# GLOBAL-MANAGER: A Serious Game for Providing Training in Project Manager Skills

Rubén Márquez
University of Castilla-La Mancha
Ciudad Real, Spain
Ruben.Marquez@uclm.es

Aurora Vizcaíno
University of Castilla-La Mancha
Ciudad Real, Spain
Aurora.Vizcaino@uclm.es

Félix Oscar García
University of Castilla-La Mancha
Ciudad Real, Spain
Felix.Garcia@uclm.es

Antonio Manjavacas
University of Castilla-La Mancha
Ciudad Real, Spain
Antonio.Manjavacas@uclm.es

## ABSTRACT

The emergence of Global Software Development (GSD) has led to certain difficulties in the life cycle of global projects, in addition to the traditional challenges of collocated development, particularly as regards Project Management (PM). These difficulties are caused by the geographical, linguistic and cultural distance among the members of the team, signifying that the project manager requires special skills with which to mitigate these issues. Bearing this in mind, this paper describes a serious game (SG), denominated as GLOBAL-MANAGER, whose objective is to provide training in the management of GSD projects. The game attempts to develop several skills in its players whilst simultaneously providing them with an immersive, pleasant and attractive experience. The skills developed are related to coordination, communication and control, which are three of the principal challenges in GSD.

## CCS CONCEPTS

• Social and Professional Topics• Professional topics• Computing education • Computing education programmes • Software engineering education

## KEYWORDS

Project Management, Global Software Development Training, Global Software Engineering Training, Soft Skill, Serious Game

## 1 Introduction

Global Software Development involves several challenges, which may come about as the result of a variety of factors, such as the geographical distance among work teams, the coexistence of different languages and cultures, and the handling of different time zones [1], along with other problems such as a lack of communication or cooperation among members [2].

The issues that may exist in GSD projects signify that there is an evident need for their workers to have additional competences to those required in traditional software development [3], which would make it possible to avoid the problems. Moreover, educators must teach future software engineers to work in these type of projects [3], [4]. This is especially true as regards the management of the project, since this is the central part where the project is coordinated and controlled, thus enabling it to advance without incidents. Moreover, most of the failures of these projects are a direct consequence of the project managers' lack of skills and experience. Finding qualified, experienced project managers with a range of skills with which to effectively manage this type of project is, therefore, a real challenge [5].

SGs could, however, be of use in this task of training engineers to meet the challenges that a project manager may confront when supervising this type of project. These games can be defined as educational games that go beyond pure entertainment and which aim to impart knowledge to their players [6]. SGs are, moreover, excellent tools for the learning process, since they play a specific role in simulating an environment without the player having to be in a real context, thus avoiding putting a project at risk, while training students [3], [5]. Furthermore, according to Zhoe et al. [7], SGs provide support in the learning process of new students in an independent manner, thus improving their skills to confront and solve problems. SGs can, however, also be simultaneously viewed as a tool for practitioners that will enable them to train in certain tasks and skills and assist them to acquire a range of new abilities and learn to address any problems that may arise [8].

An educational game called GLOBAL-MANAGER, which takes into account all the aspects discussed above, will be presented in this paper. The aim of this SG is to allow students and practitioners to train in certain skills that are needed in the role of project manager in distributed settings. The player will have to tackle the management of a fictitious GSD project from its beginning to the delivery of the product and attempt to solve several difficulties and obstacles that may arise. We believe that this game will help future project managers to specialise, by gaining experience independently and easily, thus enabling them to successfully confront their work on the real market.

The remainder of this paper is structured as follows. Section 2 discusses the background and related work regarding the skills required to manage a GSD project, in addition to outlining the SGs that have already been developed to teach similar types of knowledge. The details of the game, as regards its design and execution, are discussed in Section 3. The paper ends in Section 4 with a conclusion and discussion of future work.

GLOBAL-MANAGER: A Serious Game for Providing Training in Project Manager Skills
R. Márquez, A. Vizcaíno, F.O. García, A. Manjavacas


*ICGSE '20, October 5–6, 2020, Seoul, Republic of* Korea

ACM ISBN 978-1-4503-7093-6/20/05…$15.00
https://doi.org/10.1145/3372787.3390443

**Proceedings**

# 2020 ACM/IEEE 15th International Conference on Global Software Engineering

# ICGSE 2020

**Seoul, Republic of Korea**
**26-28 June 2020**

**Proceedings**

# 2020 ACM/IEEE 15th International Conference on Global Software Engineering

# ICGSE 2020

**Seoul, Republic of Korea**
**26-28 June 2020**


The Association for Computing Machinery
2 Penn Plaza, Suite 701
New York New York 10121-0701



ACM ISBN: 978-1-4503-7093-6

Editorial production by Lisa O'Conner
Cover art production by Annie Jiu


*IEEE Computer Society*
***Conference Publishing Services* (CPS)**
http://www.computer.org/cps

# 2020 ACM/IEEE 15th International Conference on Global Software Engineering (ICGSE)

# ICGSE 2020

## Table of Contents



## Agile Methods



## Human Aspects of Software Development

## Onboarding and Trust



## Open Sourcing and Crowdsourcing

## Requirements and Design Thinking



## Tools and Processes

# Message from the General Chair and Program Co-Chairs

## ICGSE 2020

Welcome to the 15th IEEE/ACM International Conference on Global Software Engineering (ICGSE 2020).

