

### Towards to a new vision of Global Software Development

José Ángel Cuadrado Mingo, Roberto Esteban Santiago

Departamento de Informática

Universidad Carlos III de Madrid

Madrid - España

{joseangel.cuadrado, roberto.esteban}@uc3m.es

**Abstract:** *We live in a globalized world in which people communicate with other and interact with each. Among these interactions, we can find people that work with other people that live in the other extreme of the world. In Software Engineering, this way of working is called Global Software Development. Many professionals in the sector have talked about this, but there are little works collecting information and work in a unique vision of this technique. For this, we analyze in this paper various definitions of Global Software Development and we offer our own definition of this. Also, we compare the advantages and drawbacks of this technique and propose the collaboration between Global Software Development and other research areas.*

**Keywords:** *Global Software Development, definition, advantage, drawback, e-learning, gamification, coaching, agile methodologies.*

## 1. Introduction

Nowadays, the world is more and more globalized. This is due to the large quantity of ways to communicate between people [1]. While a couple of decades ago, the more common way of communication was postal mail (taking several days to deliver within a country, and several weeks across continents), now, the most common way of communication is the e-mail and other Internet services, by which a person can communicate with another that is thousands of miles away in a few seconds.

As today it is possible to communicate in a few seconds with the opposite extreme of the world, this allows doing other things, such as learning and working in the distance. And, if we talk about work in the distance, we can also speak of global projects, or in the context of software engineering, of Global Software Development.

The origin of the Global Software Development dates back to the decade of 1970. In these years, there was a big demand of software applications, but the number of software engineers was insufficient to cover all the work. For this reason, the organizations dedicated to information systems started to contract external engineers that could help them [2].

Moreover, there were other problems. On the one hand, there were economic problems; on the other hand, the problem of insufficiency of engineers caused delays in the delivery of projects. This resulted in that organizations searched some solution for both economic and temporal problems. The solution found involved associating with other companies and engineers of other countries [3], such that it was possible to hire cheaper engineers and coordinate the work with other teams in different countries and different time zones.

In one of the publications of Erran Carmel [4], he shows what the influential factors for the establishment of Global Software Development were. One of these factors was the fusion between IT companies. These organizations want to complement their product lines and fuse with other companies as a way to enter in

new markets and meet this objective. Due to this, the development teams must collaborate with other teams and work like a unique global development team.

Another factor was the globalization of the company: other global organizations prefer a global provider instead of a negotiation with several small local companies. It is also important that the company has a market approach, such that it can improve its reputation, clients can know the organization, people may invest in it, and governments can provide tax benefits.

An important point for Global Software Development is the possibility of access to the best developers. If an organization wants to develop the best software products, it must hire the best developers, and these don't have to be found in a unique location.

But the two more important factors are the reduction of development costs and time to market. On the one hand, the companies located in countries with elevated costs outsource their tasks to teams located in countries with cheaper costs, like China, Brazil, India or countries of East Europe. On the other hand, the dispersion of the software developers around the world enables software to be developed 24 hours a day and reduce the development cycles (due to the increase the amount of time spent daily software development).

And why is it important to study Global Software Development? Different studies show some reasons for studying it. A research of Kwan-Sik Na [5] specified that 75 percent of the US companies have outsourced their work to other countries.

Another survey of the SIIA (Software and Information Industry Association) [6], involving the participation of 114 firms, shows that Global Software Development initiatives are advancing rapidly (62 percent of these companies began their initiatives less than three years ago -on the date on which the study was conducted) and has a positive net impact on both the revenues and profitability of software companies (none of companies who current work with global development teams report a negative impact on the overall revenues, and 87 percent report a positive impact on their profits).

Also, many companies who used Global Software Development experienced process improvement. About companies with no global development operations, three in four companies not currently working offshore expect to initiate this activity in the future -the vast majority of those planning to go global intend to do so within 18 months-.

The paper is structured as follows: section 2 presents a review of the definitions proposed by different authors who have written on the subject; section 3 discusses the various advantages and disadvantages of Global Software Development; section 4 analyzes the collaboration and integration between Global Software Development and different research areas; and section 5 concludes with the observations presented in this paper.

## **2. Definitions**

Now, there are a lot of definitions about Global Software Development, however, it is very difficult to find a common definition in the field of research. For this reason, we will make a compilation of the literature, thus being able to find all aspects of the definitions of GSD, to make our own definition correct on Global Software Development.

