880	34	.001
I know the concept of "Follow-the-Sun"	.819	34	.000

Table 73. Post questionnaire Scenario B (test of normality)

	Shapiro-Wilk		
	Statistic	df	Sig.
I am familiar with the kinds of mistakes that Hindi speakers make when speaking English	.885	34	.002
I am aware of how people from an Indian culture feel about processes, rules and documentation	.868	34	.001
I am aware of Indians' punctuality when attending meetings or arriving at their workplace	.852	34	.000
I am aware of how open the Indian culture is as regards asking questions and seeking advice	.810	34	.000
I am aware of Indians' willingness to work extra hours/overtime	.883	34	.002
I am aware of how Indians can show concern when they do not agree with you	.849	34	.000
Some cultures tend to always agree, others tend to be more critical and argue, providing reasons. I know which of these styles is more common for Indians	.840	34	.000
I am aware of how the chain of command works in an Indian culture	.882	34	.002
I know the concept of "Unproductive Productivity"	.846	34	.000
I know the concept of "Follow-the-Sun"	.772	34	.000

Appendix X: Training Scenario (Session A)

Maria: Hi, you can start your conversation with Carlos now by answering his question.

Carlos: Hello, how are you?

Sue: Fine, and you?

Carlos: Nice to know that. I´m ok too, thanks.

Maria: It will ease the conversation flow if you start with some neutral ice-breaking chat. For example, ask him about the weather and work life in general. It would seem too abrupt if you launched straight into the problem.

Sue: How is the weather with you? It´s raining here.

Carlos: It's been raining here all day, but we expect better weather tomorrow. I hope I get a chance to enjoy it!

Carlos: I wanted to talk to you about the plannification for the new test cases

Maria: 'Plannification' is a common mistake in Spanish. He wanted to say plan or schedule.

Sue: Ok, go ahead. I would like to hear your plan.

Carlos: The client needs the set of 40 test cases executed for the module M by the end of the month. Can you tell me what the status of those test cases is?

Maria: Instead of just mentioning the progress, share information related to any challenges that you are confronting. Here, out of the 40 test cases, 10 are very complex and we will need more time to execute them.

Sue: Ten of these test cases are more complex and require more time.

Carlos: I see. We are also planning some code walkthrough tasks for some of the modules. Would you send me the cyclomatic complexity of all the functions in order to prepare a schedule?

Maria: Code Walkthrough is a manual testing technique in which source code logic is traced by a group with a small set of test cases. The Cyclomatic Complexity is a measure of the complexity of the code, which is related to the number of decisions or conditions. Just tell him that you will send him this information later.

Sue: I´ll send you this information tomorrow.

Carlos: Actually, I need that report right away.

Maria: You should clarify what he means by ‘right away’. Perhaps he means before the end of the day and you should drop every other priority, or perhaps he means before the next meeting.

Sue: When exactly do you need the information?

Carlos: I have to prepare the schedule by the beginning of next week, is that ok with you?

Sue: Ok, we will try to meet that deadline.

Carlos: I think that in order to meet the deadline, we would need at least one more tester on your side, is that possible?

Maria: Typically, in Global Software Development settings, problems occur when dealing with times and deadlines. Explain that the most complex test cases will take longer to execute and cannot be done in parallel.

Sue: An extra tester is not a good option, because it is not convenient to execute some of these test cases in parallel.

Carlos: How many testers should we have to do this?

Maria: We should have 2 testers for the next four weeks.

Sue: Two of them.

Carlos: I think that should be enough. We have strict time constraints for this, and it should be completely finished in no more than a month. Is that OK with you?

Maria: 4 weeks is a good estimation for executing the test cases.

Sue: Yes, no problem.

Carlos: Very well, the plan is to schedule the work for 4 weeks. 2 weeks for executing the test cases and 2 weeks for a buffer in order to solve problems. OK?

Sue: Sure.

Maria: Carlos is leading you to misunderstand the time restrictions. He requires the test cases to be executed in 2 weeks, when your initial agreement was 4 weeks. The 2 weeks for a buffer are going to be used for solving problems but not for executing test cases. Explain that, from your point of view, it is not possible to meet the deadline.