ICGSE 2020 brings together researchers and practitioners to share their research findings, experiences, and new ideas on diverse themes related to Global Software Engineering (GSE). For 2020, the theme of the conference is “Human-centered global software engineering”.

The software industry is increasingly becoming more global with stakeholders and infrastructure distributed across the world. GSE is still challenging, with a considerable share of global projects not meeting the expectations, especially regarding cost savings and time to market. Over the past decade, research on GSE uncovered that cultural, geographical, and temporal differences hinder collaboration, coordination, and communication in global projects. Organizations have drawn on these insights and developed approaches to mitigate the impact of these differences. However, the challenges faced in globally distributing traditional software engineering persist. Adopting modern approaches such as agile software development gives rise to new difficulties. Uncovering the issues underlying these difficulties deserves more research effort. Moreover, there is also room for investigating how different factors (social, cultural, organizational, and technical), their differences, and their complementarities may foster innovation, new business opportunities, and new approaches to improve software project performance and quality.

As the COVID-19 pandemic continues to develop around the globe, we are unsure whether these proceedings will eventually find life in the conference program. The steering committee of ICGSE has agreed to remain co-located with the International Conference on Software Engineering (ICSE 2020). Therefore, herein we collect our exciting contributions from both practitioners and academics, and we present additional events, which will occur subject to the conference taking place as planned.

Starting from our two distinguished keynote speakers, we are proud to announce that Professor Jim Herbsled, from the Carnegie Mellon University (CMU), and Diane Mueller, from RedHat (the leading provider of enterprise open source solutions), will open each of the conference days.

ICGSE has always been especially proud of the high interest of industrial practitioners for the conference, and this year makes no exception. Thanks also to the work of the Industry and Academic Liaisons, the contributions to the ICGSE 2020 proceedings are a well-balanced mix of industrial experiences and academic research providing plenty of opportunities for fruitful discussions, which will help us to ‘fill in the gap’ and further our understanding of how to manage GSE challenges more effectively as well as further encourage industry-academia collaborations.

ICGSE 2020 proceedings include overall 18 high-quality contributions to two different tracks:

- Research Paper Track: 8 papers (32% acceptance rate)
- Experience Report Track: 10 reports

Building upon the experience from previous editions, this year we have introduced some innovations to the program. First, we have taken advantage of a unique opportunity to collaborate with the organizers of the ICSSP 2020 conference, also co-located with ICSE. We are running several formal and informal joint events in which participants of both communities will be able to cross-contaminate by challenging and learning from each other. Second, continuing a recent tradition, ICGSE 2020 partners with IEEE Software and the Journal of Systems and Software (the novelty of this year) to incorporate journal-first presentations in the conference program, thus, providing ICGSE attendees with an additional offering in the Research Track. Finally, we have kept the tradition of soliciting submissions from practitioners to propose industry talks.

We gratefully acknowledge the hard work of the program committees of the various tracks: Your support was critical to the proceedings and the success of the conference! Also, we are particularly grateful to the other organizers: Anna Filippova and Josiane Kroll Noll (Experience Report Co-Chairs); Igor Wiese (Proceedings Chair); Gustavo Pinto (Publicity & Social Media Chair); Aivars Sablis (Webmaster); and all our Industry and Academic Liaisons. We also thank the ICGSE Steering Committee, led by Filippo Lanubile, for the encouragement, advice, and guidance received.

Finally, we would like to thank both IEEE and ACM for sponsoring ICSE and all its co-located events. We are thankful for their invaluable support for the conference series.

We are excited about the conference proceedings and program. We hope you will enjoy the contributions and find the presentations, discussions, and events an exciting and thought-provoking opportunity. We welcome you to the 15th IEEE/ACM International Conference on Global Software Engineering (ICGSE 2020)!