First, Allen, in 1984, defined *“Global Software Development (GSD) as software development that uses teams from multiple geographic locations. In some cases, these teams may be from the same organization; in other cases, there may be collaborations or outsourcing that involves different organizations. These teams could be within one country (in fact, evidence suggests that once teams are separated by more than 50 meters, further distance is immaterial) or on other sides of the world”* [7]. This definition, despite being of 1984, already had in mind that the Global Software Development teams could be located at one or more sites.

On the other hand, Carmel, in 1999, defined *“One specific case of a 'hybrid' team is a globally distributed software development team. Globally distributed projects involve two or more teams working together from different geographical locations to accomplish common project goals. In addition to geographical dispersion, globally distributed teams face time-zone and cultural differences that may include different languages, national traditions, values and norms of behavior that may greatly reduce the extent of socialization between remote counterparts”* [8]. In this definition it is noteworthy that the teams are in multiple geographic locations, but in our opinion this is not strictly necessary, because in Global Software Development teams may be located in the same geographic locations.

From 2001, we have found many definitions about GSD; we found a new definition of Carmel with Agarwal: *“Global software development means that the software development is dispersed along several sites (software development centers) that could be located in different countries and even continents. A global software team executes the activities collaborating on a common software development project”* [9]. As in the previous definition, this definition also identifies that the development team must be in multiple geographic location. Additionally, it correctly identifies that the global software team executes the activities of a common project.

In that year, Audris Mockus and James Herbsleb also defined GSD: *“global software development (GSD), in which the software development activities are distributed across multiple sites”* [10]. This brief definition also takes into account that the development team must be in multiple sites.

The last definition found in 2001 is of Herbsleb and Moitra that defined the GSD as *“organizations have increasingly been adopting a geographically distributed software development paradigm, which characterizes software development as a multisite, multi-culture, and global undertaking”* [11]. This short definition identifies the main aspects about GSD: development as a multisite, multi-culture, and global undertaking.

In the paper *A Framework for Overcoming Supplier Related Threats in Global Projects* there is a definition about GSD, defined by Karolak, Sahay, et al.: *“The concept of global software development (GSD) addresses distribution of common software life cycle activities among teams separated by various boundaries, such as contextual, organizational, cultural, temporal, geographical, and political. Risks associated with these boundaries make managers struggle with pressures unique to this type of environment”* [12,13]. In this definition, the researchers identified correctly that the teams are separated by various boundaries, but these boundaries may be contextual, or geographical, and not only geographical, as other researchers identified.

Later, in 2005, Ågerfalk et al., defined as Global Software Development *“All activities of the software lifecycle where the project involves actors who are dispersed across at least two locations which are separated by country or continental borders, and typically across multiple time zones with a degree of socio-*

*cultural distance amongst the actors*" [14]. In this definition as in the Carmel one, it is identified that the team must be dispersed across multiples sites, with different time zones with a degree of socio-cultural distance.

Finally, the last definition found is in 2011, by Darja Šmite and Claes Wohlin: *"Many software companies are responding by shipping their development work to offshore locations, either through outsourcing projects in whole or in parts to a third party or through insourcing to an offshore site within their own organizations"* [15]. Noteworthy in this definition that may be outsourcing and insourcing, these aspects are not taken into account in the other definitions.

Other researches, refer the GSD with virtual teams, for example in 2009, Bird et al., identified the software development team as *"A software development team can geographically be distributed within a campus, city, country, or globally"* [16]. In this definition about software development teams, it is identified what must be distributed a team a project of GSD.

In 1988, Curtis, Krasner and Iscoe, defined virtual team as *"a popular structure in software development for several reasons: they provide access to lower-cost labor as well as access to a range of disciplines and technical specialties"* [17]. In this definition, the researchers identified some of the most important features in the global software development teams, which are the cost reduction and the access to a range of disciplines and technical specialties.

On the other hand, Townsend, DeMarie and Hendrickson, in 1998, defined virtual team as *"Virtual teams are composed of geographically distributed coworkers linked though information technologies to achieve an organizational task"* [18]. According to our opinion, this definition is not accurate enough, as the virtual team is not strictly required to be geographically distributed.

In the paper *Occurrence and effects of leader delegation in virtual software teams*, of Suling Zhang et al., of 2009, defined virtual team as *"The structure that is typically in place does not constitute virtuality for each and every team member but, rather, distributed teams in which some subsection of the software development team is co-located and other sections are virtual [...]. The overall team is working on the same software product, but the work has typically been compartmentalized in some way so that each co-located portion of the team has specific assignments. However, the work is such that there is a need for continued communication between each of the non-co-located portions of the team to resolve integration issues"* [19]. In these definitions, the researchers identified correctly that the teams can be co-localized or not. Furthermore, so also correct, each team is assigned a project some related tasks.