Sue: Sorry, I thought we had 4 weeks for executing the test cases, and now I see that it wouldn´t be enough.

Carlos: Well, I do not understand your reasons. I think you shouldn't drown in a glass of water; it should be enough if you start next week.

Maria: The Spanish expression 'to drown in a glass of water' is equivalent to 'to make a mountain out of a molehill'. In order to be more persuasive, try to make him consider your position with new arguments. Mention that the infrastructure required is not available, as testers need new software and training that will delay the start of the project.

Sue: The testers will need new software, and there will be a delay at the beginning.

Carlos: Actually, they will not need so much time for installing and getting used to the infrastructure, don't worry about that.

Maria: He is trying to convince you that the infrastructure is not going to delay you. Mention that you still do not have the documentation of the project required to start.

Sue: We are also waiting for the documentation.

Carlos: I know, we are working on it and will send you the documentation ASAP.

Maria: You will need at least one more week.

Sue: Would it be possible to extend the deadline one more week?

Carlos: I do not think it is possible to extend the deadline. Two weeks should be enough, but I will speak with the project manager and let you know.

Maria: Now you must speak about a technical problem that you have. It is related to a webservice that the offshored team developed and you are experiencing problems when using it. This webservice was developed by the Mexican team.

Sue: We also have a problem with a webservice.

Carlos: OK, I remember, we realized this webservice some months ago. Can you give me more details about the problem?

Maria: 'Realize' is a typical Spanish false-friend made when speaking English. He means 'develop'. Now explain that your team has been able to connect, but the service is still not working.

Sue: The service is not working.

Carlos: Ok, it should work, but please explain what the problem is.

Maria: When you are trying to use the webservice, you receive an error and the connection is refused.

Sue: The connection is refused.

Carlos: I see. Do you have any error message that I could review?

Maria: Your team is receiving the following error: Cannot import wsdl:portType: The datatype 'Array' is missing.

Sue: The array datatype is missing.

Carlos: Oh, I will take a look at this.

Maria: After studying the problem, you found that the webservice is not following the WSDL standard because the datatype "Array" is not defined in this standard and they should change this datatype.

Sue: You have to change the datatype array.

Maria: You should try to rephrase this sentence to be less direct.

Sue: Would it be possible to change the datatype?

Carlos: I know what you mean. Let me speak with the developer about this.

Carlos: Can I know the actual version of the webservice that you are using?

Maria: 'Actual' is also a typical Spanish mistake. He wanted to say 'current' or 'most recent'. You must answer that you are working with the latest version.

Sue: The latest version.

Carlos: In that case, I don't think it is a problem of the webservice, since I tested it previously.

Maria: You are sure the problem is in the webservice

Sue: It must be the webservice

Carlos: I understand. I will speak with the supervisor when he has a minute in order to know if the developer can spend time on this next week.

Maria: A characteristic of the Mexican team is that they tend to rigorously respect the chain of command. You cannot wait for so long - it is a critical problem. Ask him to solve it even if they have to work during the weekend.

Sue: Would he be able to work on it this weekend?

Carlos: Are you sure there is not another way to solve it?

Maria: In this case, Carlos will try to convince you that extra hours are not needed. He is afraid of telling this bad news to the project manager. The tendency to cover up critical information or to distort negative news by presenting it as more positive information is known as the 'Mum Effect'. You must insist on the urgency of the solution

Sue: It is a critical problem.

Carlos: I have spoken with the developer and he told me that he cannot work extra hours. We must look for an alternative solution. I will look for another developer to review the problem in order to have it solved today.

Maria: Mexicans do not usually allow schedules to interfere with plans involving family and friends. The Mexican team is not used to working extra hours and this proposition may be uncomfortable for them.

Sue: Ok, let me know.

Carlos: I will keep you updated.

Maria: You should at least plan a meeting for Thursday (this week)

Sue: Can we have a meeting on Thursday?

Carlos: Sorry, this Thursday I was going to take my daughter to the doctor, she recently had health problems.

Maria: For Mexicans, one's family comes before one's work, and you must understand that and try to delay the appointment.