Sincerely,

Paolo Tell, General Chair
IT University of Copenhagen, Denmark

Igor Steinmacher, Program Co-Chair
Northern Arizona University, United States

Ricardo Britto, Program Co-Chair
Ericsson / Blekinge Institute of Technology, Sweden

# Organizing Committee

## ICGSE 2020

**General Chair**

Paolo Tell
*IT University of Copenhagen*
*Denmark*

**Steering Committee Chair**

Filippo Lanubile
*University of Bari*
*Italy*

**Program co-Chair**

Igor Steinmacher
*Northern Arizona University*
*United States*

**Program co-Chair**

Ricardo Britto
*Ericsson*
*Blekinge Institute of Technology*
*Sweden*

**Experience Report co-Chair**

Josiane Kroll
*University of Manitoba*
*Canada*

**Experience Report co-Chair**

Anna Filippova
*GitHub*
*United States*

**Proceedings Chair**

Igor Wiese
*Federal University of Technology, Paraná (UTFPR)*
*Brazil*

**Social media Chairs**

Gustavo Pinto
*Federal University of Pará (UFPA)*
*Brazil*

**Webmaster**

Aivars Sablis
*Blekinge Institute of Technology*
*Sweden*

## Academic Liaisons

Akinori Ihara
*Wakayama University, Japan*

Alberto Avritzer
*eSulabSolutions, United States*

Darja Šmite
*Blekinge Institute of Technology, Sweden*

Kelly Blincoe
*University of Auckland, New Zealand*

Lincoln Souza Rocha
*Federal University of Ceará, Brazil*

## Industry Liaisons

Dongwon Hwang
*SAP, South Korea*

Gleydson Lima
*eSIG, Brazil*

Johan Christensson
*City Network, Sweden*

Shinobu Saito
*NTT Software Innovation Center, Japan*

V. S. Mani
*Siemens, India*

# Program Committee

## ICGSE 2020

Akinori Ihara
*Wakayama University, Japan*

Alberto Avritzer
*eSulabSolutions, United States*

Alexander Nolte
*University of Tartu, United States*

Alpana Dubey
*Accenture, India*

Anh Nguyen Duc
*University College of Southeast Norway, Norway*

Aurora Vizcaíno
*Universidad de Castilla-La Mancha, Spain*

Awdren Fontão
*Federal University of Mato Grosso do Sul (UFMS), Brazil*

Ayushi Rastogi
*TU Delft, the Netherlands*

Ban Al-Ani
*ADP, United States*

Bedir Tekinerdogan
*Wageningen University, The Netherlands*

Casper Lassenius
*Aalto University, Finland*

Christoph Treude
*The University of Adelaide, Australia*

Daniela Damian
*University of Victoria, Canada*

Daniela S. Cruzes
*SINTEF, Norway*

Darja Šmite
*Blekinge Institute of Technology, Sweden*

David Redmiles
*University of California, Irvine, United States*

Davide Fucci
*Blekinge Institute of Technology, Sweden*

Deepika Badampudi
*Blekinge Institute of Technology, Sweden*

Denae Ford
*Microsoft Research, United States*

Eduardo Figueiredo
*Federal University of Minas Gerais (UFMG), Brazil*

Fabio Calefato
*University of Bari, Italy*

Filippo Lanubile
*University of Bari, Italy*

Guilherme Amaral Avelino
*Federal University of Piaui (UFPI), Brazil*

Hideaki Hata
*Nara Institute of Science and Technology, Japan*

Iftekhar Ahmed
*University of California, Irvine, United States*

Igor Wiese
*Federal University of Technology - Paraná (UTFPR), Brazil*

Ita Richardson
*Lero - Irish Software Engineering Research Centre, University of Limerick, Ireland*