With all the definitions found, we will get the consistent characteristics, from our point of view, and we will propose our own definition. It is worth noting that this definition should not be too extensive, but rather a brief one which is easy to understand. For definition, first, we must point it out that we want to develop a software project. Furthermore, the teams members may or may not be in the same geographical location and therefore can have different time zone and / or socio-cultural scopes. Then, with these features greater performance and effectiveness can be achieved for the project. In summary, GSD is the development of a software project in which the members could be in different places, time zones or even having different socio-cultural scopes, in order to achieve a higher performance and effectiveness in project implementation.

### 3. Advantages and drawbacks y Negocios

Actually, there are a lot of drawbacks to perform a Global Software Development Project, because the person is resistant to change and does not like the change in the organization of the company. However, there are not only drawbacks, there are many advantages about Global Software Development, which are the following:

A distributed Software Configuration Management (for managing the system component versions) reduces miscommunication because it enforces a common work process and a common view of the project [9]. With this advantage, everything will be much more documented, and then later will be more easily to solve the bugs.

Similarly the advantage of the previous, Martha Maznevski and Kathy Chudoba found that effective virtual teams had a *“deep rhythm” of regular team meetings, both face-to-face and over distance. Of course, a meeting is but communication formalism. Communication need not always occur within a formal, hierarchical configuration. When global software teams collaborate on innovative projects, informal channels of coordination —or lateral channels— are critical. They “help developers fill in the details in the work, handle exceptions, correct mistakes...”* [9].

For example, if an organization can manage daily handoffs of work between remote sites and focus attention around the clock on critical-path tasks, it is possible to take advantage of widely dispersed time zones. We could theoretically extend the productive hours of the day from the current 8- to 10- hour norm to somewhere near the limit of 24 hours [11]. In this way, gained time zone effectiveness and reduced cost in various countries. This advantage is one of the big reasons that may motivate companies for doing a project using GSD, as because it takes less time to complete the project on time effectiveness, the project cost is also lower.

In the paper *A review of awareness in distributed collaborative software engineering*, it is stated that distributed software development offers a number of theoretical benefits, including shortened time-to-market cycles and rapid response to customer needs since collaborations are independent of time and space. The mixing of developers with different cultural backgrounds can help trigger new ideas. Finally, distributed software development benefits from access to a larger qualified resource pool with the promise of reduced development cost [20]. One notable advantage of all these previous mentioned paper, is that the mixing of developers with different cultural backgrounds can help trigger new ideas, as this idea is an important idea and no other paper mentions, and the company is an aspect that may differ from your competitors.

This continues to be facilitated by the availability of well educated and technically competent software engineers in low-cost centers in Eastern Europe, Latin America, India and the Far East [9,21]. It is a commonly held belief that these savings can be coupled with the opportunity for round the clock development facilitated by the temporal difference between remote development locations. The logic underpinning this approach is that these two factors can facilitate competitive pricing and reduce time to market, thus enabling companies to compete more effectively by gaining, expanding or maintaining their market share [11,22].

According to the paper A Structured Approach to Global Software Development, 10 factors were determined which were directly relevant and needed to be specifically addressed in order to establish and facilitate the operation of globally distributed virtual teams. These factors are summarized as follows [22]:

1. Understand why, at what cost and what risk a distributed strategy is undertaken
2. Provision of effective infrastructure and documented process
3. Requirement to effectively establish the teams
4. Implement an efficient distributed team project management strategy
5. Ensure the development of common goals, objectives and rewards
6. Need for the clear definition of roles and responsibilities
7. Address issues related to culture, communication, motivation and fear
8. Ensure provision of adequate training and knowledge transfer
9. Facilitate and monitor the operation of collaborative and supportive teams
10. Document and leverage lessons learned

Following these ten factors, a GSD project would be much easier to carry out; thus, the quantity of GSD projects was increased.

On the other hand, another factor very important is that English is mandatory, and language classes are provided in most non-native speaking countries to leverage skills [23]. Furthermore English is spoken by most people in the world.

Also, it is now easier to do these projects because electronic workspaces based on groupware systems can provide a supportive infrastructure for an evaluation in the context of GSD [14].

A drawback of the Global Software Development is that some of the places where some virtual work teams are not enough qualified. Thus, in India, most of the large IT organizations provide training programs for their new employees [24]. Then, the drawback is not so serious because unskilled teams are formed.