Sue: ok, I understand. When is a good time for you?

Carlos: Thank you, moreover this Friday is a regional holiday here. Could we do it on Monday?

Sue: Sure, but please, understand that this is important.

Carlos: Very well, I'll keep you informed. Have a good day!

Carlos: Bye!

Maria: Well done, you have completed the scenario within the time limit. Now you can close this window.

Appendix XI: Training Scenario (Session B)

Maria: As you are on the management site, you have to try to lead the conversation. Before getting to the point, start by showing interest in how he is.

Amal: Hello, good to talk to you.

Sue: Hi Amal! How are you?

Amal: I'm fine, thanks.

Maria: Amal has recently gone back to work after 2 days' medical leave. Try to break the ice by talking about that.

Sue: I hope you are better after your leave.

Amal: Thanks, I am fully recovered now to start working again.

Maria: The other day you had a problem on your site and, as Amal was not in the office, you spoke with his colleague Suraj. After solving the problem a report was generated. Tell him that you sent this documentation to Suraj

Sue: I spoke with Suraj about something and I sent him a document.

Amal: Ok, will I need to read the documentation?

Maria: He needs to have these documents (Amal sits in the office opposite Suraj's), and they are on Amal's desk

Sue: Yes, you can pick it up from Suraj's office

Maria: Status is very important in the Indian culture. Amal has a high status in the company and you should not suggest that he goes to the office and takes the documentation. As this is a manual task, he could feel offended. Just suggest that he obtains the documentation.

Sue: Could you obtain the document?

Amal: Sure, I will obtain the documents.

Maria: Now ask him about the progress on the development of the installer on which they have been working.

Sue: What's the status of the installer?

Amal: The installer is quite preponed, I will start preparing the documentation tomorrow.

Maria: In the Hindi language 'preponed' is supposed to be the opposite of postponed. This is a typical error in the use of English. He meant the installer is quite 'advanced'. You now need to know how long they will be working on the documentation for the installer.

Sue: How long are you going to be working on the documentation?

Amal: I could not say. When would you like it?

Maria: In his culture, it's the boss's job to give direction and the employee's job to please the boss. It would be too presumptuous of him to estimate it. A good boss (for Amal) should do that work. Indicate that you will need it for this Wednesday.

Sue: We would need it for this Wednesday.

Amal: This Wednesday?

Sue: Yes.

Amal: Wednesday afternoon?

Maria: In Amal's culture they always tend to agree, especially when dealing with authority figures. However, they have several ways of showing disagreement. In this case, Amal knows the deadline is too soon and he is letting you know this by repeating the question. Ask him to send it on Thursday.

Sue: Is Thursday more convenient for you?

Amal: That would be perfect, thanks. We will have the installer finished by the end of Thursday, ok?

Sue: That's ok with me.

Maria: Amal is leading you to misunderstand his commitment. He has only agreed to finish the installer on time. However, you also require the documentation. Explain that you need both.

Sue: But I would also need the documentation.

Amal: If you also need the documentation it would be great if we could have one more day.

Maria: It is critical to have it on Friday morning. This must be finished because the new project will start next month and you need people available.

Sue: It is critical to have it on Friday. We will need people available for the next project.

Amal: No problem. An important project is finishing tomorrow, and we will have people available with experience in follow-the-sun for the next project.

Maria: In follow-the-sun, the tasks are passed around daily between work sites that are many time zones apart in order to reduce project duration and increase responsiveness (the work never stops). Now tell him that you will prepare the plan, counting on 6 developers.

Sue: I will prepare the plan considering 6 developers

Amal: OK, we will wait for more news on this.

Maria: You sent Amal the Test Case document for the new project last week. However, you know that there are some points that may be confusing. Ask him whether he understood everything

Sue: Did you understand the test case document?

Amal: Yes, I will review some points in more depth later, thanks.

Maria: In Amal's culture there is a certain reluctance to ask questions or for advice. This can lead the team to make their own unilateral assumptions about how things should be. In order to avoid problems, you must be responsible and ensure that they understand everything. The most complex part is related to the user's management. Ask him about it.