Ivana Bosnić
*University of Zagreb, Croatia*

Juho Mäkiö
*University of Applied Sciences Emden/Leer, Germany*

Julian Bass
*University of Salford, UK*

Jürgen Münch
*Reutlingen University, Germany*

Klaas-Jan Stol
*Lero, University College Cork, Ireland*

Leonardo Murta
*Universidade Federal Fluminense (UFF), Brazil*

Leticia Santos Machado
*Federal University of Pará (UFPA), Brazil*

Lincoln Souza Rocha
*Federal University of Ceará (UFCE), Brazil*

Mahmood Niazi
*King Fahd University of Petroleum and Minerals, Saudi Arabia*

Mansooreh Zahedi
*University of Copenhagen, Denmark*

Marcelo Cataldo
*EMC Corp, United States*

Marco Gerosa
*Northern Arizona University, United States*

Marco Kuhrmann
*Clausthal University of Technology, Germany*

Maria Paasivaara
*Aalto University / IT University of Copenhagen, Denmark*

Martin Naedele
*ABB Corporate Research, Switzerland*

Muhammad Usman
*Blekinge Institute of Technology, Sweden*

Nazareno Aguirre
*University of Rio Cuarto, Argentina*

Nils Brede Moe
*SINTEF, Norway*

Pooyan Jamshidi
*University of South Carolina, United States*

Raula Gaikovina Kula
*NAIST, Japan*

Sarah Beecham
*Lero - the Irish Software Engineering Research Centre, Ireland*

Sebastian Baltes
*The University of Adelaide, Australia*

Stefanie Betz
*Karlsruhe Institute of Technology, Germany*

Steven D. Fraser
*Innoxec, United States*

Tayana Conte
*Universidade Federal do Amazonas (UFAM), Brazil*

Tony Clear
*Auckland University of Technology, New Zealand*

Viktoria Stray
*University of Oslo, Norway*

Yan Wang
*Google MTV, United States*

# Steering Committee

## ICGSE 2020

Alpana Dubey
*Accenture, India*

Christof Ebert
*Vector Consulting, Germany*

Darja Smite
*Blekinge Institute of Technology, Sweden*

David Redmiles
*University of California, Irvine, United States*

Fabio Calefato
*University of Bari, Italy*

Filippo Lanubile
*University of Bari, Italy*

Igor Steinmacher
*Northern Arizona University, United States*

Marcelo Cataldo
*EMC Corp, United States*

Maria Paasivaara
*Aalto University / IT University of Copenhagen, Denmark*

Paolo Tell
*IT University of Copenhagen, Denmark*

Ricardo Britto
*Ericsson / Blekinge Institute of Technology, Sweden*

Tony Clear
*Auckland University of Technology, New Zealand*

V.S. Mani
*Siemens, India*

# Sponsors

## ICGSE 2020





## 2 Background

In this section, we review what work exists in the current literature as regards issues related to the two important aspects that had to be taken into account when developing our SG. We shall first deal with the skills required to manage a GSD project, and will then go on to describe some of the SGs that have been developed to teach or educate in these two areas (PM and GSD).

### 2.1 Soft Skills

Members of a software project (SP) require certain technical and non-technical skills in order to deliver a product satisfactorily. This number of skills increases in a GSD setting owing to the large quantity of difficulties that may be encountered.

It should be noted that companies currently complain that when students finish university, they have technical knowledge but no experience or non-technical skills (soft skills), since they have not been involved in real projects. It is, therefore, important to teach students certain soft skills by means of fictitious projects (similar to real ones), thus enabling them to take projects successfully [9].

Several papers currently describe the soft skills that are most important when working in a global setting. One of these is a systematic review [10] that indicates *problem resolution related to cultural differences*, *time management*, *ability to communicate in a common language* or *control skills*. Another paper [11] states that the most important skills are *communication in English*, *intercultural cooperation*, *remote cooperation* and *cooperation with customers*. The authors of [9] state that *time management*, *codes of ethics* and *decision making and problem solving* are also important, while [12] mentions *regular communication with distributed team members* or *using collaborative technologies*.

Apart from those mentioned above, some other soft skills are required when managing a GSD project. The authors of [13] indicate that project managers require soft skills to manage the three principal challenges in a GSD setting: *coordination*, *communication* and *control skills*; and [14] maintains that the *team building skill* is also an important skill, along with the above-mentioned *decision making and problem solving*. Finally, [15] proposes that a project manager in a GSD project needs a *positive attitude*, *initiative and leadership*, *identification of competences*, *meeting management* and *requirement estimation and prioritization*, among many other skills.

All these contributions point to the convenience for a game to teach a combination of GSD and PM skills. Some of the most important skills that have been implemented in our game are *decision making and problem solving*, *communication, coordination* and *control skills*, *intercultural cooperation* and *time management*. It should be noted that some of the remaining soft skills (such as initiative and leadership) have not been included owing to the difficulty of implementing them in an SG, or because they may change depending on the company or country (i.e. codes of ethics). However, other soft skills may be included in future work.

### 2.2 Related Works

It should be noted that progress has also been made as regards teaching GSD concepts through the use of courses [16]. Some examples are [17], which teaches the role of communication in GSD, [18], which teaches coordination in GSD projects or [19], a GSD course on which different universities participate to learn how to work with the Scrum methodology in a GSD project. In addition, Damian et al. [20] propose a framework that teaches GSD skills in order to prepare graduates for software engineering.

In the case of SGs used to teach GSD we have found:

VENTURE [2], which consists of a virtual training environment in which the learners can interact with Virtual Agents from different cultures, acquiring skills related to teamwork, conflict resolution or socio-cultural differences. However, the game is provided through a single communication scenario with the Virtual Agents, signifying that the player cannot interact with a sandbox-focused environment as in our SG. Moreover, VENTURE focuses on software engineers' skills.

Another work considered is the GSD Sim [1], which allows players to manage both a globally-distributed SP, by assigning different work teams to a variety of locations around the world. Its main strength is that the interface via a world map is very useful. However, this game focuses only on the initial problems that could be found in a GSD project.