Many companies do not realize GSD projects because there are many drawbacks, however now there are a lot of techniques to reduce these drawbacks, as these [9]:

- Intensive collaboration
- Cultural distance
  - Bridgehead
  - Internalization of Foreign Entity
  - The cultural liaison
  - Language
- Temporal distance

So far, we have seen many of the advantages of a GSD project, however there are many drawbacks. Below are shown the drawbacks found in the literature on the GSD.

Working on a globally distributed project means operating costs for planning and managing people, along with language and cultural barriers. It also creates jealousy as the more expensive engineers (who are afraid of losing their jobs) are forced to train their much cheaper counterparts [23]. These drawbacks are important, because creating jealousy between teams can disastrously affect to the project development.

The paper *Global Software Development shows many drawbacks to perform a GSD project* [11]. Based on statistical modeling of development interval and on survey results, that multisite development tasks take much longer than comparable collocated tasks and that communication and coordination play major roles in this delay. Moreover, deciding how to divide up the work across sites is difficult. Solutions are constrained by the resources available at the sites, their levels of expertise in various technologies, the infrastructure, and so on.

In the same paper, another fundamental challenge is the organization's resistance to GSD. This resistance often surfaces because of misalignment between senior and middle management on the intent and perceived benefits of GSD. Many individuals might believe their jobs are threatened, experience a loss of control, and fear the possibility of relocation and the need for extensive travel [11]. This challenge is important because if the team members work under pressure, can affect team effectiveness.

GSD requires close cooperation of individuals with different cultural backgrounds [11]. This aspect can be a drawback or an advantage, because although some people do not like this aspect, there may be other people who like it and want to learn about other cultures.

On the other hand, previous qualitative research suggests that multi-site development may increase development cycle time [25].

Gary Anthes presents a telling example of poor communication in a Global Software Development project, when a tester interpreted a spacebar instruction as a "b-l-a-n-k," clearly not the intended message of the sender. Furthermore, is more difficult to share knowledge with no co-located teams, because for example if a team member has a question and another member is co-located can performing the question.

In the paper *A review of awareness in distributed collaborative software engineering* also identifies some of the drawbacks of the GSD. Developing and maintaining such awareness is more difficult in distributed software teams than co-located ones. This is because the awareness information required during software collaboration is tacit, inherent, dynamic and contextual and therefore extremely challenging to distribute automatically. It is tacit since most of what developers do in collaboration spaces builds from experience, skills, heuristics and interactions that can hardly be documented and inherent since this knowledge is deeply bound to these developers. Its dynamic nature stems from the ever changing state of software projects. Finally, the relevance of such information varies across differing project contexts: developer, task and artifact [20].

Studies have revealed the problems caused by these particular attributes of distributed teams. These include poor visibility and control of remote resources; inadequate communication, collaboration and coordination across distributed teams; diminished trust; and, lack of shared context awareness [20].

On the other hand, another researches identified that some of the difficulties encountered include such factors as the problem of understanding requirements, testing of systems and the coordination of these types of projects [8].

These difficulties are further compounded by cultural and language differences, lack of communication, geographical and temporal distance from team members and the customer, different process maturity levels, development and testing tools, standards, technical ability and experience. As a result the management of globally distributed software development projects has been recognized as a difficult and complex task [8].

Distance has been identified as a key problem and by its very nature introduces barriers and complexity into the management of globally distributed projects [8].

All drawbacks above are difficulties for project managers, and managing all these is a challenge for them. Furthermore, although some advantages reduce project development time, the drawbacks increase management time because is very difficult to coordinate all this.

With all these advantages and drawbacks, the companies could decide whether it would be appropriate to undertake a Global Software Development project or not. To make this decision, the company must consider future projections having it, because the company may want or not have customers in other countries.

Researcher	Advantage	Researcher	Drawback
[9]	Reduces miscommunication	[23]	Costs for planning and managing people
[11]	Gain time zone effectiveness	[23]	Language and cultural barriers
[11]	Reduced cost in various countries	[23]	Creates jealousy as the more expensive engineers
[20]	Shortened time-to-market cycles	[11]	Multisite development tasks take much longer than comparable co-located tasks
[20]	Rapid response to customer needs	[11]	Communication and coordination play major roles in this delay
[23]	English is mandatory	[11]	To divide up the work across sites is difficult
[11,22]	Expanding or maintaining their market share	[11]	Many individuals might believe their jobs are threatened, experience a loss of control, and fear the possibility of relocation and the need for extensive travel
[9]	There are a lot of techniques to reduce drawbacks	[20]	Poor visibility and control of remote resources
		[20]	Inadequate communication, collaboration and coordination across distributed teams
		[20]	Diminished trust; and, lack of shared context awareness
		[8]	Problem of understanding requirements, testing of systems and the coordination of these types of projects