Sue: Any problem with the user's management part?

Amal: There are some aspects that I would like to study in more detail.

Maria: Ask him about the most critical aspect that he has found. That way he will let you know this in a subtle manner.

Sue: What's the most critical aspect of this?

Amal: Well, I see some redundancies in the definition of the test cases and I think it is not necessary to conduct all of them.

Maria: A common problem occurs when the offshore team works on tasks that are not really necessary. The offshore team is considered to be productive when they generate the results that are documented. However, Amals' perception is that some of these tasks are unproductive. This is known as unproductive productivity. Let him know that you will send him a new version of the document after reviewing it.

Sue: I will send a new version after considering your comments.

Amal: Thank you, I appreciate that.

Maria: You will need to send them a complete schedule for the testing tasks. However, the architect is on holiday, and the project is starting next week. Make him understand that they will need to start the project without this document for the moment.

Sue: We will need to start working next week without the schedule, as the architect is not around.

Amal: I understand, but it would be good if we could have a better idea about this schedule from the beginning.

Maria: You cannot give him more details at the moment.

Sue: Sorry, I do not have more details at the moment.

Amal: What kind of tasks should we work on in the meantime?

Maria: Amal is letting you know by his questions that he thinks it would be more convenient not to start the project without this information. The Indian team is used to applying the methodology in a highly disciplined way. Let him know that you will prepare a provisional document with this information that will evolve in the following weeks.

Sue: I will prepare a provisional document to start, that will evolve, ok?

Amal: I see. It also peeks my curiosity if we are going to need any additional resources or architecture before starting this project.

Maria: Amal means 'to pique my curiosity' instead of 'peek'. Indicate that at the moment they will need a template with corporative images and colors that you are going to send next week.

Sue: I will send you a template next week.

Amal: As both sites are going to work on the same modules, do you think the project could be delayed because of the interdependencies?

Maria: An ideal arrangement in Global Software Development would let the sites operate as independently as possible; however, this is not always possible. Mention that you must define periodic meetings in order to track this problem.

Sue: We will have periodic meetings in order to see how it is going.

Amal: I understand.

Maria: It is also necessary to ensure that on both sites, every developer commits the code by the end of their working shift in order to minimize problems. Mention that they must be especially careful with this.

Sue: Every member will need to commit every day before they leave.

Amal: But they should only commit when their code compiles and does not have errors, no?

Maria: The structure of this question is another common mistake of Hindi speakers. It should be 'Should they commit even if their code has errors?'. Mention that you will set up rules in the code version system in order to determine the conditions in which it is possible to commit.

Sue: We will set up a policy in the server so that we will determine these conditions.

Amal: By the way, I have to remind you that during the last project our team had some problems because we had to wait for answers from your team for too long. Sometimes your team did not pick up calls and this was a big problem for us.

Maria: Your team does not tend to work before or after their working hours. The different time shifts and holidays have caused problems in the past. Mention that you will make an effort to improve the documentation and provide quick responses with yourself as a coordinator.

Sue: We will have to improve the documentation and I will try to be available.

Amal: Thanks for letting me know.

Maria: These previous problems were understood by your team as a lack of competence on the part of the Indian team. From your perspective, you must let them know that they have to consider and anticipate all these problems in order to avoid these situations.

Sue: You must also anticipate these situations.

Maria: You should try to rephrase this sentence to be less direct.

Sue: It would be good if you could also anticipate this kind of situation.

Amal: I know, but it is not always easy to anticipate something that is evolving all the time.

Maria: Before finishing the conversation, you must establish a regular schedule for the meetings for the new project. Although the Indian members work for long hours, they do not tend to be punctual. Ask him for the earliest time after his lunch at which he could be available on Tuesdays

Sue: When is the best moment to have a regular meeting on Tuesdays?

Amal: I could be available at 14:30 (IST).

Maria: IST is the acronym for Indian Standard Time, which is equal to GMT + 5:30. Just agree to this time and say goodbye.

Sue: Perfect, see you then.

Amal: Perfect, talk to you later.

Amal: Bye!

Maria: Well done, you have completed the scenario within the time limit. Now you can close this window.