The SGs that help train future project managers are:

PMG-2D[5], which teaches some of the main concepts that a project manager should know when managing projects, identifying the stakeholders, creating a roadmap or hiring and dismissing employees. The player will acquire various skills, such as decision-making, risk-taking, and the management of costs, time or human resources. However, this SG is not able to deal with aspects that are relevant in a global setting, such as intercultural communication management.

The authors of [21] present the Software Engineering Simulation by Animated Models (SESAM), a simulator that allows students to manage a project, by reading and typing text, or hiring and firing employees. However, SESAM uses a purely textual interface and not a more user-friendly graphic interface.

SimSE [22] is another example of an SE simulation game that teaches PM, in which the player takes on the role of project manager. Players must manage a team of developers, aiming to complete a project by hiring and firing employees, assigning tasks, monitoring progress and purchasing tools. The players will learn the life cycle of a SP and decision-making skills. Nevertheless, SimSE lacks important aspects, such as visual and auditory feedback, and does not give its players a sense of immersion that makes it seem more like a simulation.

There is also the SG ProDec [23], which helps students to train in PM concepts. The player must confront problems that may arise throughout the life cycle of a SP, and tackle these by making decisions. Another version is ProDecAdmin[24], which allows

teachers to create game scenarios and overcome the weaknesses found in ProDec. The creation of game situations corresponds to an advantage; however, it does not consider aspects of GSD such as the existence of different cultures.

Other games, such as [25], help students to better understand value-based software engineering knowledge and how to manage the earned value in a SP. In this game, the players can change different parameters of the project, employ a strategy or apply certain processes, along with obtaining information about the project by visiting different rooms in the environment. However, it focuses too much on earned value and not on more general aspects, such as project coordination and control.

Another work considered is the SCRUMIA [26], which corresponds to an educational game that trains students in agile PM techniques by applying SCRUM. Thanks to SCRUMIA, students are able to remember the concepts and artefacts related to SCRUM. However, SCRUMIA corresponds to a manual paper & pencil game, while our purpose is the development of an SG, as it is more independent and flexible for the player.

## 3 GLOBAL-MANAGER

Our approach consists of a serious single-player game, in which the players have to take on the role of a project manager who has to manage a fictitious SP at a global scale from its beginning, continuing through the development process, until the delivery of the product. The player will be able to play different games by choosing from a set of available kinds of projects. The development process of the fictitious project is divided into two phases: the first in which the different parameters are configured before the start of the project (configuration phase), and the second corresponding to the simulation of a global PM situation. Each of these phases will be described below.

### 3.1 Configuration Phase

Players first choose the kind of project that they wish to manage, and then, configuring a series of initial parameters that will influence the evolution of the GSD project (Figure 1).

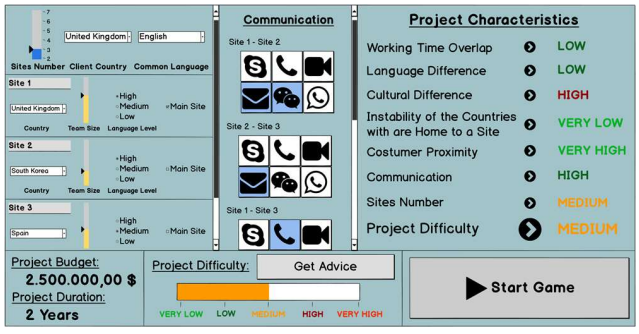


**Figure1: Project Configuration Interface**

In the first instance, the players should choose the number of sites, the country in which the customer is located, and the common language for communication. For example, let us imagine that a player chooses 3 sites, that the customer will be in the United Kingdom, and English will be the main language. Then, the player has to define the configuration of each site, and 3 small boxes will appear beneath this configuration. These boxes will show a definition of the country in which the project is located, the size of the team, the average level of the common language and, of course, the main site. The player could, for example, establish that the main site is in the United Kingdom, with a high level of English, that another site is in South Korea, with a high level of English, and that the last site is in Spain, where the level of English is medium, because it is supposed that the work team will not speak that language fluently. In the last step in the configuration, the type of communication that will exist between each of the sites and with the corresponding customer, needs to be managed. For example, telephone or Skype for synchronous communication, and email or forums in the case of asynchronous communication. In the example that is shown, the player could choose communication by email or via forums between the United Kingdom and South Korea, owing to the time difference between these countries, which would make synchronous communication difficult. The player might also decide to carry out communication by email between the United Kingdom and Spain, since the latter workgroup does not speak English fluently, and communication by phone could be tricky for them.