Table 1. Advantages and Drawbacks

#### 4. Areas research

Global Software Development is a great tool for the software development due to the several advantages that have been mentioned before. However, although this is referent to software, it can apply to many fields, inferring this technique as globalization of processes development. In this context, there are other research areas where globalization and software development can be integrated. Thanks to globalization, it is possible that many tasks can be performed, like the education or the team-working.

In reference to education, some comments will be made about e-learning. It is possible defining e-learning as *"all forms of electronic supported learning and teaching, which are procedural in character and aim to effect the construction of knowledge with reference to individual experience, practice and knowledge of the*

learner. *Information and communication systems, whether networked or not, serve as specific media [...] to implement the learning process.*" [26]. In other words, e-learning is the ability to learn through online services and mobile devices.

This is an example of the advantages of globalization, which allow students learning without being in a physical class. Thus, it enables the people who are working or can't move learning without the necessity to go to the class physically.

In the same line, but in the opposite direction, it is necessary to teach students the concept of globalization, because in this world, more people are getting connected every day. If the focus is Software Engineering, instead of teaching the concept of globalization, then Global Software Development must be taught.

In the university, there are various courses which teach concepts about software development and Global Software Development. Also, studies mention that it is necessary that the courses are performed together by different universities so students can interact with other students from different cultures [27]. However, there are difficulties for obtain an adequate level of coordination and collaboration between the universities that allow interaction among students [28].

It is also important that students can practice the theory. Thus, there are some universities that organize practical activities with other universities in different countries in which students communicate with others through e-mail, telephone and instant messaging, and so, they can simulate what happen in the company [29].

Another possibility is to learn Global Software Development in business environment. Although this is uncommon, there are some experiences in which was given a training course for the employers of a company -the course included topics such as communicative practices, cultural differences, coordination and confidence among the participants- [30].

Because Global Software Development is a new technique, it has been shown how the concept of globalization and learning are taught in the context of software engineering. Now, it is possible to know how to integrate other research areas and Global Software Development to motivate staff and increase the benefits that brings this technique so that, ultimately, IT companies can improve its software development department.

Another research area in which is possible to integrate Global Software Development is gamification. Gamification is *"the use of game design elements in non-game contexts"* [31], like team-working or processes development, with the goal of adopt some behavior in people.

This technique may encourage people to do boring tasks. It is possible to get this by maximizing the individual competition (for example, when a person plays to a videogame, he or she returns to play for overcome his or her score), or, over all, maximizing the social competition (when a person wants to overcome the score of the rest of people), in which should get the best performances. However, this competence can produce negative effects among the employers, so it is necessary control an excessive competence.

Gamification will have great importance in the future as shown by a study of the consultant and technological research Gartner, that predicts *"over 70 percent of Global 2000 organizations will have at least one gamified application by 2014"* [32] and *"50 percent of innovations will be gamified by 2015"* [33].

But, how gamification can help to Software Engineering? The main contribution of gamification is to motivate people for improving the performance of a process in which is participant. Regarding Global Software Development, this technique can gamificate tools used for the coordination of people or teams wherever, repositories that are used by the team, process manager information used to collect the activity of team members, or tools for evaluation and testing.

Another example of using gamification in a global context is encouraging the social competence between different groups from around the world who belong to the same development teams. Thus, it motivates different development groups to be the best and improve the performance of all team.

A real example of this happened with the tool of cloud storage Dropbox. An initiative of Dropbox, called "*Space Race*" [34], made that universities from around the world compete for achieving the highest number of students enrolled in the web service. As a reward, a higher number of students of a university resulted in a higher storage capacity for those students. Therefore, each university students was motivated in order to its university could win the competition.

Another way to motivate people and employers of a company is coaching. Coaching allows to "*develop and maximize what is best in each individual, keep the individual focused and aware of new opportunities for growth and development, identify and change the thoughts or beliefs that limit the development of the individual, and reconcile private and professional life*" [35].

There are a lot of kinds of coaching: sports coaching, personal coaching, executive coaching, business coaching, etc. Coaching is based in the relation between the coach and the sportsman (coaching is born in sports context), where the coach try to improve the qualities of the sportsman. This technique was moved to other contexts, like the personal and the business. Personal coaching "*is aimed at clarifying values and visions, as well as the establishment of new objectives and actions for the individual to lead a more satisfactory life*" [35].

Otherwise, the business and executive coaching are oriented to improve the interpersonal and communication skills, leadership development, balance professional and private sphere, planning and strategy capacity development, conflict management and increase productivity.

As already stated, coaching allows motivating people. In the case of business coaching, this motivation is oriented to people are working in a company, from the highest office to the lowest office. However, this technique permits the development of general competencies and the improvement of learning and performance instead of develop technical competencies and professional careers how doing other techniques, like mentoring.

Sometimes, people confuse the coaching and mentoring concepts and use both interchangeably for referring the same one. However, the definition of mentoring can be summarized in "*the matching of a novice with a more experienced person in the same role*" [36], or in other words, some experienced person teaches to a new employee in the company how develop his -or her- career. Note is the fact that mentoring is one of the practices of one of maturity levels of People-CMM, a maturity framework that helps address the critical people issues in an organization [37].

Talking about Global Software Development, there are several studies that show the benefits of coaching in this area. In various studies of Christof Ebert [23,38], it is concluded that it is necessary a certain level of coaching in global projects for reduce cost of non-quality -the time needed to detect and correct defects-

and improve the performance of the inexperienced engineers. In another study of Maria Paasivaara [39], she investigates the effect of coaching in three global projects when these introduce agile practices in their IT organizations, and concludes that change management can succeed if experienced coaches collaborate from an early phase of the project. Mentoring is also important in Global Software Development, because it may be a way to bridge the gap between people, and solve cultural issues in the organizations [40].

One of the principal issues in Global Software Development is the different cultures of the people, as show Geert Hofstede in his publications [41] (it is important to mention his website [42], where it is possible find a comparison between several countries about the dimensions that include his study), but coaching in short and medium term, and mentoring in long term, can help to solve this and others challenges of Global Software Development.

With reference to the investigation of Maria Paasivaara mentioned before, agile practices in global projects was discussed. Thereby, it is also a good idea to talk about agile methodologies in the context of Global Software Development. The principles of agile methodologies allow teams to develop software and responding quickly to changes that may arise during the project [43].

Going into further detail, agile methodologies, through "*Agile Manifesto*" [44], reward the team interactions, the correct functionality of software, collaboration with the client and respond to the changes. Therefore, everyday it is more common the use of agile methodologies in the organizations. Moreover, there are a lot of agile methodologies that can use the organizations, but the main methodologies are Scrum, eXtreme Programming, Kanban and Crystal.

Due to rapid growth of agile methodologies, and the relation of this with the dealing of people (team-working, collaboration with the client, etc.), it is also important linking agile methodologies with globalization and, in particular, with Global Software Development. However, how can these areas be linked when agile methodologies need a face to face communication and in globalization people can be thousands of miles away?

Thanks to agile methodologies, it is possible to improve the communications in global projects [45-47]. Also, the shorts iterations of agile methodologies bring various benefits, like transparency of work progress to all partners. While the customer can monitor real progress, the distributed developers can get instant feedback on their work, which is motivating [46].

The collaboration between agile methodologies and Global Software Development may be beneficial for the latter and can help meet the challenges of this; only it is necessary a balanced level between these areas [47].

## 5. Conclusions

In this paper, we have analyzed different definitions of this technique by several authors. From the definitions that have been presented, we have elaborated our own definition, including various aspects like the situation and context of the team members and expected results using this process.

Also, we performed a comparison between the advantages and drawbacks of Global Software Development in which discusses all the details of this.

Finally, we explore different knowledge areas which can apply in Global Software Development for improve the performance and the use of this in the companies. In particular, we could see how this technique is integrated with e-learning, gamification, coaching and mentoring, and agile methodologies.

For future research, we have various ideas for continue this work. On the one hand, it is possible to make a study of the problems and its possible solutions of Global Software Development; too it is possible to make a systematic review. On the other hand, a future line is to deepen in the different research areas and write various papers about the collaboration between Global Software Development and each one of those.