According to the data chosen by the player, the game calculates the project risks automatically, and those features that correspond to some of those mentioned in [27] will, therefore appear on the right-hand side of the screen. The values of these features are "*very high*", "*high*", "*medium*", "*low*" and "*very low*". The system will calculate these parameters in real time. They are:

- *Working Time Overlap*: if there is a large time difference between the countries of the sites, this parameter will be considered as low, since the time overlap is short, thus making it difficult to maintain synchronous communication. In the example, there is a site in South Korea, and the others are in Europe, signifying that the overlap may be *low*.
- *Language Difference*: if the level of the common language of the project on each site is high, this difference will be low. In our case, all the sites have a high level of English except one, signifying that this parameter will have a *low* value.
- *Cultural Difference*: this will be supported by Hofstede's proposal [28], which proposes a way in which to calculate the cultural difference between countries. If this difference is high, this parameter will, therefore, also have a high value. In our case, there are certain differences among the countries, signifying that the value will be *high*.
- *Instability of the Countries which are Home to a Site*: this parameter will simply calculate whether the sites are in a country with socio-political instabilities. In our case the value will be *very low*, as these countries are currently stable.
- *Customer Proximity*: if the main site is near the client's location, this parameter will be high. In our case the value is *very high*, since they are in the same country.
- *Communication*: this parameter checks that the configuration of the communication tools is correct, i.e. it checks whether

asynchronous communication has been chosen between two sites that have a high level of difference (cultural, geographical, or the level of the common language of the project). In these cases, synchronous communication could result in misunderstandings. It also monitors whether synchronous communication has been chosen if there is any difference between the sites. The more correctly the player chooses the type of communication between the sites, the higher the value of the parameter will be. In our case, with the subset of communications that has been defined, we can say that there is a *high* level of communication.

- *Sites Number*: this checks the number of sites defined, where the higher the value, the more difficult it will be to manage the project. Since, according to [29], a GSD project seeks to harness the overlap of the teams' work schedules in order to avoid delays in response times, increasing the number of sites will make communication more difficult, in addition to a greater number of cultures and personnel in control. Three sites can be considered to have a *medium* value.

This set of risk factors will help the player to understand how the configuration of the project may influence it success. There is also a "get advice" button, which will be used to receive useful hints, in order to improve their configurations. For example, if players obtain a *low* overlap, they can set up the sites in other countries with a higher time overlap. With regard to the levels of risk factors of the project, the system will also calculate a level of difficulty. This will allow the players to know at a glance if the configuration that has been established is easily achievable. In the example, the project difficulty set is *medium*. In addition to that, a project budget and duration will be offered by the system.

In general, this screen will enable the player to begin to become familiar with the different factors involved in a distributed project, since these factors may have an important influence on the project. The player can change them and discover how these factors affect the risks within the project. Once the configuration has been completed, the players proceed to the simulation of the fictitious project using the "Start Game" button.

## 3.2 Game Phase

As soon as the player has configured the fictitious project in the aforementioned window, defining each of the necessary parameters, the simulation of the development of the project will start in a new window. This new window is where the development of the project will take place, and where the player's character will appear, which is represented with a player's avatar.

This window is divided into two parts. The lower part is where the relevant information of the project will be located. In the central section of this part, the players can find initial information, such as the difficulty, the budget, or the development time. On the left, they will find: relevant information about the player, along with parameters that evolve the development of the project progresses, which are represented by progress bars; the current budget, which will go down as the project evolves; the remaining time to delivery, where, as the delivery date of the product approaches, the time will be reduced; the current progress of the project, and a level of stress related to the character, which will be explained later as it evolves. On the right-hand side, they will find data on the motivation of the sites, the level of communication, an evaluation of whether the site communicates with the others continuously, and the workload, depending on the percentage of tasks in the project that the site is in charge of.

In the upper part of the window the players will find the game screen, on which the character will be able to move about freely. As the evolution of the project, different objects will drop onto the screen. These objects represent situations that may occur in a GSD project, and the players will, therefore, have either a certain positive repercussion (objects with a green border) or a negative one (objects with a red border) as regards some of the parameters of the project. We have used several metaphors for this function, since the objects represented by a telephone will refer to communication events, the post-its will have to do with coordination events, and the magnifying glasses will represent the control events. In this way, if the player interacts with any of these objects, the description of the event and where the player will be able to solve it will appear in a small window. With regard to the stress bar, if at any time there are a large number of events on the screen, the bar will start to increase, and if not many appear, it may decrease. The game also provides the possibility of making this bar become smaller by pressing a button called "take a break". Some examples of events that may occur are shown below:

- *Communication event*: "You must communicate to a member of the project that there has been a change in the requirements of the product, making sure that this has been understood, however this individual belongs to a culture that always says "yes", since saying "no" is considered rude". Three different emails will be displayed in the window, and the player must choose the most appropriate option.
- *Coordination event*: "Two workers from a work team have certain dependencies between their tasks. Study which tasks they should coordinate and communicate this to them". A table will be shown with the tasks that each one must carry out, and the player must select those tasks that require coordination between both workers.
- *Control event*: "The requirements have been extended, and there is a new task to be assigned. Choose the most appropriate person to carry out this work from among these people". The player must select the most suitable person from among 3 (some are project workers and others are not). Their respective experience, strengths and weaknesses are indicated, along with the additional cost and, in the case of project workers, their work-load, their current task, etc.

The simulation of the fictitious project will continue until certain situations. In the first place, the players may exceed the stress level, the project may run out of budget, or reach project delivery time without the product being finished. In these cases, the players will have failed to manage the project. On the other hand, if the players

are able to complete the progress of the project on time and within budget, they will have been victorious in the game.

## 4 Conclusion and Future Work

This paper presents a SG denominated as GLOBAL-MANAGER, which is intended to train students and practitioners. The aim is for these individuals to become familiar with different PM concepts and gain more in-depth knowledge of GSD, all of which is done in an entertaining manner. The players will also be able to receive training and acquire important soft skills through simulated situations especially as regards those situations related to the communication, coordination and control of a GSD project. The players will additionally train in problem solving, by being submitted to a certain level of stress, by continuously dropping objects onto the screen for them to deal with.

In future work we will evaluate the GLOBAL-MANAGER, this evaluation will take place next year in the subject of Software Engineering, with the objective of attaining initial feedback from these students. A second evaluation, such as that which took place in [31] will later be performed using, in this case, PM practitioners.

## ACKNOWLEDGMENTS

This work has been funded by the G3SOFT project (Consejería de Educación, Cultura y Deportes de la Junta de Comunidades de Castilla La Mancha, y Fondo Europeo de Desarrollo Regional FEDER, SBPLY/17/180501/000150) and by the BIZDEVOPS-GLOBAL project (Ministerio de Ciencia, Innovación y Universidades, y Fondo Europeo de Desarrollo Regional FEDER, RTI2018-098309-B-C31).

## REFERENCES

[1] J. Noll, A. Butterfield, K. Farrell, T. Mason, M. McGuire, and R. McKinley, "GSD Sim: A Global Software Development Game," *Proc. - Int. Comput. Softw. Appl. Conf.*, vol. 18-21-Augu, pp. 15–20, 2014.

[2] M. J. Monasor, J. Parkes, J. Noll, A. Vizcaíno, M. Piattini, and S. Beecham, "Global software development education: A commercial perspective from a case study," *Proc. - 2014 IEEE 9th Int. Conf. Glob. Softw. Eng. ICGSE 2014*, pp. 173–182, 2014.

[3] S. Beecham, T. Clear, D. Damian, J. Barr, J. Noll, and W. Scacchi, "How Best to Teach Global Software Engineering? Educators Are Divided," *IEEE Softw.*, vol. 34, no. 1, pp. 16–19, 2017.

[4] S. Beecham, T. Clear, J. Barr, M. Daniels, M. Oudshoorn, and J. Noll, "Preparing Tomorrow's Software Engineers for Work in a Global Environment," *IEEE Softw.*, vol. 34, no. 1, pp. 9–12, 2017.

[5] J. E. N. Lino, M. A. Paludo, F. V. Binder, S. Reinehr, and A. Malucelli, "Project management game 2D (PMG-2D): A serious game to assist software project managers training," *Proc. - Front. Educ. Conf. FIE*, vol. 2015, 2015.

[6] A. Calderón, M. Ruiz, and R. V. O'Connor, "A multivocal literature review on serious games for software process standards education," *Comput. Stand. Interfaces*, vol. 57, pp. 36–48, 2018.

[7] D. Zhao, A. Chis, G.-M. Muntean, and C. H. Muntean, "a Large-Scale Pilot Study on Game-Based Learning and Blended Learning Methodologies in Undergraduate Programming Courses," *EDULEARN18 Proc.*, vol. 1, pp. 3716–3724, 2018.

[8] M. Kosa, M. Yilmaz, R. V. O'Connor, and P. M. Clarke, "Software engineering education and games: A systematic literature review," *J. Univers. Comput. Sci.*, vol. 22, no. 12, pp. 1558–1574, 2016.

[9] M. J. Monasor, A. Vizcaíno, and M. Piattini, "Training Global Software Development skills through a simulated environment," *ICSOFT 2010 - Proc. 5th Int. Conf. Softw. Data Technol.*, vol. 2, pp. 271–274, 2010.

[10] M. J. Monasor, A. Vizcaíno, M. Piattini, and I. Caballero, "Preparing students and engineers for global software development: A systematic review," *Proc. - 5th Int. Conf. Glob. Softw. Eng. ICGSE 2010*, pp. 177–186, 2010.