## References

- [1] Azizi Yahaya, Jamaludin Ramli, Shahrin Hashim, Mohd. Ali Ibrahim, Raja Roslan Raja Abd Rahman, and Noordin Yahaya, "Discipline Problems among Secondary School Students in Johor Bahru, Malaysia," *European Journal of Social Sciences*, vol. 11, no. 4, 2009.
- [2] Jae-Nam Lee, Minh Q. Huynh, Kwok Ron Chi-wai, and Shih-Ming Pi, "The Evolution of Outsourcing Research: What is the Next Issue?," *Proceedings of the 33rd Hawaii International Conference on System Sciences*, 2000.
- [3] D. Šmite, C. Wohlin, T. Gorschek, and R. Feldt, "Empirical evidence in global software engineering: a systematic review," *Empir Software Eng*, vol. 15, no. 1, pp. 91–118, Feb. 2010.
- [4] E. Carmel, *Global software teams: collaborating across borders and time zones*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1999.
- [5] K.-S. Na, J. T. Simpson, X. Li, T. Singh, and K.-Y. Kim, "Software development risk and project performance measurement: Evidence in Korea," *Journal of Systems and Software*, vol. 80, no. 4, pp. 596–605, April 2007.
- [6] SIIA, "SIIA Global Software Development Survey Report," Dec. 2006.
- [7] R. Sangwan, N. Mullick, and M. Bass, *Global Software Development Handbook*. CRC Press, 2007.
- [8] I. Oshri, 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*, vol. 16, no. 1, pp. 25–49, Mar. 2007.
- [9] E. Carmel and R. Agarwal, "Tactical approaches for alleviating distance in global software development," *IEEE Software*, vol. 18, no. 2, pp. 22–29, Apr. 2001.
- [10] A. Mockus and J. Herbsleb, "Challenges of global software development," in *Software Metrics Symposium, 2001. METRICS 2001. Proceedings. Seventh International, 2001*, pp. 182–184.
- [11] J. D. Herbsleb and D. Moitra, "Global software development," *IEEE Software*, vol. 18, no. 2, pp. 16–20, Apr. 2001.
- [12] D. W. Karolak, *Global Software Development: Managing Virtual Teams and Environments*, 1st ed. Los Alamitos, CA, USA: IEEE Computer Society Press, 1999.
- [13] S. Sahay, B. Nicholson, and S. Krishna, *Global IT Outsourcing: Software Development across Borders*. Cambridge University Press, 2003.
- [14] M. Ali Babar, "A framework for groupware-supported software architecture evaluation process in global software development," *Journal of Software: Evolution and Process*, vol. 24, no. 2, pp. 207–229, 2012.
- [15] D. Smite and C. Wohlin, "Strategies Facilitating Software Product Transfers," *IEEE Software*, vol. 28, no. 5, pp. 60–66, Oct. 2011.
- [16] C. Bird, N. Nagappan, P. Devanbu, H. Gall, and B. Murphy, "Does distributed development affect software quality?: an empirical case study of Windows Vista," *Commun. ACM*, vol. 52, no. 8, pp. 85–93, Agosto 2009.
- [17] B. Curtis, H. Krasner, and N. Iscoe, "A field study of the software design process for large systems," *Commun. ACM*, vol. 31, no. 11, pp. 1268–1287, Nov. 1988.
- [18] A. M. Townsend, S. M. DeMarie, and A. R. Hendrickson, "Virtual teams: Technology and the workplace of the future.," *ACAD MANAGE PERSPECT*, vol. 12, no. 3, pp. 17–29, Aug. 1998.
- [19] S. Zhang, M. Tremaine, R. Egan, A. Milewski, P. O'Sullivan, and J. Fjermestad, "Occurrence and Effects of Leader Delegation in Virtual Software Teams," *International Journal of e-Collaboration*, vol. 5, no. 1, pp. 47–68, 2009.
- [20] I. Omoronyia, J. Ferguson, M. Roper, and M. Wood, "A review of awareness in distributed collaborative software engineering," *Software: Practice and Experience*, vol. 40, no. 12, pp. 1107–1133, 2010.
- [21] Galen B. Crow and Balakrishnan Muthuswamy, "International Outsourcing in the Information Technology Industry: Trends and Implications," presented at the Communications of the International Information Management Association, 2003, vol. 3.

- [22] Valentine Casey and Ita Richardson, "A Structured Approach to Global Software Development," *European Systems and Software Process*, 2008.
- [23] C. Ebert and P. De Neve, "Surviving global software development," *IEEE Software*, vol. 18, no. 2, pp. 62–69, Apr. 2001.
- [24] O. Gotel, V. Kulkarni, M. Say, C. Scharff, and T. Sunetnanta, "Quality indicators on global software development projects: does 'getting to know you' really matter?," *Journal of Software: Evolution and Process*, vol. 24, no. 2, pp. 169–184, 2012.
- [25] J. D. Herbsleb, 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. ICSE 2001*, 2001, pp. 81–90.
- [26] Djamshid Tavangarian, Markus E. Leybold, Kristin Nölting, Marc Röser, and Denny Voigt, "Is e-Learning the Solution for Individual Learning?," *Electronic Journal of e-Learning*, vol. 2, no. 2, pp. 273–280, 2004.
- [27] D. Petkovic, G. D. Thompson, and R. Todtenhoefer, "Assessment and comparison of local and global SW engineering practices in a classroom setting," *SIGCSE Bull.*, vol. 40, no. 3, pp. 78–82, Jun. 2008.
- [28] U. Bellur, "An Academic Perspective on Globalization in the Software Industry," in *Computer Software and Applications Conference, 2006. COMPSAC'06. 30th Annual International*, 2006, vol. 1, pp. 53–54.
- [29] W. L. Honig 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*, New York, NY, USA, 2007, pp. 181–185.
- [30] R. Prikladnicki and L. Pilatti, "Improving Contextual Skills in Global Software Engineering: A Corporate Training Experience," in *IEEE International Conference on Global Software Engineering, 2008. ICGSE 2008*, 2008, pp. 239–243.
- [31] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: defining 'gamification'," in *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, New York, NY, USA, 2011, pp. 9–15.
- [32] Christy Pettey, "Gartner Predicts Over 70 Percent of Global 2000 Organisations Will Have at Least One Gamified Application by 2014," *Gartner*, 09-Nov-2011. [Online]. Available: <http://www.gartner.com/it/page.jsp?id=1844115>.
- [33] Brian Burke, "What's Next: The Gamification of Everything," *Gartner*. 27-Jan-2011.
- [34] Dropbox, "Space Race," *Dropbox*, 2012. [Online]. Available: <https://www.dropbox.com/spacerace>.
- [35] Ricardo Colomo Palacios and C. C. Lumbreras, "Mentoring & Coaching. Perspectivas en las TICs Mentoring & Coaching. IT Perspective. Mentoring & Coaching. IT Perspective.," 2006.
- [36] Reiss, K., *Leadership and coaching for educators*. Thousand Oaks, California, USA: Corwin Press, 2007.
- [37] Bill Curtis, Sally A. Miller, and William E. Hefley, *People Capability Maturity Model (P-CMM)*, vol. 2. Carnegie Mellon University, Pittsburgh, Pennsylvania: Software Engineering Institute, 2001.
- [38] C. Ebert, C. H. Parro, R. Suttels, and H. Kolarczyk, "Improving validation activities in a global software development," in *Proceedings of the 23rd International Conference on Software Engineering*, Washington, DC, USA, 2001, pp. 545–554.
- [39] M. Paasivaara, "Coaching Global Software Development Projects," in *2011 6th IEEE International Conference on Global Software Engineering (ICGSE)*, 2011, pp. 84–93.
- [40] C. Casado-Lumbreras, R. Colomo-Palacios, P. Soto-Acosta, and S. Misra, "Culture dimensions in software development industry: the effects of mentoring," Jun-2011. [Online]. Available: <http://e-archivo.uc3m.es/handle/10016/14307>. [Accessed: 09-Jan-2013].
- [41] G. Hofstede, *Culture's Consequences: International Differences in Work-Related Values*. SAGE, 1980.
- [42] Hofstede, Geert, "The Hofstede Centre," *The Hofstede Centre*, 2012. [Online]. Available: <http://geert-hofstede.com/countries.html>.
- [43] José H. Canós, Patricio Letelier, and María del Carmen Penadés, "Métodologías Ágiles en el Desarrollo de Software," Universidad Politécnica de Valencia, 2003.
- [44] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas, "Manifesto for Agile Software Development," *Manifesto for Agile Software Development*, Feb-2001. [Online]. Available: <http://agilemanifesto.org/>.
- [45] A. Avritzer, F. Bronsard, and G. Matos, "Improving Global Development Using Agile," in *Agility Across Time and Space, ISBN 978-3-642-12441-9. Springer-Verlag Berlin Heidelberg, 2010, p. 133*, 2010, p. 133.
- [46] M. Paasivaara and C. Lassenius, "Could Global Software Development Benefit from Agile Methods?," in *International Conference on Global Software Engineering, 2006. ICGSE'06*, 2006, pp. 109–113.

[47] B. Ramesh, L. Cao, K. Mohan, and P. Xu, "Can distributed software development be agile?," *Commun. ACM*, vol. 49, no. 10, pp. 41–46, Oct. 2006.