[11] I. Bosnić, I. Čavrak, and M. Žagar, "Assessing the impact of the distributed software development course on the careers of young software engineers," *ACM Trans. Comput. Educ.*, vol. 19, no. 2, pp. 1–27, 2019.

[12] M. Paasivaara, C. Lassenius, D. Damian, P. Raty, and A. Schroter, "Teaching students global software engineering skills using distributed Scrum," *Proc. - Int. Conf. Softw. Eng.*, pp. 1128–1137, 2013.

[13] J. M. Verner, O. P. Brereton, B. A. Kitchenham, M. Turner, and M. Niazi, "Risks and risk mitigation in global software development: A tertiary study," *Inf. Softw. Technol.*, vol. 56, no. 1, pp. 54–78, 2014.

[14] Z. Mansor, K. Sutling, S. Widyarto, S. Lecthumanan, and N. Arshad, *Understanding of Project Manager Competency in Agile Software Development Project: The Taxonomy*, vol. 339. 2015.

[15] J. Ramos, A. Sanz-Esteban, J. García Guzmán, and A. de Amescua, "Skills and abilities for working in a global software development team: A competence model," *J. Softw. Evol. Process*, vol. 26, Mar. 2014.

[16] T. Clear *et al.*, *Challenges and recommendations for the design and conduct of global software engineering courses: A systematic review*. 2015.

[17] M. Kuhrmann and J. Münch, *Distributed Software Development with One Hand Tied Behind the Back: A Course Unit to Experience the Role of Communication in GSD*. 2016.

[18] M. Nordio *et al.*, "An experiment on teaching coordination in a globally distributed software engineering class," *2014 IEEE 27th Conf. Softw. Eng. Educ. Training, CSEE T 2014 - Proc.*, pp. 109–118, 2014.

[19] D. Damian, C. Lassenius, M. Paasivaara, A. Borici, and A. Schröter, "Teaching a globally distributed project course using Scrum practices," *2012 2nd Int. Work. Collab. Teach. Glob. Distrib. Softw. Dev. CTGDSD 2012 - Proc.*, pp. 30–34, 2012.

[20] D. Damian, A. Hadwin, and B. Al-Ani, "Instructional design and assessment strategies for teaching global software development: A framework," *Proc. - Int. Conf. Softw. Eng.*, vol. 2006, pp. 685–690, 2006.

[21] A. Drappa and J. Ludewig, "Simulation in software engineering training," *Proc. - Int. Conf. Softw. Eng.*, pp. 199–208, 2000.

[22] E. O. Navarro and A. Van Der Hoek, "SimSE: An interactive simulation game for software engineering education," *Proc. Seventh IASTED Int. Conf. Comput. Adv. Technol. Educ.*, pp. 12–17, 2004.

[23] A. Calderón and M. Ruiz, "ProDec: a serious game for software project management training," *ICSEA 2013 - Proc. 8th Int. Conf. Softw. Eng. Adv.*, no. c, pp. 565–570, 2013.

[24] A. Calderón, M. Ruiz, and R. O'Connor, *ProDecAdmin: A Game Scenario Design Tool for Software Project Management Training*. 2017.

[25] A. Jain and B. Boehm, "SimVBSE: Developing a game for value-based software engineering," *Softw. Eng. Educ. Conf. Proc.*, vol. 2006, pp. 103–111, 2006.

[26] C. G. Von Wangenheim, R. Savi, and A. F. Borgatto, "SCRUMIA - An educational game for teaching SCRUM in computing courses," *J. Syst. Softw.*, vol. 86, no. 10, pp. 2675–2687, 2013.

[27] A. Vizcaíno, F. García, J. C. Villar, M. Piattini, and J. Portillo, "Applying Q-methodology to analyse the success factors in GSD," *Inf. Softw. Technol.*, vol. 55, no. 7, pp. 1200–1211, 2013.

[28] G. Hofstede, *Culture and Organizations*, vol. 10, no. 4. 1980.

[29] E. Ó. Conchuir, P. J. Ågerfalk, H. H. Olsson, and B. Fitzgerald, "Global software development: Where are the benefits?," *Commun. ACM*, vol. 52, no. 8, pp. 127–131, 2009.

[30] L. Silverman and R. M. Felder, "Learning and Teaching Styles in Engineering Education," *J. Eng. Educ. -Washington-*, vol. 78, pp. 674–681, Jan. 1988.

[31] A. Vizcaíno, F. García, I. G. R. De Guzmán, and M. Á. Moraga, "Evaluating GSD-aware: A serious game for discovering global software development challenges," *ACM Trans. Comput. Educ.*, vol. 19, no. 2, pp. 1–23, 2019.